World Population Analysis Project

project Title:

World population anaysis

Lanuage/Tools:

Machine leraning,Python,Excel,jupyter notebook.

Domain:

Data Analyst

About DataSet

Context:

The current US Census Bureau world population estimate in June 2019 shows that the current global population is 7,577,130,400 people on earth, which far exceeds the world population of 7.2 billion in 2015. Our own estimate based on UN data shows the world's population surpassing 7.7 billion.

The following 11 countries that are the most populous in the world each have populations exceeding 100 million. These include the United States, Indonesia, Brazil, Pakistan, Nigeria, Bangladesh, Russia, Mexico, Japan, Ethiopia, and the Philippines. Of these nations, all are expected to continue to grow except Russia and Japan, which will see their populations drop by 2030 before falling again significantly by 2050.

This population growth will be significantly impacted by nine specific countries which are situated to contribute to the population growing more quickly than other nations. These nations include the Democratic Republic of the Congo, Ethiopia, India, Indonesia, Nigeria, Pakistan, Uganda, the United Republic of Tanzania, and the United States of America. Particularly of interest, India is on track to overtake China's position as the most populous country by 2030. Additionally, multiple nations within Africa are expected to double their populations before fertility rates begin to slow entirely.

Content:

In this Dataset, we have Historical Population data for every Country/Territory in the world by different parameters like Area Size of the Country/Territory, Name of the Continent, Name of the Capital, Density, Population Growth Rate, Ranking based on Population, World Population Percentage, etc.

Dataset column names and description:

- **Rank:** Rank by Population.
- **CCA3:** 3 Digit Country/Territory code.
- **Country/Territories:** Name of Country/territories.
- **Capital:** Name of the Capital.
- **Continent:** Name of the continent
- **2022 population:** population of the country/territories in the yar 2022
- **2020 Population:** Population of the Country/Territories in the year 2020.
- **2015 Population:** Population of the Country/Territories in the year 2015.
- **2010 Population:** Population of the Country/Territories in the year 2010.
- **2000 Population:** Population of the Country/Territories in the year 2000.
- **1990 Population:** Population of the Country/Territories in the year 1990.
- **1980 Population:** Population of the Country/Territories in the year 1980.
- **1970 Population:** Population of the Country/Territories in the year 197
- **Area(km^2):** Area size of the Country/territories in square kilometer
- **Density(per km^2):** Population Density per square kilometer.
- **Growth Rate:** Population Growth rate by country /territories.
- **World Population Percentage:** The Popultion Percentage by each Country/territories.

Project Overview:

The Goal of the project is to analyze global population trends using historical data and predict future population growth. This involves using machine learning techniques to explore demograpic data,identify key factors influening population changes and build predictive model.

Steps and Implementation:

1. Data Collection
2. Data Preprocessing
3. Exploratory Data Analysis (EDA)
4. Feature Engineering
5. Model Building
6. Model Evaluation
7. Visualization
8. Report Generation

1. Data Collection:

```
- Source of data.
- Description of Dateset Features.
```

In [1]:
```
#importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.subplots as sp
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [2]:
```
# Load the Dataset
data=pd.read_csv('world_population.csv')
```

In [3]:
```
#Display Info about the Dataset.
data.info()  # another format :print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 17 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Rank                          234 non-null    int64
 1   CCA3                          234 non-null    object
 2   Country/Territory            234 non-null    object
 3   Capital                       234 non-null    object
 4   Continent                     234 non-null    object
 5   2022 Population               234 non-null    int64
 6   2020 Population               234 non-null    int64
 7   2015 Population               234 non-null    int64
 8   2010 Population               234 non-null    int64
 9   2000 Population               234 non-null    int64
 10  1990 Population               234 non-null    int64
 11  1980 Population               234 non-null    int64
 12  1970 Population               234 non-null    int64
 13  Area (km²)                    234 non-null    int64
 14  Density (per km²)             234 non-null    float64
 15  Growth Rate                   234 non-null    float64
 16  World Population Percentage  234 non-null    float64
dtypes: float64(3), int64(10), object(4)
memory usage: 31.2+ KB
```

In [4]:
```
data.head()  #another format: print(data.head())
```

Out[4]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	Po
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672	19542982	10694796	12486631	1
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399	3182021	3295066	2941651	
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344	30774621	25518074	18739378	1
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849	58230	47818	32886	
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519	66097	53569	35611	

2. Data Preprocessing

- Handling missing values.
- Feature selection

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
Rank                           0
CCA3                           0
Country/Territory              0
Capital                        0
Continent                      0
2022 Population                0
2020 Population                0
2015 Population                0
2010 Population                0
2000 Population                0
1990 Population                0
1980 Population                0
1970 Population                0
Area (km²)                     0
Density (per km²)              0
Growth Rate                    0
World Population Percentage    0
dtype: int64
```

In [6]:
```
data.isna().sum()
```

Out[6]:
```
Rank                           0
CCA3                           0
Country/Territory              0
Capital                        0
Continent                      0
2022 Population                0
2020 Population                0
2015 Population                0
2010 Population                0
2000 Population                0
1990 Population                0
1980 Population                0
1970 Population                0
Area (km²)                     0
Density (per km²)              0
Growth Rate                    0
World Population Percentage    0
dtype: int64
```

In [7]:
```
data=data.dropna()
```

In [8]:
```
print(f"Amount of duplicates: {data.duplicated().sum()}" )
```

```
Amount of duplicates: 0
```

In [9]:
```
data.shape
```

Out[9]:
```
(234, 17)
```

In [10]:
```
data.head()
```

Out[10]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	Po
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672	19542982	10694796	12486631	1
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399	3182021	3295066	2941651	
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344	30774621	25518074	18739378	1
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849	58230	47818	32886	
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519	66097	53569	35611	

Feature selection & engineering

Create additional features if necessary

In [11]:
```
data.head()
```

Out[11]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	Po
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672	19542982	10694796	12486631	1
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399	3182021	3295066	2941651	
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344	30774621	25518074	18739378	1
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849	58230	47818	32886	
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519	66097	53569	35611	

In [12]:
```
data.columns
```

Out[12]:
```
Index(['Rank', 'CCA3', 'Country/Territory', 'Capital', 'Continent',
       '2022 Population', '2020 Population', '2015 Population',
       '2010 Population', '2000 Population', '1990 Population',
       '1980 Population', '1970 Population', 'Area (km²)', 'Density (per km²)',
       'Growth Rate', 'World Population Percentage'],
      dtype='object')
```

In [13]:
```
data.drop(['CCA3','Capital'],axis=1,inplace=True)
```

In [14]:
```
data.head()
```

Out[14]:

	Rank	Country/Territory	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population	Are (km
0	36	Afghanistan	Asia	41128771	38972230	33753499	28189672	19542982	10694796	12486631	10752971	65223
1	138	Albania	Europe	2842321	2866849	2882481	2913399	3182021	3295066	2941651	2324731	2874
2	34	Algeria	Africa	44903225	43451666	39543154	35856344	30774621	25518074	18739378	13795915	238174
3	213	American Samoa	Oceania	44273	46189	51368	54849	58230	47818	32886	27075	19
4	203	Andorra	Europe	79824	77700	71746	71519	66097	53569	35611	19860	46

3. Exploratory Data Analysis (EDA)

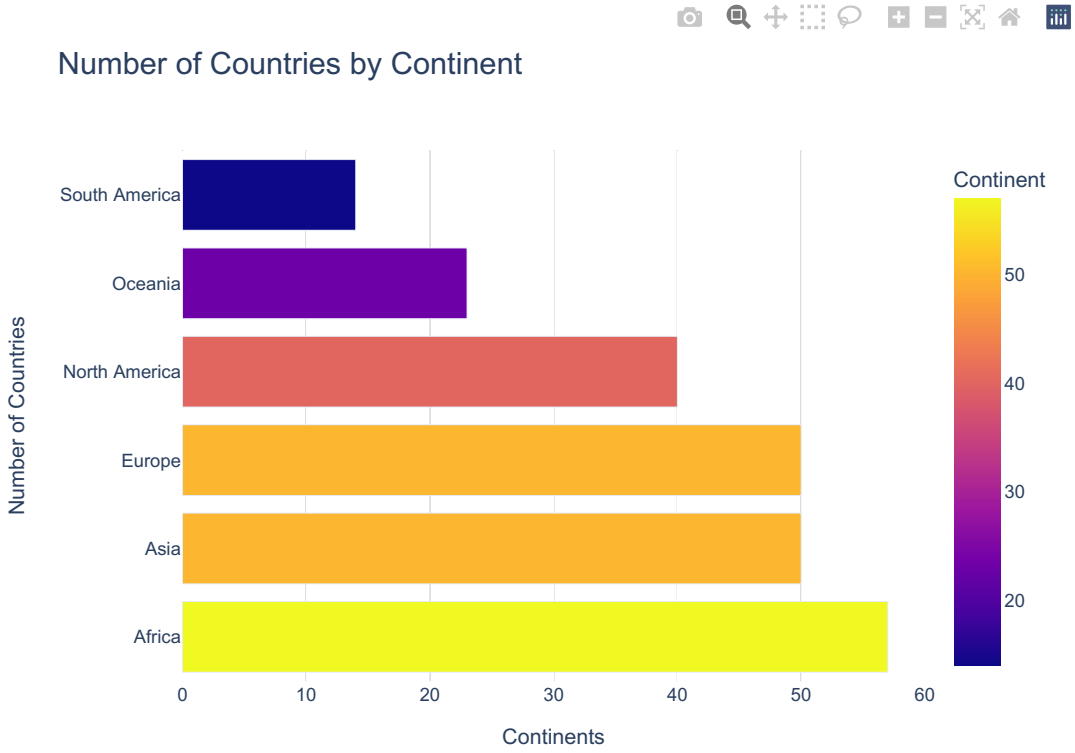
- Exploratory analysis revealed significant trends in population growth over the years. Key factors such as birth rate, death rate, and fertility rate were visualized to understand their impact on population changes.
- Summary statistics
- Visualization of population trends
- Analysis of key factors affecting population growth

In [15]:
```
custom_palette= ['#0b3d91', '#e0f7fa', '#228b22', '#1e90ff', '#8B4513', '#D2691E','#DAA520', '#556B2F']
```

In [16]:
```
countries_by_continent=data['Continent'].value_counts().reset_index()
```

A Bar chart to Represent Number of Countries by Continent

In [17]:
```
fig = px.bar(countries_by_continent,
x='Continent',y='index',
color='Continent',
# text='countries_by_continent.values',
title='Number of Countries by Continent',
color_discrete_sequence=custom_palette
)
# Customize the layout
fig.update_layout(
xaxis_title='Continents',
yaxis_title='Number of Countries',
plot_bgcolor='rgba(0,0,0,0)', # Set the background color to transparent
font_family='Arial', # Set font family
title_font_size=20 # Set title font size
)
# Show the plot
fig.show()
```



In [18]:
```
continent_population_percentage = data.groupby('Continent')['World Population Percentage'].sum().reset_index()
```

In [46]:
```
## Create the pie chart
fig = go.Figure(data=[go.Pie(labels=continent_population_percentage['Continent'],
values=continent_population_percentage['World Population Percentage'])])
fig.show()
```

In [49]:
```
# Update layout
fig.update_layout(
title='World Population Percentage by Continent',
template='plotly',
paper_bgcolor='rgba(255,255,255,0)', # Set the paper background color to transparent
plot_bgcolor='rgba(255,255,255,0)' # Set the plot background color to transparent
)
# Update pie colors
fig.update_traces(marker=dict(colors=custom_palette, line=dict(color='#FFFFFF',
width=1)))
fig.show()
```

In [21]:

In [22]:
```
# Melt the DataFrame to have a long format
data_melted = data.melt(id_vars=['Continent'],
value_vars=['2022 Population', '2020 Population', '2015 Population',
'2010 Population', '2000 Population', '1990 Population',
'1980 Population', '1970 Population'],
var_name='Year',
value_name='Population')
```

In [23]:
```
# Convert 'Year' to a more suitable format
data_melted['Year'] = data_melted['Year'].str.split().str[0].astype(int)
```

In [24]:
```
# Aggregate population by continent and year
population_by_continent = data_melted.groupby(['Continent',
'Year']).sum().reset_index()
```

In [43]:
```
fig = px.line(population_by_continent, x='Year', y='Population', color='Continent',
title='Population Trends by Continent Over Time',
```

```
labels={'Population': 'Population', 'Year': 'Year'},
color_discrete_sequence=custom_palette)

fig.update_layout(
template='plotly_white',
xaxis_title='Year',
yaxis_title='Population',
font_family='Arial',
title_font_size=20,
)
fig.update_traces(line=dict(width=3))
#Show the plot
fig.show()
```

World Population Comparison: 1970 to 2020

In [26]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 15 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Rank                         234 non-null    int64
 1   Country/Territory            234 non-null    object
 2   Continent                    234 non-null    object
 3   2022 Population              234 non-null    int64
 4   2020 Population              234 non-null    int64
 5   2015 Population              234 non-null    int64
 6   2010 Population              234 non-null    int64
 7   2000 Population              234 non-null    int64
 8   1990 Population              234 non-null    int64
 9   1980 Population              234 non-null    int64
 10  1970 Population              234 non-null    int64
 11  Area (km²)                   234 non-null    int64
 12  Density (per km²)            234 non-null    float64
 13  Growth Rate                  234 non-null    float64
 14  World Population Percentage  234 non-null    float64
dtypes: float64(3), int64(10), object(2)
memory usage: 27.5+ KB
```

In [42]:
```
features=['1970 Population' ,'2020 Population']
for feature in features:
    fig = px.choropleth(data,
locations='Country/Territory',
locationmode='country names',
color=feature,
hover_name='Country/Territory',
template='plotly_white',
title = feature)
#Show the plot
fig.show()
```

In [28]:
```
growth = (data.groupby(by='Country/Territory')['2022 Population'].sum()-data.groupby(by='Country/Territory')['1
```

In [41]:
```
fig=px.bar(x=growth.index,
y=growth.values,
text=growth.values,
color=growth.values,
title='Growth Of Population From 1970 to 2020 (Top 8)',
template='plotly_white')
fig.update_layout(xaxis_title='Country',yaxis_title='Population Growth')
#Show the plot
fig.show()
```

In [30]:
```
top_8_populated_countries_1970 = data.groupby('Country/Territory')['1970 Population'].sum().sort_values(ascendi
top_8_populated_countries_2022 = data.groupby('Country/Territory')['2022 Population'].sum().sort_values(ascendi
```

In [40]:
```
features = {'top_8_populated_countries_1970': top_8_populated_countries_1970,
'top_8_populated_countries_2022': top_8_populated_countries_2022}
for feature_name, feature_data in features.items():
    year = feature_name.split('_')[-1] # Extract the year from the feature name
fig = px.bar(x=feature_data.index,
```

```
y=feature_data.values,
text=feature_data.values,
color=feature_data.values,
title=f'Top 8 Most Populated Countries ({year})',
template='plotly_white')
fig.update_layout(xaxis_title='Country',
yaxis_title='Population Growth')
#Show the plot
fig.show()
```

In [32]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [33]:
```
data.head()
```

Out[33]:

	Rank	Country/Territory	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population	Are (km
0	36	Afghanistan	Asia	41128771	38972230	33753499	28189672	19542982	10694796	12486631	10752971	65223
1	138	Albania	Europe	2842321	2866849	2882481	2913399	3182021	3295066	2941651	2324731	2874
2	34	Algeria	Africa	44903225	43451666	39543154	35856344	30774621	25518074	18739378	13795915	238174
3	213	American Samoa	Oceania	44273	46189	51368	54849	58230	47818	32886	27075	19
4	203	Andorra	Europe	79824	77700	71746	71519	66097	53569	35611	19860	46

In [34]:
```
#Feature Engineering
# Create additional features if necessary (e.g., population growth rate)
data['GrowthRate'] = data['World Population Percentage'].pct_change() * 100
data = data.dropna()
```

In [35]:
```
# Define features and target variable
features = ['2022 Population','2020 Population','2015 Population','2010 Population','2000 Population','1990 Pop
X = data[features]
y = data['World Population Percentage']
```

In [36]:
```
# Splitting the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

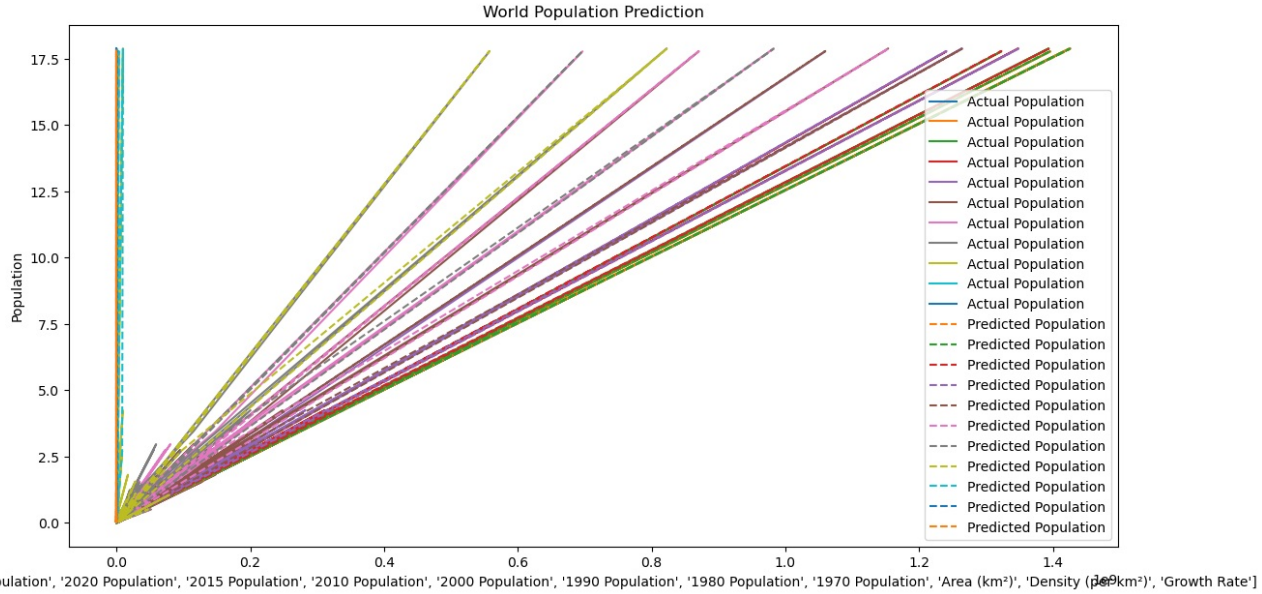
In [37]:
```
# Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [38]:
```
# Model Building
# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

```
# Predict on the test set
y_pred = model.predict(X_test_scaled)
# Model Evaluation
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R^2 Score:", r2_score(y_test, y_pred))
```

```
Mean Squared Error: 1.1189874098794918e-05
R^2 Score: 0.9999988140357875
```

In [39]:
```
# Visualization of Results
plt.figure(figsize=(14,7))
plt.plot(data[features], data['World Population Percentage'], label='Actual Population')
plt.plot(X_test[features], y_pred, label='Predicted Population', linestyle='--')
plt.xlabel(features)
plt.ylabel('Population')
plt.title('World Population Prediction')
plt.legend()
plt.show()
```



['2022 Population', '2020 Population', '2015 Population', '2010 Population', '2000 Population', '1990 Population', '1980 Population', '1970 Population', 'Area (km²)', 'Density (per km²)', 'Growth Rate']

In [ ]: