(CODE FOR THE PROJECT)

```
import pandas as pd

df = pd.read_csv('gene_expression_dataset.csv')
print(df.head())
# 1. Basic Tumor Size Regression in Python

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt

# Load your dataset
df = pd.read_csv('gene_expression_dataset.csv')

# Separate features and target
X = df.drop(columns=['Tumor_Size_cm'])  # input genes
y = df['Tumor_Size_cm']                 # target tumor size

# Train-test split (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build and train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Evaluation
print(f"R² Score: {r2_score(y_test, y_pred):.2f}")
print(f"Mean Absolute Error (MAE): {mean_absolute_error(y_test, y_pred):.2f}")
print(f"Root Mean Squared Error (RMSE): {np.sqrt(mean_squared_error(y_test,
y_pred)):.2f}")
# Plot Actual vs Predicted Tumor Sizes

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')  # perfect prediction line
plt.xlabel('Actual Tumor Size (cm)')
plt.ylabel('Predicted Tumor Size (cm)')
plt.title('Actual vs Predicted Tumor Size')
plt.grid(True)
plt.show()
!pip install streamlit
!pip install pyngrok
%%writefile app.py
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

st.title('Tumor Size Prediction from Gene Expression')

uploaded_file = st.file_uploader("Upload your gene expression dataset (.csv)",
type=["csv"])

if uploaded_file is not None:
    data = pd.read_csv(uploaded_file)
    st.write("Preview of Dataset:", data.head())

    X = data.drop(columns=['Tumor_Size_cm'])
    y = data['Tumor_Size_cm']

    model = LinearRegression()
    model.fit(X, y)

    st.subheader("Enter Gene Expression Values for Prediction")

    input_data = []
    for gene in X.columns:
        value = st.number_input(f"Enter expression value for {gene}", min_value=0.0,
max_value=20.0, step=0.1)
        input_data.append(value)

    if st.button('Predict Tumor Size'):
        input_array = np.array(input_data).reshape(1, -1)
        prediction = model.predict(input_array)
        st.success(f"Predicted Tumor Size: {prediction[0]:.2f} cm")
else:
    st.info("Please upload the CSV file to continue.")
# Upload your CSV file
from google.colab import files
uploaded = files.upload()

# It will ask you to upload your 'gene_expression_dataset.csv'
# Make sure the file name matches exactly
%%writefile /content/app.py
import streamlit as st
st.title("Hello, Everyone👋")
st.write("Welcome to Tumor Size Predictor – A non-invasive gene-based estimation tool
powered by Machine Learning.")
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

st.title('Tumor Size Prediction from Gene Expression')

uploaded_file = st.file_uploader("Upload your gene expression dataset (.csv)",
type=["csv"])

if uploaded_file is not None:
   data = pd.read_csv(uploaded_file)
   st.write("Preview of Dataset:", data.head())

   X = data.drop(columns=['Tumor_Size_cm'])
   y = data['Tumor_Size_cm']

   model = LinearRegression()
   model.fit(X, y)

   st.subheader("Enter Gene Expression Values for Prediction")

   input_data = []
   for gene in X.columns:
      value = st.number_input(f"Enter expression value for {gene}", min_value=0.0,
max_value=20.0, step=0.1)
      input_data.append(value)

   if st.button('Predict Tumor Size'):
      input_array = np.array(input_data).reshape(1, -1)
      prediction = model.predict(input_array)
      st.success(f"Predicted Tumor Size: {prediction[0]:.2f} cm")
else:
   st.info("Please upload the CSV file to continue.")
!ls /content/
!pip install pyngrok #Install pyngrok library.
# Import pyngrok and start the Streamlit app
from pyngrok import ngrok

# Replace 'YOUR_AUTHTOKEN' with your actual authtoken
ngrok.set_auth_token("2wJgIolJM0UZvxswaQzhieoQTmr_7hAQseZ9XnsYVmbY5nP5J")

# Start the Streamlit app
!streamlit run app.py &>/content/logs.txt &

# Start streamlit server
```

```
!nohup streamlit run /content/app.py --server.port 8501 &

!pip install pyngrok #Install pyngrok library.
# Import pyngrok and start the Streamlit app
from pyngrok import ngrok

# Replace 'YOUR_AUTHTOKEN' with your actual authtoken
ngrok.set_auth_token("2wJgIolJM0UZvxswaQzhieoQTmr_7hAQseZ9XnsYVmbY5nP5J")

# Start the Streamlit app
!streamlit run app.py &>/content/logs.txt &

# Start streamlit server
!nohup streamlit run /content/app.py --server.port 8501 &

# Create public URL
# The 'addr' parameter is used instead of 'port'
public_url = ngrok.connect(addr='8501')
print(f"Your Streamlit app is available at: {public_url}")
```