```
#!/usr/bin/env python
# coding: utf-8

# ## **PREDICTING_COVID-19_IN_HUNGARY_USING_MACHINE_LEARNING**

# ##### IMPORTING LIBRARIES

# In[1]:


import pandas as pd
import seaborn as sns
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error


# ##### LOADING THE DATASET

# In[2]:


covid_dataset = pd.read_csv(r'E:\Data_Science\PROJECTS\PREDICTING_COVID-19_USING_MACHINE_LEARNING\covid.csv')


# ##### EXPLORING THE DATASET

# In[3]:


covid_dataset.head(10)


# In[4]:


covid_dataset.tail(10)


# In[5]:


covid_dataset.shape


# In[6]:


covid_dataset.columns


# In[7]:


covid_dataset.dtypes


# In[9]:


covid_dataset.describe()


# In[10]:


covid_dataset['location'].value_counts()


# ##### DATA WARNGLING

# In[11]:


# how many duplicated rows we have
covid_dataset.duplicated().sum()


# In[14]:


# checking for missing values
covid_dataset.isna().any()


# In[15]:


# sum of null values
covid_dataset.isna().sum()


# ##### We are interested in the cases in Hungary, so I create a dataframe from the cases occured in Hungary

# In[16]:


covid_Hungary=covid_dataset[covid_dataset["location"]=="Hungary"]
```

```
# In[17]:


covid_Hungary.head(10)


# In[18]:


covid_Hungary.tail(10)


# In[19]:


covid_Hungary.shape


# In[36]:


# Total cases per day
sns.lineplot(x="date",y="total_cases",data=covid_Hungary)
plt.show()


# In[25]:


# Total cases in the first 10 days
first_10 = covid_Hungary.head(10)


# In[27]:


sns.lineplot(x='date', y = 'total_cases', data = first_10, color = 'r')
plt.show()


# In[33]:


# Total cases in the last 10 days
last_10 = covid_Hungary.tail(10)


# In[34]:


sns.lineplot(x = 'date', y = 'total_cases', data = last_10, color = 'r')
plt.show()


# In[31]:


# Total death cases in the first 10 days
sns.lineplot(x = 'date', y = 'total_deaths', data = first_10, color = 'g')
plt.show()


# In[32]:


# Total death cases in the last 10 days
sns.lineplot(x = 'date', y = 'total_deaths', data = last_10, color = 'g')
plt.show()


# In[40]:


# Top 5 countries with the most cases on the last day
last_date_data = covid_dataset[covid_dataset["date"]=="2020-05-24"]
max_cases=last_date_data.sort_values(by="total_cases",ascending=False)
max_cases[1:6]


# In[47]:


# European countries with the most cases and Hungary
Hungary_Italy_Germany=covid_dataset[(covid_dataset["location"] =="Italy") | (covid_dataset["location"]=="Germany") | (covid_dataset["location"]=="Hungary")]


# In[49]:


#  Visualizing the growth of cases across Hungary, Italy and Germany
sns.set(rc={'figure.figsize':(15,10)})
sns.barplot(x="location",y="total_cases",data=Hungary_Italy_Germany,hue="date")
plt.show()


# In[50]:


# Creating a bar plot for countries with top cases
sns.barplot(x="location",y="total_cases",data=max_cases[1:6],hue="location")
plt.show()
```

```
# ##### LINEAR REGRESSION

# In[53]:


lr = LinearRegression()


# In[59]:


# defining the variables
x = covid_Hungary['date']
y = covid_Hungary['total_cases']


# In[60]:


x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)


# In[61]:


lr.fit(np.array(x_train).reshape(-1,1),np.array(y_train).reshape(-1,1))


# In[62]:


y_pred=lr.predict(np.array(x_test).reshape(-1,1))


# In[63]:


mean_squared_error(x_test,y_pred)


# In[64]:


lr.predict(np.array([[737573]]))


# In[ ]:
```