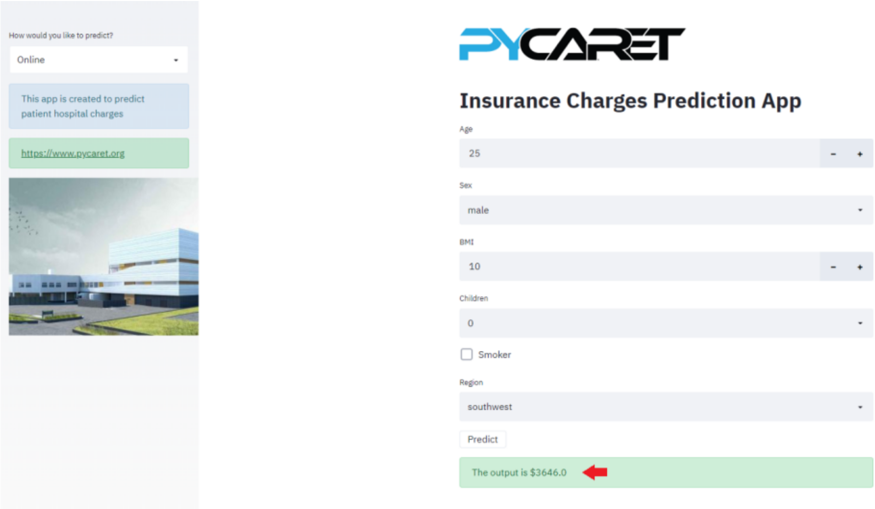
01

INTRODUCCIÓN Y HERRAMIENTAS

Entrenaremos un modelo de **ML** usando **PyCaret** y crearemos una **aplicación web con Streamlit**

Predicciones en **línea** y predicciones **por lotes**

Mas información: https://pycaret.org/



How would you like to predict?

Online

This app is created to predict patient hospital charges

https://www.pycaret.org

Insurance Charges Prediction App

Age: 25

Sex: male

BMI: 10

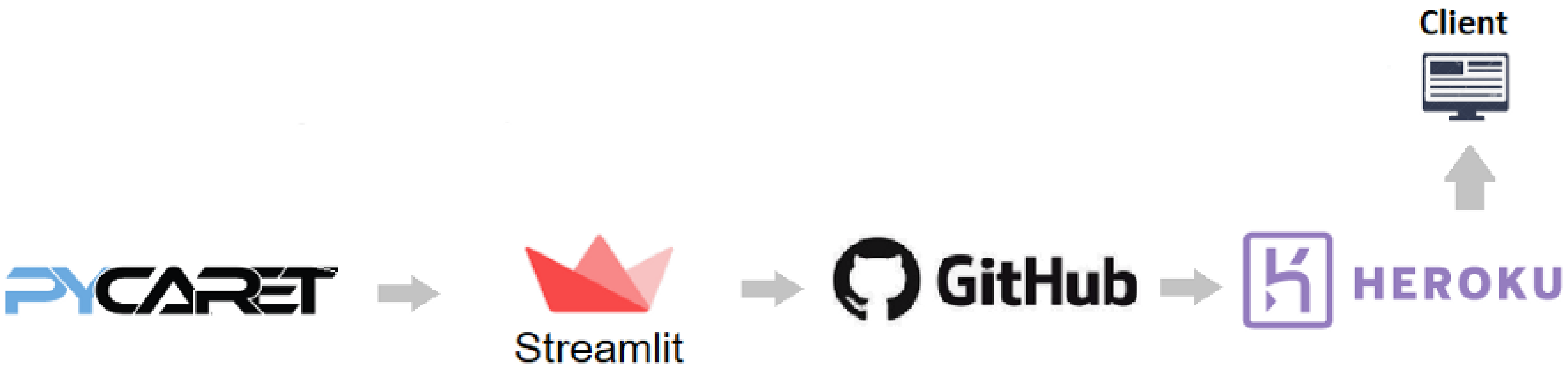
Children: 0

□ Smoker

Region: southwest

Predict

The output is $3646.0



Train Machine Learning Pipeline

Build front-end web app

Host repository / source code

PaaS to host web endpoint for ML

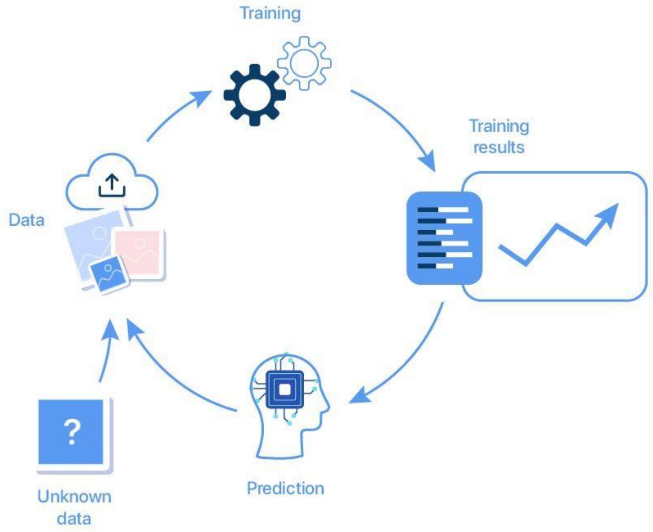
¿Por qué desplegar modelos de ML?

☞ Predicciones en línea

Generar predicciones una por una

☞ Predicciones por lotes

Generar predicciones para un conjunto de observaciones a la vez



Fuentes: https://nanonets.com/blog/machine-learning-production-retraining/ https://towardsdatascience.com/build-and-deploy-machine-learning-web-app-using-pycaret-and-streamlit-28883a569104

02

FUNDAMENTOS DE PYCARET

03

ENTRENAMIENTO Y VALIDACIÓN DEL MODELO CON PYCARET

04

CREACIÓN DE LA APLICACIÓN WEB CON STREAMLIT

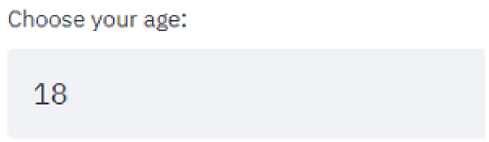
Con **Streamlit** , desarrollar **una web** para su modelo de ML se ha vuelto **increíblemente fácil.**

Ventajas de usar Streamlit :

- ¡No se necesitan conocimientos de HTML!
- Poco código para una aplicación
- Widgets se tratan como variables
- Almacenamiento en caché

Widgets

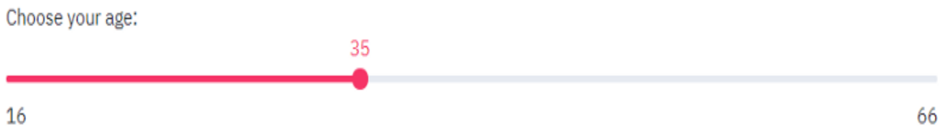
```
age = streamlit.selectbox("Choose your age: ",
np.arange(18, 66, 1))
```



Choose your age:

18

```
age = streamlit.slider("Choose your age: ", min_value=16,
max_value=66, value=35, step=1)
```

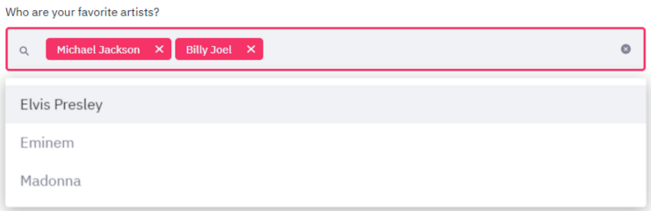


Choose your age:

35

16 66

```
artists = st.multiselect("Who are your favorite artists?",
["Michael Jackson", "Elvis Presley",
"Eminem", "Billy Joel", "Madonna"])
```



Who are your favorite artists?

Michael Jackson ✕ Billy Joel ✕

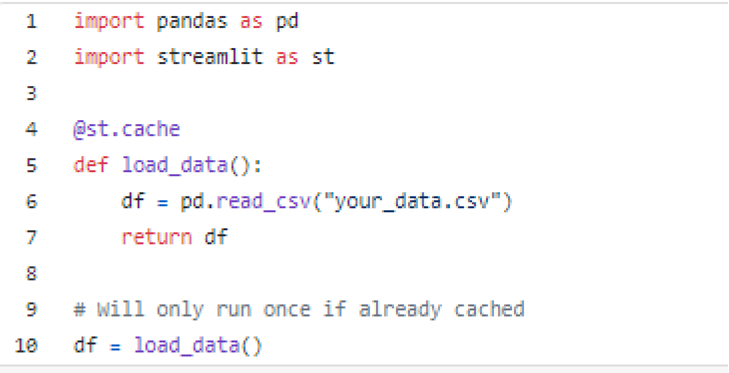
Elvis Presley

Eminem

Madonna

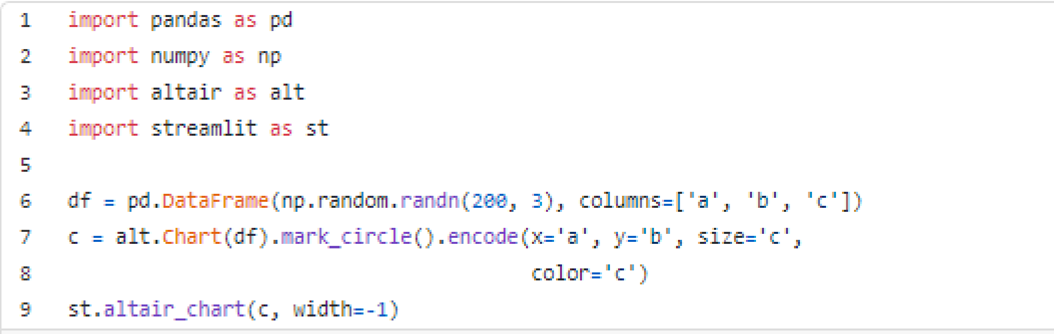
Fuente: https://github.com/MaartenGr/streamlit_guide

Caching

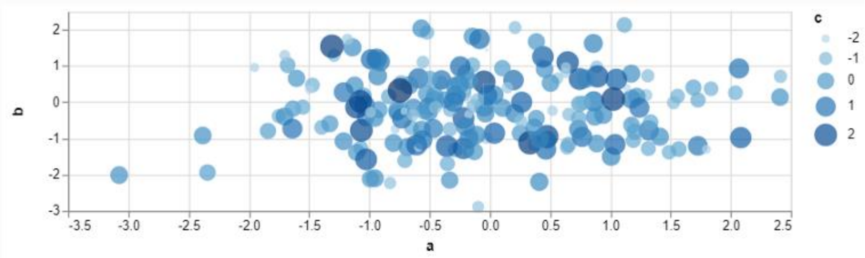


```python
import pandas as pd
import streamlit as st

@st.cache
def load_data():
    df = pd.read_csv("your_data.csv")
    return df

# Will only run once if already cached
df = load_data()
```

Visualización

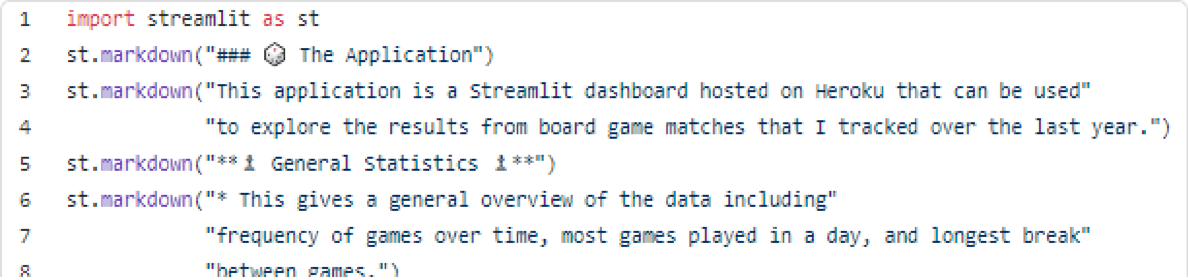


```python
import pandas as pd
import numpy as np
import altair as alt
import streamlit as st

df = pd.DataFrame(np.random.randn(200, 3), columns=['a', 'b', 'c'])
c = alt.Chart(df).mark_circle().encode(x='a', y='b', size='c',
                                       color='c')
st.altair_chart(c, width=-1)
```

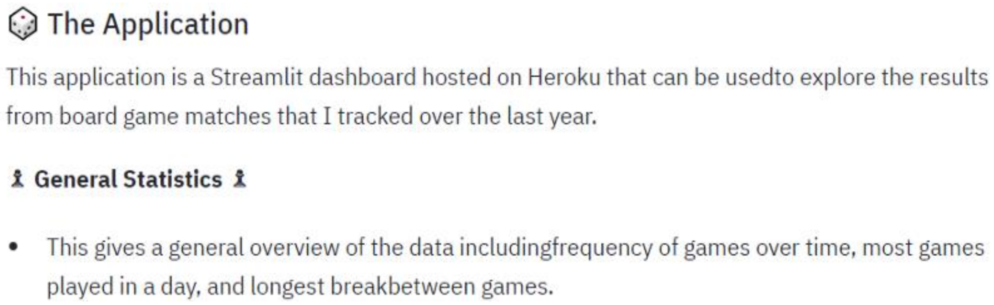


Fuente: https://github.com/MaartenGr/streamlit_guide

Markdown



```python
import streamlit as st
st.markdown("### 🎲 The Application")
st.markdown("This application is a Streamlit dashboard hosted on Heroku that can be used"
            "to explore the results from board game matches that I tracked over the last year.")
st.markdown("**♟ General Statistics ♟**")
st.markdown("* This gives a general overview of the data including"
            "frequency of games over time, most games played in a day, and longest break"
            "between games.")
```



🎲 The Application

This application is a Streamlit dashboard hosted on Heroku that can be usedto explore the results from board game matches that I tracked over the last year.

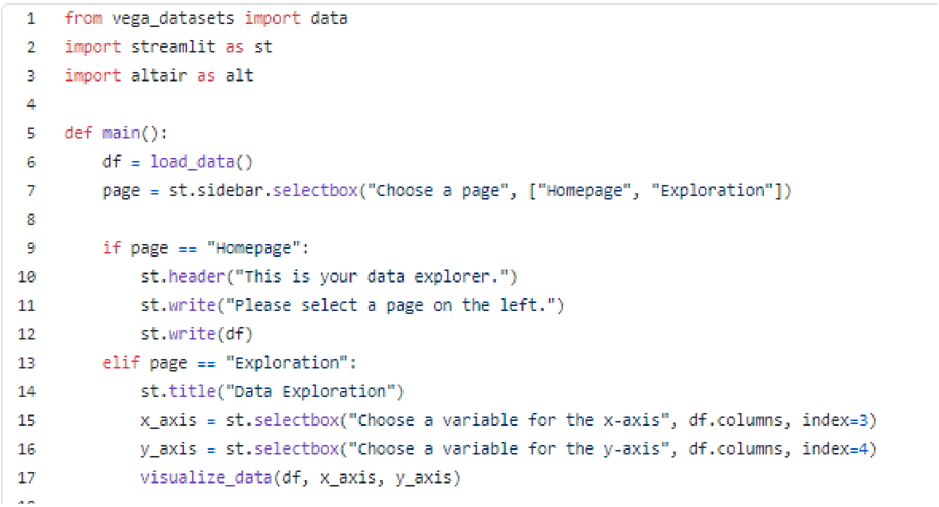
♟ General Statistics ♟

- This gives a general overview of the data includingfrequency of games over time, most games played in a day, and longest breakbetween games.

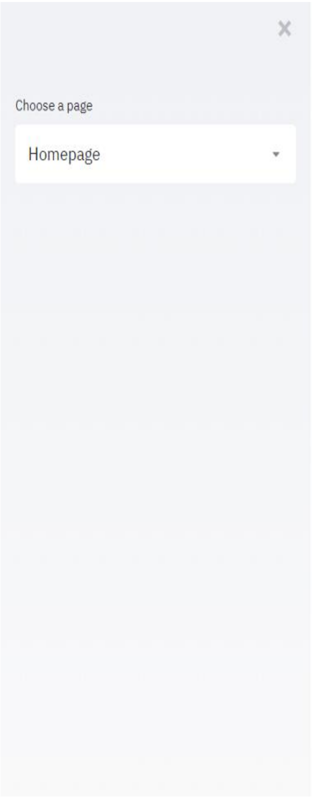
Escritura

- •write(**string**) : Markdown string.
- •write(**data_frame**) : DataFrame
- •write(**dict**) : dictionary.
- •write(**keras**) : Keras model.
- •write(**plotly_fig**) : Plotly figure.

Fuente: https://github.com/MaartenGr/streamlit_guide

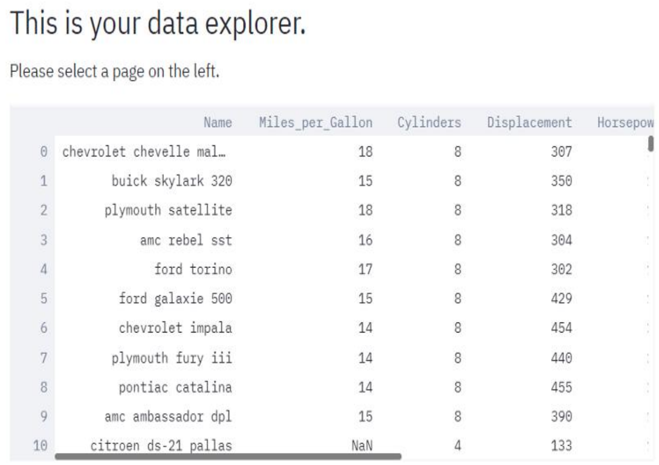


```python
from vega_datasets import data
import streamlit as st
import altair as alt

def main():
    df = load_data()
    page = st.sidebar.selectbox("Choose a page", ["Homepage", "Exploration"])

    if page == "Homepage":
        st.header("This is your data explorer.")
        st.write("Please select a page on the left.")
        st.write(df)
    elif page == "Exploration":
        st.title("Data Exploration")
        x_axis = st.selectbox("Choose a variable for the x-axis", df.columns, index=3)
        y_axis = st.selectbox("Choose a variable for the y-axis", df.columns, index=4)
        visualize_data(df, x_axis, y_axis)
```



Choose a page

Homepage

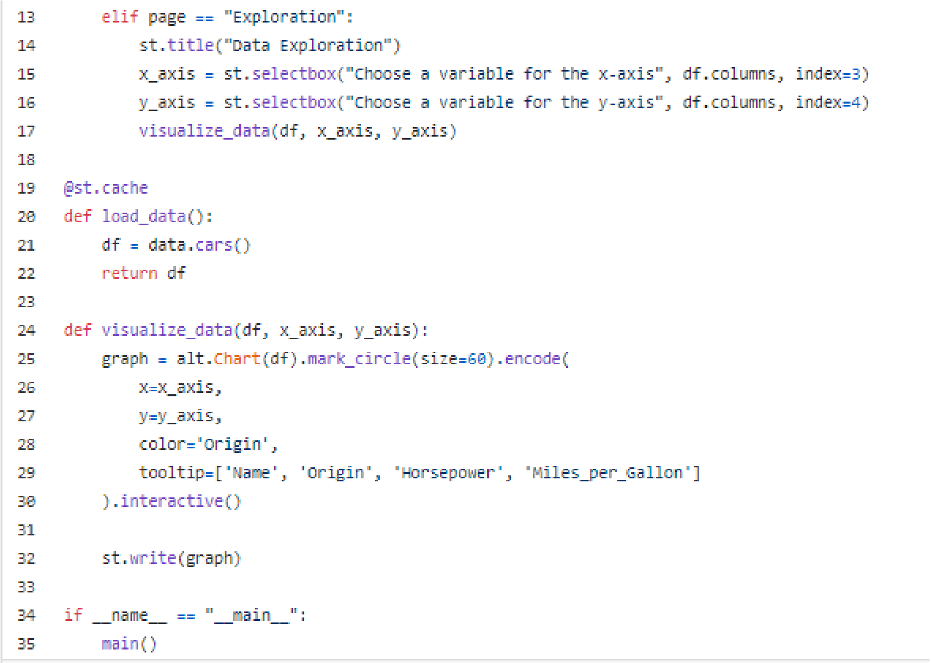


This is your data explorer.

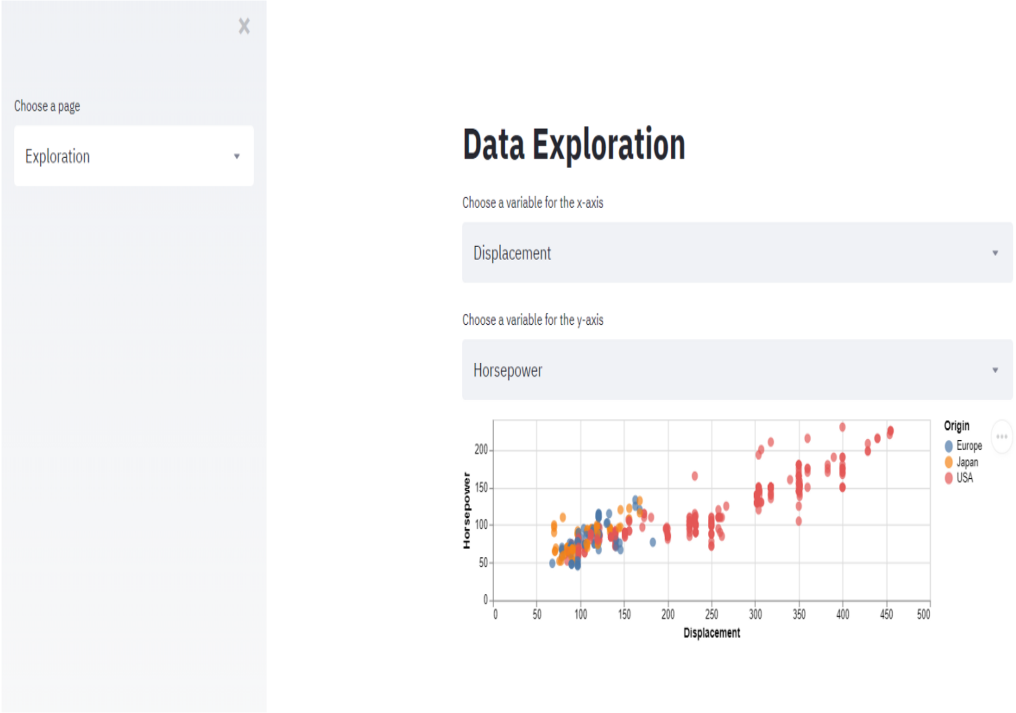
Please select a page on the left.

	Name	Miles_per_Gallon	Cylinders	Displacement	Horsepow
0	chevrolet chevelle mal…	18	8	307	
1	buick skylark 320	15	8	350	
2	plymouth satellite	18	8	318	
3	amc rebel sst	16	8	304	
4	ford torino	17	8	302	
5	ford galaxie 500	15	8	429	
6	chevrolet impala	14	8	454	
7	plymouth fury iii	14	8	440	
8	pontiac catalina	14	8	455	
9	amc ambassador dpl	15	8	390	
10	citroen ds-21 pallas	NaN	4	133	

Fuente: https://github.com/MaartenGr/streamlit_guide



```python
    elif page == "Exploration":
        st.title("Data Exploration")
        x_axis = st.selectbox("Choose a variable for the x-axis", df.columns, index=3)
        y_axis = st.selectbox("Choose a variable for the y-axis", df.columns, index=4)
        visualize_data(df, x_axis, y_axis)

@st.cache
def load_data():
    df = data.cars()
    return df

def visualize_data(df, x_axis, y_axis):
    graph = alt.Chart(df).mark_circle(size=60).encode(
        x=x_axis,
        y=y_axis,
        color='Origin',
        tooltip=['Name', 'Origin', 'Horsepower', 'Miles_per_Gallon']
    ).interactive()

    st.write(graph)

if __name__ == "__main__":
    main()
```



Choose a page

Exploration

Data Exploration

Choose a variable for the x-axis

Displacement

Choose a variable for the y-axis

Horsepower

Fuente: https://github.com/MaartenGr/streamlit_guide

05

DESPLIEGUE DEL CÓDIGO EN GIT Y HEROKU