Project Overview

This project is a Streamlit-based interactive dashboard that allows users to:

- Explore historical life expectancy trends (1990-2019).
- Predict future life expectancy (2025-2100) using Polynomial Regression.
- Compare life expectancy trends between two countries.
- Rank countries based on life expectancy for any selected year (1990-2040).
- Visualize life expectancy on an interactive world map.

Step-by-Step Code Execution

1. Importing Required Libraries

```
import streamlit as st
import pandas as pd
import plotly.express as px
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
```

This imports:

- pandas for data handling
- numpy for numerical operations
- plotly.express for interactive visualizations
- streamlit for building the web dashboard
- sklearn for Machine Learning (Polynomial Regression)

2. Loading and Preprocessing the Dataset

```
df = pd.read_csv("dataset.csv")
```

- Loads the dataset containing life expectancy data (1990-2019).

```
df_melted = df.melt(id_vars=["Country", "Country_Code", "Level", "Region"],
var_name="Year", value_name="Life Expectancy")
```

- Converts the dataset from wide-format to long-format (so each row represents a country-year pair).

```
df_melted["Year"] = pd.to_numeric(df_melted["Year"], errors="coerce")
df_melted["Life Expectancy"] = pd.to_numeric(df_melted["Life Expectancy"],
errors="coerce")
```

- Converts Year and Life Expectancy columns into numeric format.

3. Building the Streamlit Dashboard

```
st.title("Human Life Expectancy Dashboard")
```

- Sets the title of the Streamlit web app.

Life Expectancy Trends

```
country = st.selectbox("Select a Country", df_melted["Country"].unique())
country_data = df_melted[df_melted["Country"] == country]
country_data = country_data.groupby("Year", as_index=False)["Life
Expectancy"].mean()
```

- Creates a dropdown to select a country.
- Filters dataset for the selected country.
- Averages life expectancy values for multiple records in the same year.

```
fig = px.line(country_data, x="Year", y="Life Expectancy", title=f"Life
Expectancy Trend in {country}")
st.plotly_chart(fig)
```

- Creates a line chart showing the life expectancy trend for the selected country.

4. Machine Learning Model for Predictions

Predicting Future Life Expectancy

```
if len(country_data) > 5:
    poly = PolynomialFeatures(degree=2)
    X = country_data[["Year"]]
    y = country_data["Life Expectancy"].dropna()
    X = X.loc[y.index]
    X_poly = poly.fit_transform(X)
    model = LinearRegression()
    model.fit(X_poly, y)
```

- Uses Polynomial Regression (Degree 2) to train a model for each country.
- Ensures that X and y have matching indices to prevent errors.

```
future_year = st.slider("Select a Future Year", min_value=2025, max_value=2100, value=2030, step=5)
future_pred = model.predict(poly.transform(pd.DataFrame([[future_year]], columns=["Year"])))
st.write(f"Predicted Life Expectancy in {future_year}: {future_pred[0]:.2f} years")
```

- Allows the user to select a future year (2025-2100).
- Predicts life expectancy using the trained model.

5. Ranking Countries by Life Expectancy

```
selected_year = st.slider("Select a Year", min_value=1990, max_value=2040, value=2019)
ranking_data = df_melted[df_melted["Year"] == selected_year].dropna().copy()
```

- Allows the user to select a year (1990-2040).
- Retrieves the life expectancy rankings for that year.

```
ranking_data = ranking_data.sort_values(by="Life Expectancy", ascending=False).reset_index(drop=True)
st.dataframe(ranking_data[["Country", "Life Expectancy"]].rename(columns={"Life Expectancy": f"Life Expectancy in {selected_year}"}))
```

- Sorts countries by life expectancy and displays the ranking.

6. Comparing Two Countries

```
country_1 = st.selectbox("Select First Country", df_melted["Country"].unique(), index=0)
country_2 = st.selectbox("Select Second Country", df_melted["Country"].unique(), index=1)
```

- Allows users to select two countries for comparison.

```
fig_compare = px.line(title="Life Expectancy Comparison")
fig_compare.add_scatter(x=data_1["Year"], y=data_1["Life Expectancy"],
mode="lines", name=country_1)
fig_compare.add_scatter(x=data_2["Year"], y=data_2["Life Expectancy"],
mode="lines", name=country_2)
st.plotly_chart(fig_compare)
```

- Creates a comparison chart showing life expectancy trends for both countries.

7. Interactive World Map

```
latest_year = df_melted["Year"].max()
latest_data = df_melted[df_melted["Year"] == latest_year].dropna()
fig_map = px.choropleth(
    latest_data,
    locations="Country_Code",
    color="Life Expectancy",
    hover_name="Country",
    title=f"Life Expectancy Around the World ({latest_year})",
    color_continuous_scale=px.colors.sequential.Plasma,
    projection="natural earth"
)
st.plotly_chart(fig_map)
```

- Displays a world map with life expectancy values for the latest year.

Summary of Project Flow

1. Load and preprocess the dataset (convert formats, handle missing values).
2. Create an interactive dashboard using Streamlit.
3. Visualize trends with line charts.
4. Train a Machine Learning model to predict future life expectancy.
5. Allow users to compare two countries and rank them.
6. Display global data with an interactive world map.
7. Deploy the app on Streamlit Cloud.