**NAAN MUDHALVAN –DATA ANALYTICS WITH COGNOS**

**COVID-19 VACCINE ANALYSIS**

**TRAINING METHODS:**

**Data Collection:**

Gather COVID-19-related data like infection rates, demographics, vaccination data, and other relevant information. Datasets can be obtained from government health agencies, research institutions, or open data sources.

**Data Preprocessing:**

Data Cleaning: Handle missing values, outliers, and inconsistencies in the dataset.

Feature Engineering: Create relevant features based on domain knowledge.

Data Splitting: Divide the data into training, validation, and testing sets.

**Feature Selection and Scaling:**

Choose which features to include in your model.

Scale or normalize features to ensure they have similar scales.

**Model Selection and Hyperparameter Tuning:**

Select an appropriate machine learning model (e.g., linear regression, decision tree, random forest, neural network).

Tune model hyperparameters to optimize performance. This can include parameters like learning rates, tree depths, batch sizes, etc.

**Training:**

Train the model using the training dataset.

Monitor and record model performance on the validation dataset to avoid overfitting.

**Evaluation:**

Evaluate the model's performance using relevant evaluation metrics (e.g., mean squared error, accuracy, ROC AUC, F1-score, etc.) on the testing dataset.

Adjust the model or data preprocessing based on the evaluation results.

**Interpretation and Visualization:**

Interpret the model's parameters and visualize its decisions to gain insights into COVID-19-related patterns.

**Regularization and Optimization:**

Implement regularization techniques (e.g., L1 or L2 regularization) to prevent overfitting.

Optimize the model architecture and parameters.

**Deployment:**

If the model is satisfactory, deploy it in a production environment to make predictions.

**Monitoring and Maintenance:**

Continuously monitor the model's performance in a production environment.

Retrain the model with new data as it becomes available.

**Program:**

# Import necessary libraries

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.linear\_model import LinearRegression

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

# Load the dataset

data = pd.read\_csv('covid19\_vaccine\_dataset.csv')

# Data Preprocessing

# Handle missing values

data.dropna(inplace=True)

# Convert 'Date' column to datetime

data['Date'] = pd.to\_datetime(data['Date'])

# Extract year and month from the 'Date' column

data['Year'] = data['Date'].dt.year

data['Month'] = data['Date'].dt.month

# Select relevant columns for modeling

features = ['Year', 'Month', 'People Vaccinated', 'People Fully Vaccinated', 'Daily Vaccine Count']

target = 'Total Vaccinations'

X = data[features]

y = data[target]

# Split the data into training and testing sets

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

# Model training and tuning (Linear Regression as an example)

model = LinearRegression()

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

# Make predictions

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

# Model evaluation

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

print(f"R-squared (R2) Score: {r2}")

# Data Visualization

# Plot the actual vs. predicted values

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred)

plt.xlabel("Actual Total Vaccinations")

plt.ylabel("Predicted Total Vaccinations")

plt.title("Actual vs. Predicted Total Vaccinations")

plt.show()

# Visualize the distribution of 'Total Vaccinations'

plt.figure(figsize=(8, 6))

sns.histplot(data['Total Vaccinations'], kde=True)

plt.xlabel("Total Vaccinations")

plt.ylabel("Frequency")

plt.title("Distribution of Total Vaccinations")

plt.show()

OUTPUT :





