Untitled30

December 16, 2025

0.1 Import the Relevant Python Libraries

[1]:
```
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.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')
```

0.2 Set style for better visualizations

[2]:
```
plt.style.use('seaborn-v0_8-darkgrid')
sns.set_palette("husl")
```

0.3 Load the dataset

[4]:
```
df = pd.read_csv('youth_unemployment_global.csv')
```

0.4 Display basic information

[26]:
```
print("DATASET EXPLORATION - GLOBAL YOUTH UNEMPLOYMENT")
print("\n BASIC DATASET INFORMATION:")
print(f"Shape: {df.shape}")
print(f"Total Rows: {df.shape[0]}, Total Columns: {df.shape[1]}")

print("\n COLUMNS AND DATA TYPES:")
print(df.dtypes)

print("\n FIRST 5 ROWS:")
print(df.head())

print("\n MISSING VALUES:")
print(df.isnull().sum())
```

```
print("\n DESCRIPTIVE STATISTICS:")
print(df['YouthUnemployment'].describe())
```

```
DATASET EXPLORATION - GLOBAL YOUTH UNEMPLOYMENT

 BASIC DATASET INFORMATION:
Shape: (17290, 5)
Total Rows: 17290, Total Columns: 5

 COLUMNS AND DATA TYPES:
Country                       object
CountryCode                   object
Year                  datetime64[ns]
YouthUnemployment            float64
Year_Num                       int32
dtype: object

 FIRST 5 ROWS:
                       Country CountryCode       Year  YouthUnemployment  \
0  Africa Eastern and Southern          ZH 2024-01-01          13.283002
1  Africa Eastern and Southern          ZH 2023-01-01          13.367810
2  Africa Eastern and Southern          ZH 2022-01-01          13.620217
3  Africa Eastern and Southern          ZH 2021-01-01          14.955182
4  Africa Eastern and Southern          ZH 2020-01-01          14.997030

   Year_Num
0      2024
1      2023
2      2022
3      2021
4      2020

 MISSING VALUES:
Country                 0
CountryCode            65
Year                    0
YouthUnemployment    9309
Year_Num                0
dtype: int64

 DESCRIPTIVE STATISTICS:
count    7981.000000
mean       16.667732
std        11.567954
min         0.295000
25%         8.531000
50%        14.182000
75%        22.031000
```

```
max        82.409000
Name: YouthUnemployment, dtype: float64
```

0.5 Data Cleaning and Preparation

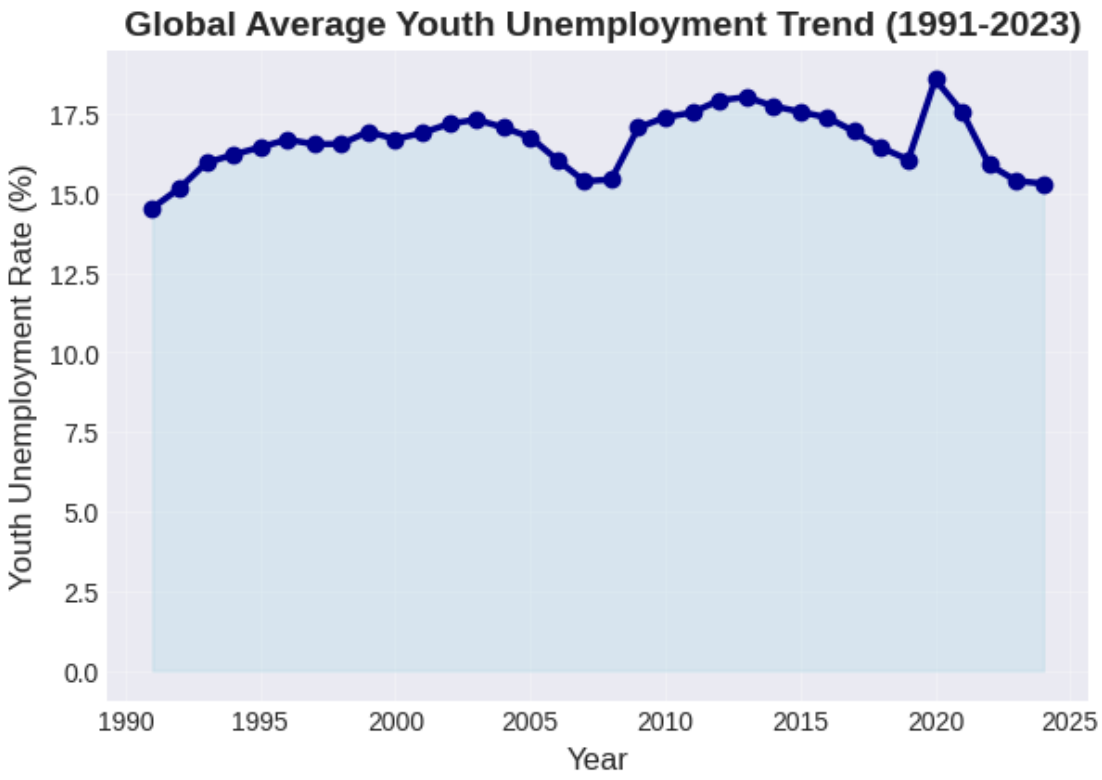
[8]:
```
print("\n DATA CLEANING PROCESS:")
# Convert Year to datetime if needed
df['Year'] = pd.to_datetime(df['Year'], format='%Y')
df['Year_Num'] = df['Year'].dt.year

# Handle missing values
initial_missing = df['YouthUnemployment'].isnull().sum()
df_clean = df.dropna(subset=['YouthUnemployment'])
print(f"Removed {initial_missing} rows with missing YouthUnemployment values")
print(f"Cleaned dataset shape: {df_clean.shape}")

# Exploratory Data Analysis
print("\n EXPLORATORY DATA ANALYSIS:")
```

```
 DATA CLEANING PROCESS:
Removed 9309 rows with missing YouthUnemployment values
Cleaned dataset shape: (7981, 5)

 EXPLORATORY DATA ANALYSIS:
```

0.6 1. Global Trends Over Time

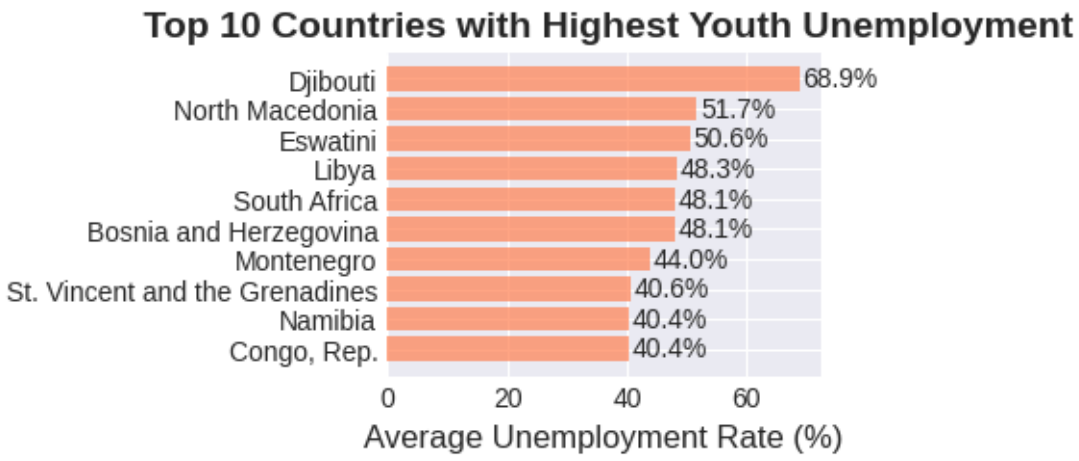
[9]:
```
plt.figure(figsize=(15, 10))

# Subplot 1: Global average trend
plt.subplot(2, 2, 1)
global_trend = df_clean.groupby('Year_Num')['YouthUnemployment'].mean().
 ↪reset_index()
plt.plot(global_trend['Year_Num'], global_trend['YouthUnemployment'],
         linewidth=2.5, color='darkblue', marker='o')
plt.title(' Global Average Youth Unemployment Trend (1991-2023)', fontsize=14,␣
 ↪fontweight='bold')
plt.xlabel('Year', fontsize=12)
plt.ylabel('Youth Unemployment Rate (%)', fontsize=12)
plt.grid(True, alpha=0.3)
plt.fill_between(global_trend['Year_Num'], global_trend['YouthUnemployment'],
                 alpha=0.3, color='lightblue')
```

[9]:
```
<matplotlib.collections.FillBetweenPolyCollection at 0x7f4bde5827b0>
```



0.7 Subplot 2: Top 10 countries with highest average unemployment

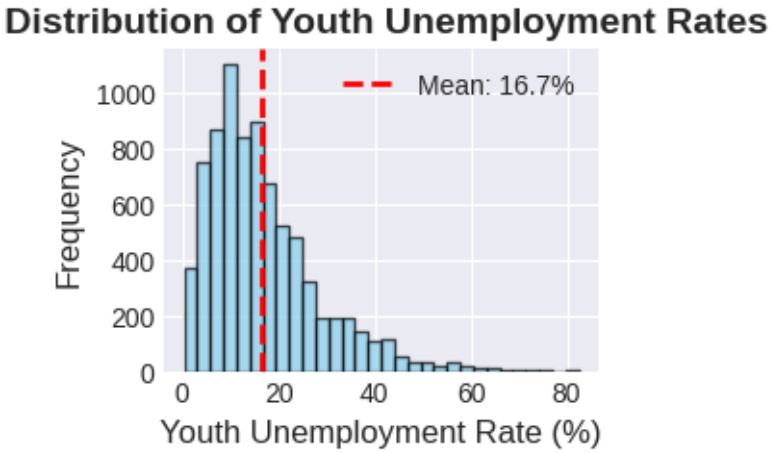
[10]:
```
plt.subplot(2, 2, 2)
top_countries = df_clean.groupby('Country')['YouthUnemployment'].mean().
 ↪nlargest(10).reset_index()
plt.barh(top_countries['Country'], top_countries['YouthUnemployment'],
         color='coral', alpha=0.7)
plt.title(' Top 10 Countries with Highest Youth Unemployment', fontsize=14,␣
 ↪fontweight='bold')
plt.xlabel('Average Unemployment Rate (%)', fontsize=12)
plt.gca().invert_yaxis()
for i, v in enumerate(top_countries['YouthUnemployment']):
    plt.text(v + 0.5, i, f'{v:.1f}%', va='center')
```



0.8 Subplot 3: Distribution of unemployment rates

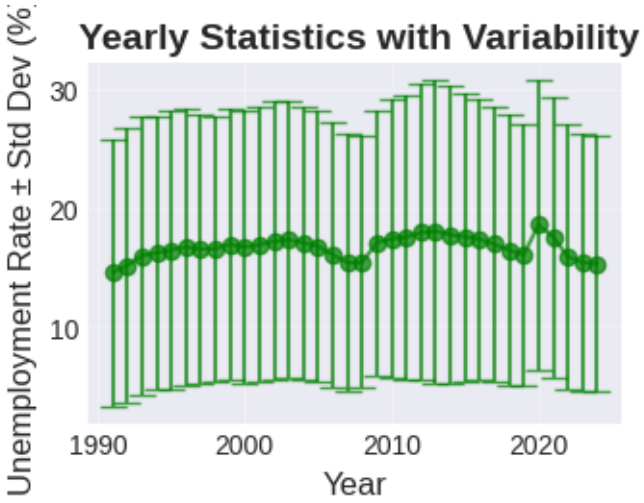
[11]:
```
plt.subplot(2, 2, 3)
plt.hist(df_clean['YouthUnemployment'], bins=30, edgecolor='black',
         alpha=0.7, color='skyblue')
plt.title(' Distribution of Youth Unemployment Rates', fontsize=14,␣
 ↪fontweight='bold')
plt.xlabel('Youth Unemployment Rate (%)', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.axvline(df_clean['YouthUnemployment'].mean(), color='red',
            linestyle='--', linewidth=2, label=f'Mean:␣
 ↪{df_clean["YouthUnemployment"].mean():.1f}%')
plt.legend()
```

[11]: <matplotlib.legend.Legend at 0x7f4bde582270>



0.9 Subplot 4: Year with highest unemployment

[12]:
```
plt.subplot(2, 2, 4)
yearly_stats = df_clean.groupby('Year_Num')['YouthUnemployment'].agg(['mean',␣
↪'std']).reset_index()
plt.errorbar(yearly_stats['Year_Num'], yearly_stats['mean'],
             yerr=yearly_stats['std'], fmt='o-', capsize=5,
             color='green', alpha=0.7)
plt.title(' Yearly Statistics with Variability', fontsize=14, fontweight='bold')
plt.xlabel('Year', fontsize=12)
plt.ylabel('Unemployment Rate ± Std Dev (%)', fontsize=12)
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

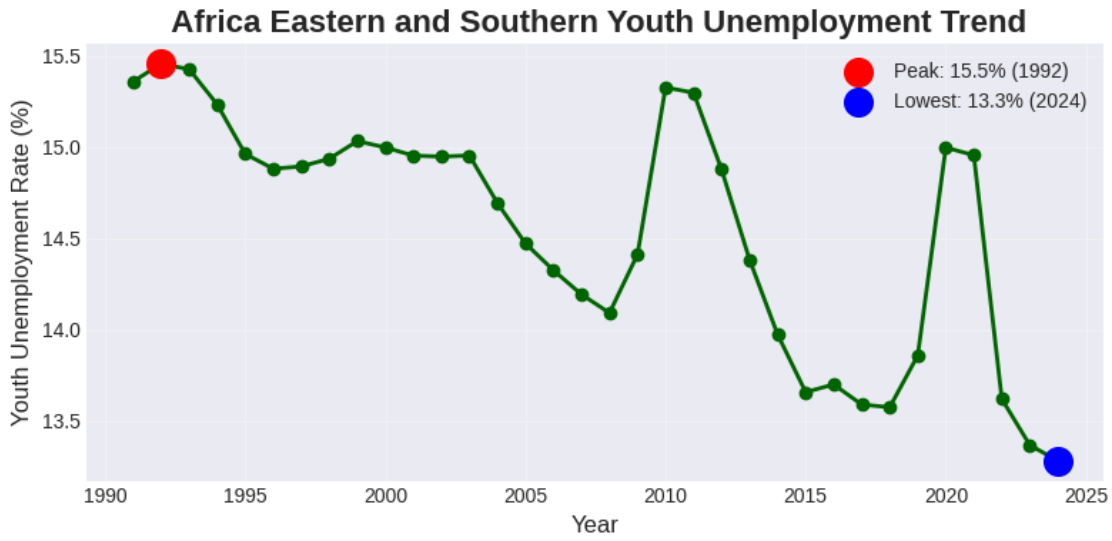


0.10 2. Regional Analysis (for Africa Eastern and Southern as shown in sample)

[16]:
```
print("\n REGIONAL ANALYSIS - AFRICA EASTERN AND SOUTHERN:")
africa_data = df_clean[df_clean['Country'] == 'Africa Eastern and Southern']

if not africa_data.empty:
    plt.figure(figsize=(8, 4))
    plt.plot(africa_data['Year_Num'], africa_data['YouthUnemployment'],
             marker='o', linewidth=2, color='darkgreen', markersize=6)
    plt.title(' Africa Eastern and Southern Youth Unemployment Trend',
              fontsize=16, fontweight='bold')
    plt.xlabel('Year', fontsize=12)
```

```
    plt.ylabel('Youth Unemployment Rate (%)', fontsize=12)
    plt.grid(True, alpha=0.3)

    # Highlight specific years
    max_year = africa_data.loc[africa_data['YouthUnemployment'].idxmax()]
    min_year = africa_data.loc[africa_data['YouthUnemployment'].idxmin()]

    plt.scatter(max_year['Year_Num'], max_year['YouthUnemployment'],
                color='red', s=200, zorder=5, label=f'Peak:␣
↪{max_year["YouthUnemployment"]:.1f}% ({int(max_year["Year_Num"])})')
    plt.scatter(min_year['Year_Num'], min_year['YouthUnemployment'],
                color='blue', s=200, zorder=5, label=f'Lowest:␣
↪{min_year["YouthUnemployment"]:.1f}% ({int(min_year["Year_Num"])})')

    plt.legend()
    plt.tight_layout()
    plt.show()
    # Statistical analysis for the region
    print(f"\n Statistics for Africa Eastern and Southern:")
    print(f"Time Period: {int(africa_data['Year_Num'].min())} -␣
↪{int(africa_data['Year_Num'].max())}")
    print(f"Mean Unemployment: {africa_data['YouthUnemployment'].mean():.2f}%")
    print(f"Highest Unemployment: {max_year['YouthUnemployment']:.2f}% in␣
↪{int(max_year['Year_Num'])}")
    print(f"Lowest Unemployment: {min_year['YouthUnemployment']:.2f}% in␣
↪{int(min_year['Year_Num'])}")
    print(f"Standard Deviation: {africa_data['YouthUnemployment'].std():.2f}%")
```

REGIONAL ANALYSIS - AFRICA EASTERN AND SOUTHERN:



```
 Statistics for Africa Eastern and Southern:
Time Period: 1991 - 2024
Mean Unemployment: 14.55%
Highest Unemployment: 15.46% in 1992
Lowest Unemployment: 13.28% in 2024
Standard Deviation: 0.66%
```

0.11 Machine Learning: Forecasting with Random Forest

[27]:
```
print(" MACHINE LEARNING: UNEMPLOYMENT FORECASTING")

# Prepare data for ML
ml_df = df_clean.copy()

# Encode categorical variables
le_country = LabelEncoder()
ml_df['Country_Encoded'] = le_country.fit_transform(ml_df['Country'])

# Create lag features for time series
ml_df['Unemployment_Lag1'] = ml_df.groupby('Country')['YouthUnemployment'].
 ↪shift(1)
ml_df['Unemployment_Lag2'] = ml_df.groupby('Country')['YouthUnemployment'].
 ↪shift(2)
ml_df['Unemployment_Lag3'] = ml_df.groupby('Country')['YouthUnemployment'].
 ↪shift(3)

# Drop rows with NaN in lag features
ml_df = ml_df.dropna()

# Features and target
X = ml_df[['Year_Num', 'Country_Encoded', 'Unemployment_Lag1',
           'Unemployment_Lag2', 'Unemployment_Lag3']]
y = ml_df['YouthUnemployment']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

print(f"\n MACHINE LEARNING DATASET:")
print(f"Training samples: {X_train.shape[0]}")
print(f"Testing samples: {X_test.shape[0]}")
print(f"Features used: {X_train.shape[1]}")
```

```
MACHINE LEARNING: UNEMPLOYMENT FORECASTING
```

```
 MACHINE LEARNING DATASET:
Training samples: 5796
Testing samples: 1449
Features used: 5
```

0.12 Train Random Forest model

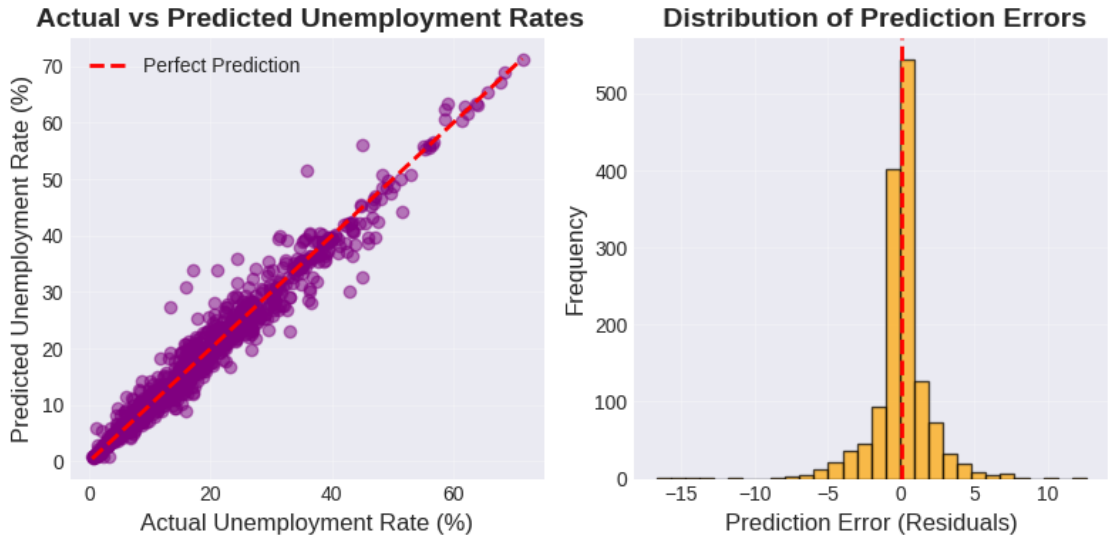
[20]:
```python
rf_model = RandomForestRegressor(
    n_estimators=100,
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=2,
    random_state=42
)

rf_model.fit(X_train, y_train)

# Make predictions
y_pred = rf_model.predict(X_test)

# Evaluate model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\n MODEL PERFORMANCE METRICS:")
print(f"Mean Absolute Error (MAE): {mae:.4f}")
print(f"Mean Squared Error (MSE): {mse:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
print(f"R² Score: {r2:.4f}")

# Feature Importance
feature_importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': rf_model.feature_importances_
}).sort_values('Importance', ascending=False)

print("\n FEATURE IMPORTANCE:")
print(feature_importance)
```

```
 MODEL PERFORMANCE METRICS:
Mean Absolute Error (MAE): 1.1882
Mean Squared Error (MSE): 4.3885
Root Mean Squared Error (RMSE): 2.0949
R² Score: 0.9665
```

```
 FEATURE IMPORTANCE:
             Feature  Importance
2  Unemployment_Lag1    0.982041
3  Unemployment_Lag2    0.006188
4  Unemployment_Lag3    0.006034
0           Year_Num    0.003397
1    Country_Encoded    0.002340
```

0.13 Visualization of predictions vs actual

[21]:
```
plt.figure(figsize=(8, 4))

plt.subplot(1, 2, 1)
plt.scatter(y_test, y_pred, alpha=0.5, color='purple')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
         'r--', lw=2, label='Perfect Prediction')
plt.xlabel('Actual Unemployment Rate (%)', fontsize=12)
plt.ylabel('Predicted Unemployment Rate (%)', fontsize=12)
plt.title(' Actual vs Predicted Unemployment Rates', fontsize=14,␣
 ↪fontweight='bold')
plt.legend()
plt.grid(True, alpha=0.3)

plt.subplot(1, 2, 2)
residuals = y_test - y_pred
plt.hist(residuals, bins=30, edgecolor='black', alpha=0.7, color='orange')
plt.axvline(x=0, color='red', linestyle='--', linewidth=2)
plt.xlabel('Prediction Error (Residuals)', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title(' Distribution of Prediction Errors', fontsize=14, fontweight='bold')
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

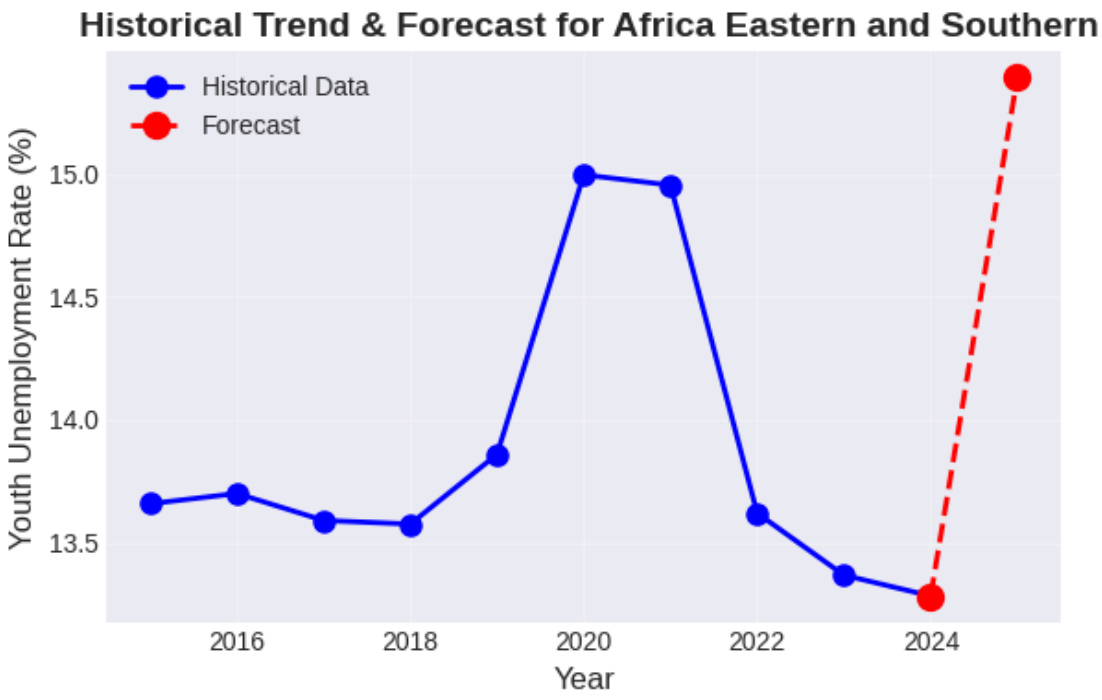


0.14 Forecasting future unemployment (example for a specific country)

[24]:
```
print("\n FUTURE FORECASTING EXAMPLE:")

# Select a specific country for forecasting demonstration
sample_country = 'Africa Eastern and Southern'
country_data = df_clean[df_clean['Country'] == sample_country].
 ↪sort_values('Year_Num')

if not country_data.empty:
    # Prepare data for the specific country
    country_code = le_country.transform([sample_country])[0]

    # Use last 3 years for prediction
    recent_data = country_data.tail(3)['YouthUnemployment'].values

    if len(recent_data) == 3:
        # Create feature vector for next year
        next_year = country_data['Year_Num'].max() + 1
        features = np.array([[next_year, country_code,
                              recent_data[2], recent_data[1], recent_data[0]]])

        # Predict next year's unemployment
        prediction = rf_model.predict(features)[0]

        print(f"Country: {sample_country}")
        print(f"Last available year: {country_data['Year_Num'].max()}")
        print(f"Last 3 years unemployment: {recent_data}")
```

```
        print(f"Predicted unemployment for {next_year}: {prediction:.2f}%")

        # Visualize historical data and prediction
        plt.figure(figsize=(6, 4))
        years = list(country_data['Year_Num'].tail(10)) + [next_year]
        unemployment = list(country_data['YouthUnemployment'].tail(10)) +␣
↪[prediction]

        plt.plot(years[:-1], unemployment[:-1], 'o-', linewidth=2,
                 markersize=8, color='blue', label='Historical Data')
        plt.plot(years[-2:], unemployment[-2:], 'o--', linewidth=2,
                 markersize=10, color='red', label='Forecast')

        plt.title(f' Historical Trend & Forecast for {sample_country}',
                  fontsize=14, fontweight='bold')
        plt.xlabel('Year', fontsize=12)
        plt.ylabel('Youth Unemployment Rate (%)', fontsize=12)
        plt.legend()
        plt.grid(True, alpha=0.3)
        plt.tight_layout()
        plt.show()
```

```
 FUTURE FORECASTING EXAMPLE:
Country: Africa Eastern and Southern
Last available year: 2024
Last 3 years unemployment: [13.62021704 13.36780956 13.2830017 ]
Predicted unemployment for 2025: 15.39%
```


Historical Trend & Forecast for Africa Eastern and Southern
Historical Data
Forecast
Youth Unemployment Rate (%)
15.0
14.5
14.0
13.5
2016
2018
2020
2022
2024
Year

[ ]: