

Ciencia de Datos Aplicada Capstone

Análisis de aterrizaje
FALCON9



Resume Ejecutivo

En el presente trabajo se analizará los datos de los lanzamientos del Cohete Falcon diseñado por SpaceX, aplicando herramientas de Ciencia de Datos, como Python, consulta de Python en Sql, Folium y dashboard.





Introducción

La ciencia de datos es un área que permite poder evaluar una gran cantidad de datos, con las herramientas podemos recopilar información desde distintas fuentes y formatos, podemos analizar la composición de los datos y extraer conclusiones de estos, para apoyar la toma de decisiones, presentantodolas en diferentes formas, ejemplo dashboard



Recopilación y Manejo de Datos

Se recolecto los datod utilizando la API de SpaceX enn donde se tuvo acceso a los datos de lanzamiento del cohete falcon, en donde se determine los lugares de lanzamiento y la altura alcanzada de cada cohete

Metodología EDA y de análisis visual interactivo

Aca se utilizaron comandos de SQL para tener acceso a los datos de la base de datos de los lanzamientos de Spce x



Metodología de análisis predictivo

usando herramientas predictivas se pudo determinar los lugares en donde aterrizaban los cohetes ya sea en mar, en tierra, en plataformas y su existo asociado




```
Archivos-de-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Preview Code Blame 618 lines (618 loc) · 67.4 KB Code 55% faster with GitHub Copilot

Let's first create a MarkerCluster object

In [11]: marker_cluster = MarkerCluster()

TODO: Create a new column in spacex_df dataframe called marker_color

In [12]: # Apply a function to check the value of 'class' column
# If class=1, marker_color value will be green
# If class=0, marker_color value will be red
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)

Out[12]:
```

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

```
TODO: For each launch result in spacex_df

In [13]: # Add marker_cluster to current map
site_map.add_child(marker_cluster)

# for each row in spacex_df
# create a Marker
# and customize it
# e.g.
```

```
Archivos-de-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Archivos-de-SpaceX / lab_jupyter_launch_site_location.ipynb
618 Lines (618 loc) · 67.4 KB Code 55% faster with GitHub Copilot

from folium.plugins import MousePosition
# Import folium DivIcon plugin
from folium.features import DivIcon

If you need to refresh your memory about folium, you may download and refer to this previous folium lab:

Generating Maps with Python

In [4]: ## Task 1: Mark all launch sites on a map

First, let's try to add each site's location on a map using site's latitude and longitude coordinates

The following dataset with the name 'spacex_launch_geo.csv' is an augmented dataset with latitude and longitude a site.

In [5]: # Download and read the 'spacex_launch_geo.csv'
from urllib.request import urlopen
import io

URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv'
resp = urlopen(URL)
spacex_csv_file = io.BytesIO(resp.read())
spacex_df = pd.read_csv(spacex_csv_file)

Now, you can take a look at what are the coordinates for each site.

In [6]: # Select relevant sub-columns: 'Launch Site', 'Lat(Latitude)', 'Long(Longitude)', 'class'
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df

Out[6]:
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

```
Above coordinates are just plain numbers that can not give you any intuitive insight
good at geography, you can interpret those numbers directly in your mind
Plotting them on a map.
```



Resultados de visualización

Se usaron Herramientas para visualizar en mapas los lugares de lanzamientos



Files

main

Go to file

COMPLADO

SpaceX_Machine Learning Prediction_Part_5.ipynb

edadataviz.ipynb

juptyer-labs-eda-sql-coursura_sqllite (1).ipynb

juptyer-labs-spaces-data-collection-api.ipynb

juptyer-labs-webscraping.ipynb

lab_juptyer_launch_site_location.ipynb

labs-juptyer-spaces-Data wrangling.ipynb

spaces_dash_app.py

Archivos-de-SpaceX / juptyer-labs-eda-sql-coursura_sqllite (1).ipynb

Preview Code Blame 1826 Lines (1826 loc) · 33.8 KB Code 55% faster with GitHub Copilot Raw

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 [18]: %sql select distinct(LAUNCH_SITE) from SPACEXTBL

* sqlite:///my_data1.db

Done.

Out [18]:

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 [11]: %sql select * from SPACEXTBL where LAUNCH_SITE Like 'CCA%' limit 5

* sqlite:///my_data1.db

Done.

Out [11]:

Date (UTC) Booster_Version Launch_Site Payload PAYLOAD_MASS_KG Orbit Customer Mission_Outc

Archivos-de-SpaceX / juptyer-labs-eda-sql-coursura_sqllite (1).ipynb

Preview Code Blame 1826 Lines (1826 loc) · 33.8 KB Code 55% faster with GitHub Copilot Raw

Task 5

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

Hint:Use min function

In [14]: %sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'

* sqlite:///my_data1.db

Done.

Out [14]:

min(DATE)

2018-12-22

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 [15]: %sql select Booster_Version from SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS_KG > 4000 and PAYLOAD_MASS_KG < 6000

* sqlite:///my_data1.db

Done.

Out [15]:

Booster_Version

F9 FT B1022

F9 FT B1026

DEL FT B1011.9

Archivos-de-SpaceX/jupyter-...

Revision por pares: Envíe su...

Skills Network Labs

Archivos-de-SpaceX/jupyter-...

o.com/patricioespino38/Archivos-de-SpaceX/blob/main/jupyter-labs-eda-sql-coursura_sqllite%20(1).ipynb

teligencia de Ne... DEC5 Credly Dashboarding tool... John Snow's data... Home | UNAB Intra... Aprendizaje a u

Archivos-de-SpaceX / jupyter-labs-eda-sql-coursura_sqllite (1).ipynb

Preview Code Blame 1826 Lines (1826 loc) · 33.8 KB Cod

2013-03-01 15:10:00 F9 v1.0 B0007 CCAFS LC-40 SpaceX CRS-2

Task 3

Display the total payload mass carried by boosters launched by

In [12]: %sql select sum(PAYLOAD_MASS_KG_) from SPACEXTBL wh

* sqlite:///my_data1.db

Done.

Out [12]:

sum(PAYLOAD_MASS_KG_)

45596

Task 4

Display average payload mass carried by booster version F9 v1

In [13]: %sql select avg(PAYLOAD_MASS_KG_) from SPACEXTBL wh

* sqlite:///my_data1.db

Done.

Out [13]:

avg(PAYLOAD_MASS_KG_)

2928.4

Task 5

List the date when the first succesful landing outcome in groun

Hint:Use min function

In [14]: %sql select min(DATE) from SPACEXTBL where Landing_0

Archivos-de-SpaceX/jupyter-...

Revision por pares: Envíe su...

Skills Network Labs

Archivos-de-SpaceX/jupyter-...

github.com/patricioespino38/Archivos-de-SpaceX/blob/main/jupyter-labs-eda-sql-coursura_sqllite%20(1).ipynb

Inteligencia de Ne... DEC5 Credly Dashboarding tool... John Snow's data... Home | UNAB Intra... Aprendizaje autom... RADIOGRAFÍA DE... Todos los favoritos

Files

main

Go to file

COMPLADO

SpaceX_Machine Learning Prediction_Part_5.ipynb

edadataviz.ipynb

juptyer-labs-eda-sql-coursura_sqllite (1).ipynb

juptyer-labs-spaces-data-collection-api.ipynb

juptyer-labs-webscraping.ipynb

lab_juptyer_launch_site_location.ipynb

labs-juptyer-spaces-Data wrangling.ipynb

spaces_dash_app.py

Archivos-de-SpaceX / jupyter-labs-eda-sql-coursura_sqllite (1).ipynb

Preview Code Blame 1826 Lines (1826 loc) · 33.8 KB Code 55% faster with GitHub Copilot Raw

Task 8

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

In [18]: %sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS_KG_ = (select max(PAYLOAD_MASS_KG_)

* sqlite:///my_data1.db

Done.

Out [18]:

Booster_Version

F9 B5 B1048.4

F9 B5 B1049.4

F9 B5 B1051.3

F9 B5 B1056.4

F9 B5 B1048.5

F9 B5 B1051.4

F9 B5 B1049.5

F9 B5 B1060.2

F9 B5 B1058.3

F9 B5 B1051.6

F9 B5 B1060.3

F9 B5 B1048.7

EDA con resultados SQL



Files

main

Go to file

COMPLADO

SpaceX_Machine Learning Prediction_Part_5.ipynb

edadataviz.ipynb

jupyter-labs-eda-sql-coursera_sqlite (1).ipynb

jupyter-labs-spaces-data-collection-api.ipynb

jupyter-labs-webscraping.ipynb

lab_jupyter_launch_site_location.ipynb

labs-jupyter-spaces-Data wrangling.ipynb

spacex_dash_app.py

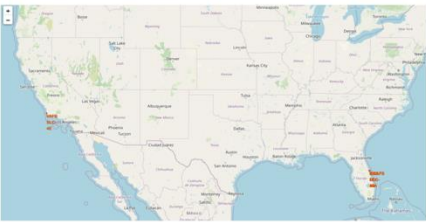
Archivos-de-SpaceX / lab_jupyter_launch_site_location.ipynb

Preview Code Blame 618 Lines (618 loc) · 67.4 KB Code 55% faster with GitHub Copilot

In [1]:

Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition,

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



Now, you can explore the map by zoom-in/out the marked areas, and try to answer the following questions:

- Are all launch sites in proximity to the Equator line?
- Are all launch sites in very close proximity to the coast?

Also please try to explain your findings.

Revisión por pares: Envíe su t...

Skills Network Labs

Archivos-de-SpaceX/lab_jup...

+

spinos38/Archivos-de-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

DEC5 Credly Dashboarding tool... John Snow's data... Home | UNAB Intra... Aprendizaje autom... RAD

Archivos-de-SpaceX / lab_jupyter_launch_site_location.ipynb

Preview Code Blame 618 lines (618 loc) · 67.4 KB Code 55% faster with G

In []:

Task 2: Mark the success/failed launches for each site on the m

Next, let's try to enhance the map by adding the launch outcomes for each site, rates. Recall that data frame spacex_df has detailed launch records, and the c_l

is successful or not

In [10]:

spacex_df.tail(10)

Out [10]:

	Launch Site	Lat	Long	class
46	KSC LC-39A	28.573255	-80.646895	1
47	KSC LC-39A	28.573255	-80.646895	1
48	KSC LC-39A	28.573255	-80.646895	1
49	CCAFS SLC-40	28.563197	-80.576820	1
50	CCAFS SLC-40	28.563197	-80.576820	1
51	CCAFS SLC-40	28.563197	-80.576820	0
52	CCAFS SLC-40	28.563197	-80.576820	0
53	CCAFS SLC-40	28.563197	-80.576820	0
54	CCAFS SLC-40	28.563197	-80.576820	1
55	CCAFS SLC-40	28.563197	-80.576820	0

Next, let's create markers for all launch records. If a launch was successful (c_l

and if a launch was failed, we use a red marker (class=0)

Note that a launch only happens in one of the four launch sites, which means ma

same coordinate. Marker clusters can be a good way to simplify a map containin

coordinate.

Let's first create a MarkerCluster object

IBM SkillsBuild

Revisión por pares: Envíe su t...

Skills Network Labs

Archivos-de-SpaceX/lab_jup...

+

github.com(patriciospinosa38)/Archivos-de-SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb

IBM Cloud Inteligencia de Na... DEC5 Credly Dashboarding tool... John Snow's data... Home | UNAB Intra... Aprendizaje autom... RADIOGRAFÍA DE... Todos los favoritos

Files

main

Go to file

COMPLADO

SpaceX_Machine Learning Prediction_Part_5.ipynb

edadataviz.ipynb

jupyter-labs-eda-sql-coursera_sqlite (1).ipynb

jupyter-labs-spaces-data-collection-api.ipynb

jupyter-labs-webscraping.ipynb

lab_jupyter_launch_site_location.ipynb

labs-jupyter-spaces-Data wrangling.ipynb

spacex_dash_app.py

Archivos-de-SpaceX / lab_jupyter_launch_site_location.ipynb

Preview Code Blame 618 Lines (618 loc) · 67.4 KB Code 55% faster with GitHub Copilot

In [12]:

Launch Site Lat Long class marker_color

46 KSC LC-39A 28.573255 -80.646895 1 green

47 KSC LC-39A 28.573255 -80.646895 1 green

48 KSC LC-39A 28.573255 -80.646895 1 green

49 CCAFS SLC-40 28.563197 -80.576820 1 green

50 CCAFS SLC-40 28.563197 -80.576820 1 green

51 CCAFS SLC-40 28.563197 -80.576820 0 red

52 CCAFS SLC-40 28.563197 -80.576820 0 red

53 CCAFS SLC-40 28.563197 -80.576820 0 red

54 CCAFS SLC-40 28.563197 -80.576820 1 green

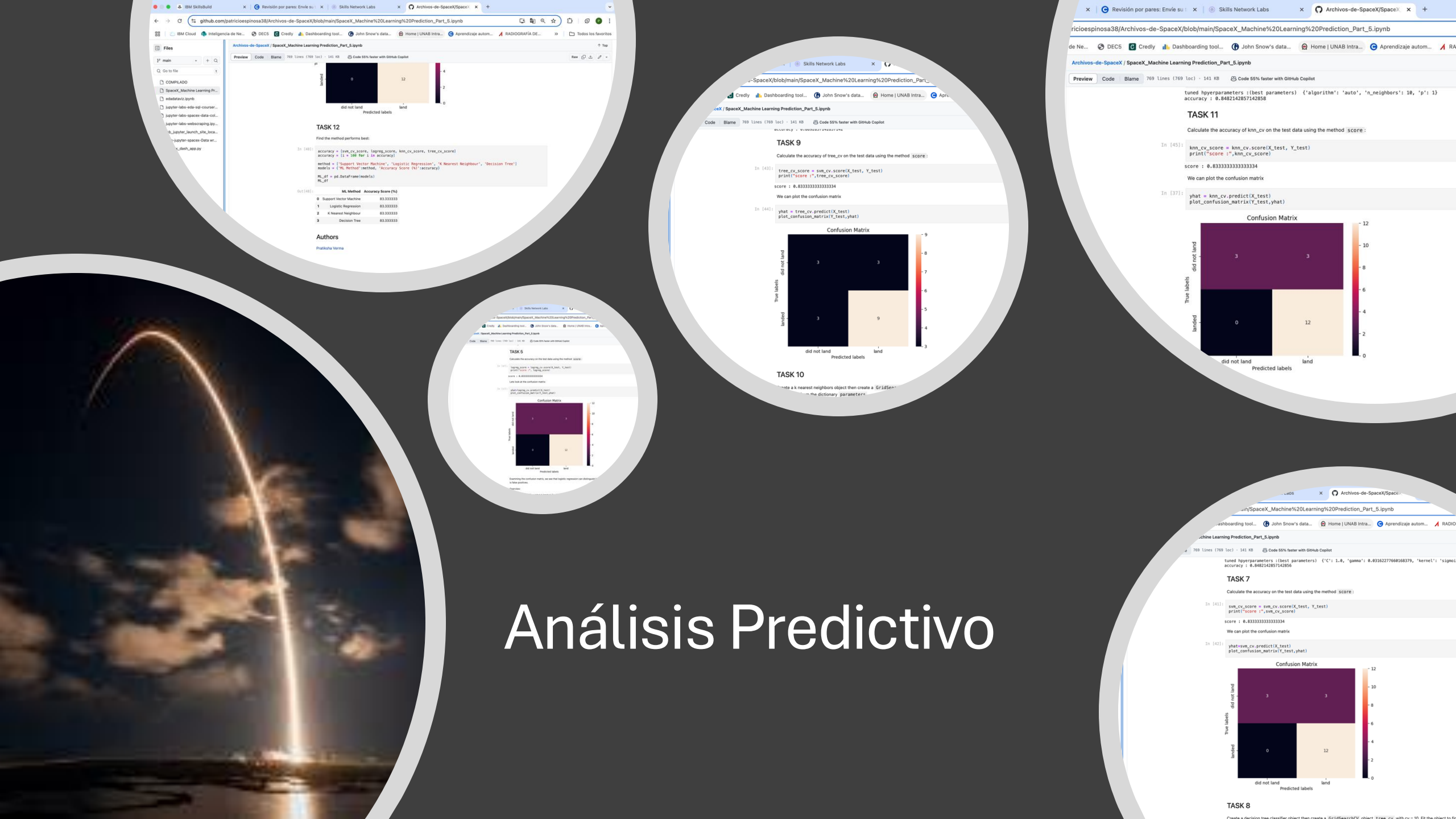
55 CCAFS SLC-40 28.563197 -80.576820 0 red

TODO: For each launch result in spacex_df data frame, add a folium.Marker to marker_cluster

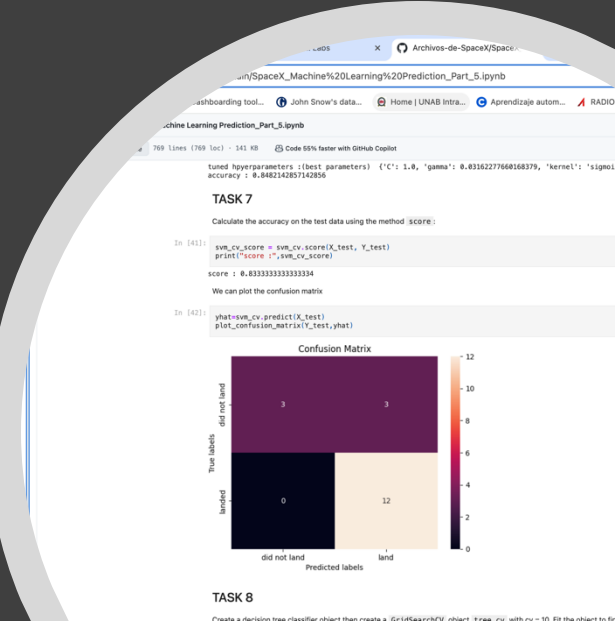
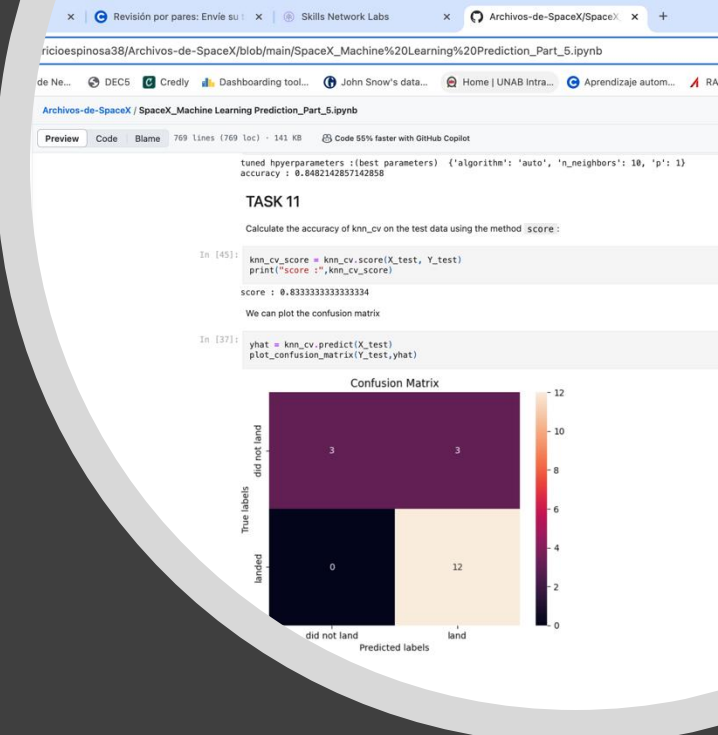
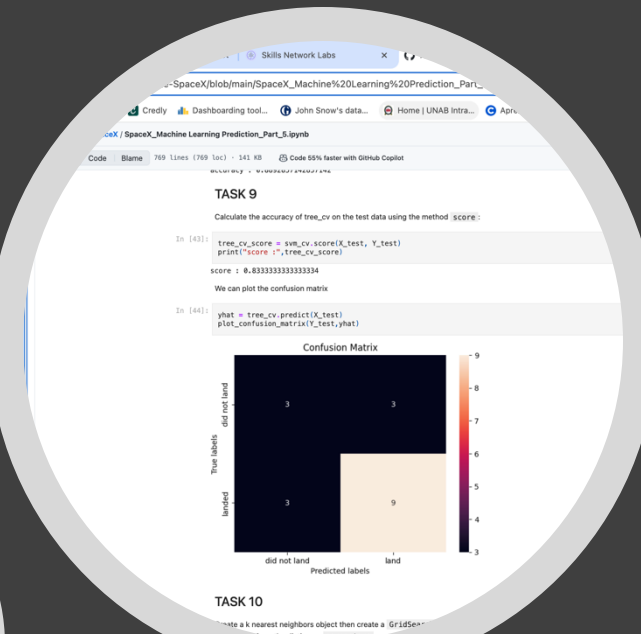
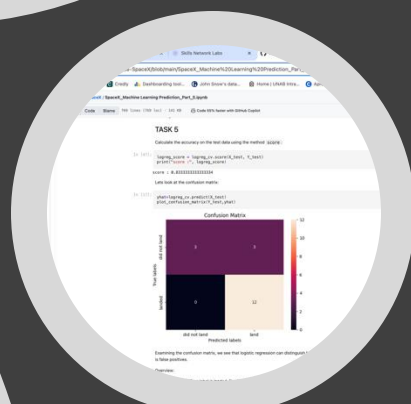
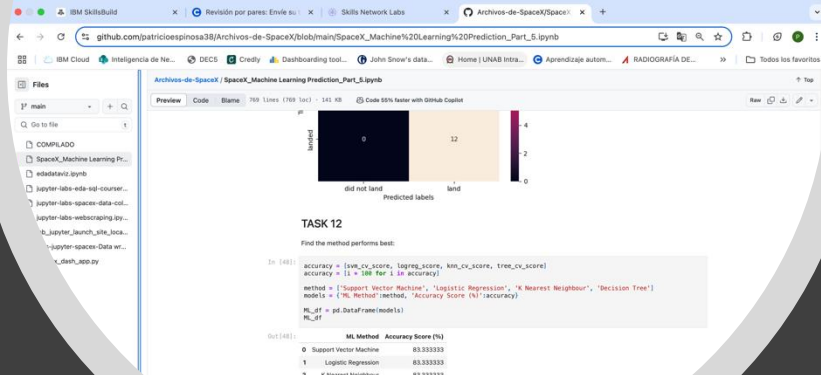
In [13]:

Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

Resultados con Folium



Análisis Predictivo



Conclusión

- El análisis de datos nos permite visualizar de forma mas limpia los datos y así determinar áreas de mejora o de cambio, es así como el análisis de estos datos ayudó a spacex a mejorar sus procesos y tener en el largo plazo un existo sustancial.

