**Covid Vaccine Analysis**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

import plotly.express as px

url = "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations.csv"

df = pd.read\_csv(url)

df['date'] = pd.to\_datetime(df['date'])

df = df[df['total\_vaccinations'].notna()]

df = df[df['iso\_code'].apply(lambda x: len(x) == 3)]

df = df.groupby(['location', 'date']).agg({

    'total\_vaccinations\_per\_hundred': 'max',

    'people\_fully\_vaccinated\_per\_hundred': 'max',

    'daily\_vaccinations\_per\_million': 'mean'

}).reset\_index()

latest = df.sort\_values('date').groupby('location').tail(1)

latest = latest.dropna()

plt.figure(figsize=(10,6))

sns.histplot(latest['people\_fully\_vaccinated\_per\_hundred'], bins=30, kde=True)

plt.title("Distribution of Fully Vaccinated People per 100")

plt.xlabel("People Fully Vaccinated per 100")

plt.ylabel("Frequency")

plt.show()

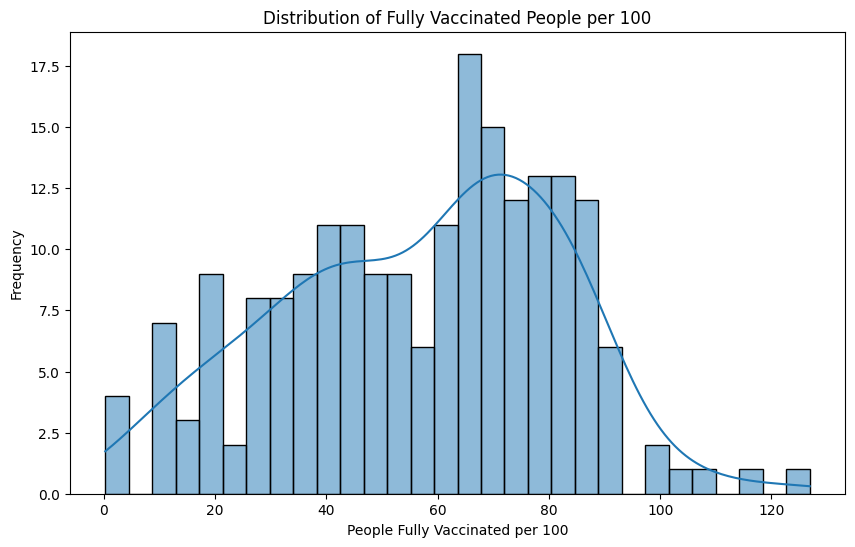
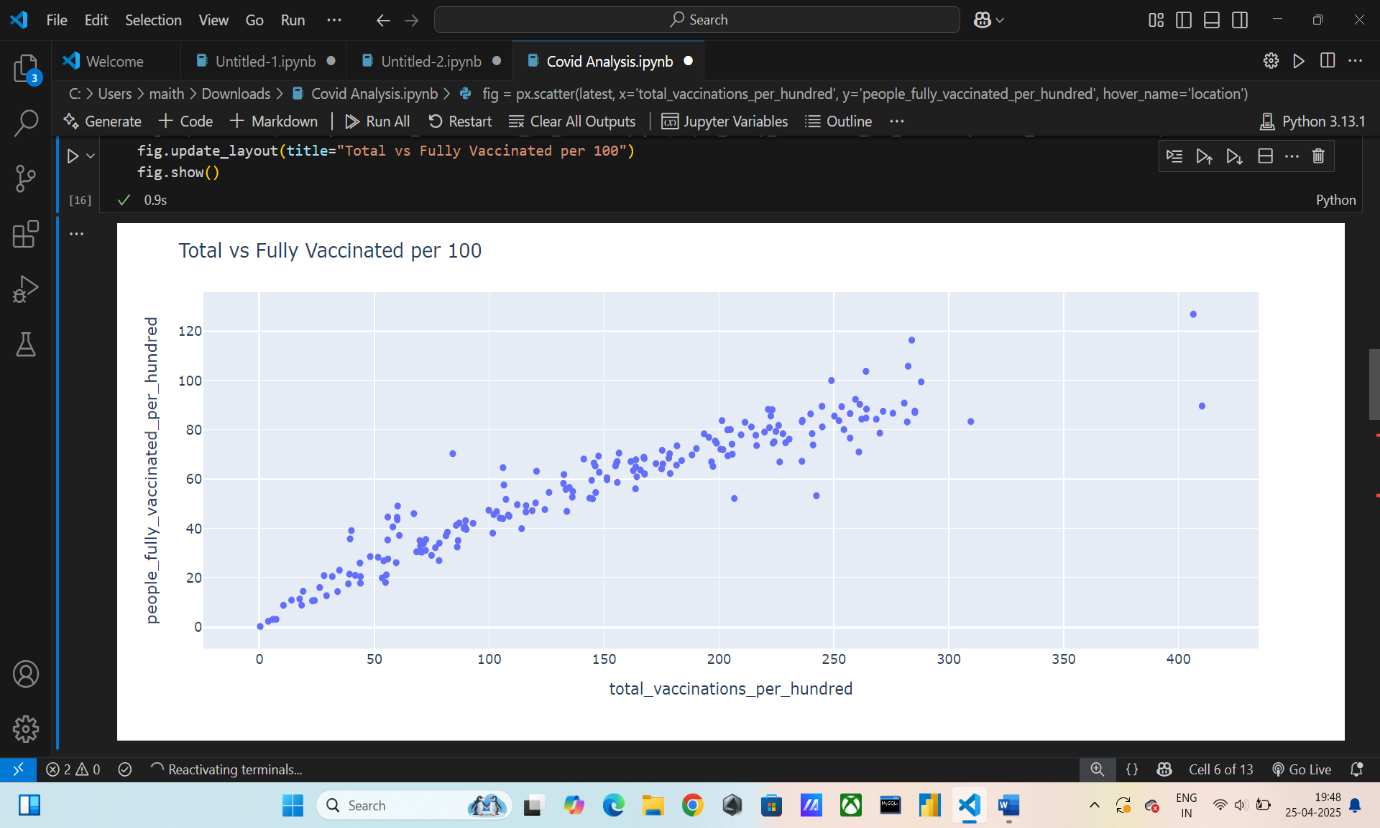


fig = px.scatter(latest, x='total\_vaccinations\_per\_hundred', y='people\_fully\_vaccinated\_per\_hundred', hover\_name='location')

fig.update\_layout(title="Total vs Fully Vaccinated per 100")

fig.show()



X = latest[['total\_vaccinations\_per\_hundred', 'daily\_vaccinations\_per\_million']]

y = latest['people\_fully\_vaccinated\_per\_hundred']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

importances = model.feature\_importances\_

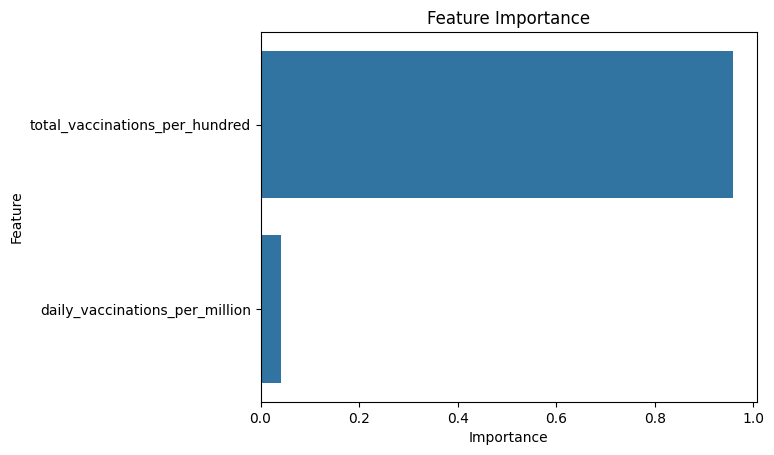
feature\_names = X.columns

feat\_imp\_df = pd.DataFrame({'Feature': feature\_names, 'Importance': importances}).sort\_values(by='Importance', ascending=False)

sns.barplot(x='Importance', y='Feature', data=feat\_imp\_df)

plt.title("Feature Importance")

plt.show()



import joblib

joblib.dump(model, 'vaccination\_model.pkl')

joblib.dump(scaler, 'scaler.pkl')

O/P:

['scaler.pkl']

print("\nTop Insights:")

print("- Countries with higher total vaccinations per hundred tend to have higher fully vaccinated rates.")

print("- Daily vaccination rate is a significant factor in full vaccination coverage.")

O/P:

Top Insights:

- Countries with higher total vaccinations per hundred tend to have higher fully vaccinated rates.

- Daily vaccination rate is a significant factor in full vaccination coverage.