Population Time Series Forecasting (PROPHET MODEL)

Understanding population dynamics is crucial for urban planning, resource allocation, and policymaking. Time series analysis of population data allows us to track demographic trends, forecast future population growth, and identify patterns that can inform decision-making processes. Cities need to anticipate future population trends to plan for housing, transportation, and urban infrastructure development. Population forecasting is crucial for governments and policymakers to plan for infrastructure, healthcare, education, and other essential services. Healthcare providers use population forecasts to estimate future healthcare demands, such as hospital beds, healthcare personnel, and medical supplies.

This project uses time series data analysis techniques to analyze population trends and provide insights into demographic changes over time.

Motivation:

Population dynamics are influenced by a wide range of factors including birth rates, migration patterns, economic conditions, government policies, social trends, and cultural shifts. The interplay of these complex and interconnected factors can lead to volatile and unpredictable population trends over time. Economic instability, social unrest, political upheavals, natural disasters, and other external events can disrupt population dynamics and lead to sudden changes in population trends. Demographic trends such as aging populations, changing family structures, urbanization, and immigration can contribute to volatility in population series forecasting.

Keeping in mind all these factors, initially, we used ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), and exponential smoothing for time series forecasting however, they differ in their underlying assumptions, modeling approach, and strengths. Here's a comparative overview:

Strengths & Challenges of Models Used:

ARIMA (AutoRegressive Integrated Moving Average):

Modeling Approach: ARIMA models capture linear relationships in the time series by incorporating autoregressive (AR), differencing (I), and moving average (MA) components. SARIMA extends ARIMA by incorporating seasonal components.

Strengths: Suitable for stationary time series with linear trends and autocorrelation. ARIMA models are flexible and can capture a wide range of temporal patterns. **SARIMA** further extends this capability to seasonal data.

Exponential Smoothing:

Modeling Approach: Exponential smoothing methods, such as Simple Exponential Smoothing (SES), Double Exponential Smoothing (Holt's method), and Triple Exponential Smoothing (Holt-Winters method), use weighted averages of past observations to forecast future values.

Strengths: Simplicity and computational efficiency. Exponential smoothing methods are easy to implement and understand. They are suitable for short-term forecasting and work well with data exhibiting trend and/or seasonality.

Based on the MSE, and RMSE results of the above three models I have explored another model. Since, all these models gave very high MSE and RMSE scores due to the volatility and non-linearity of the dataset.

Prophet:

Modeling Approach: Prophet uses an additive model that decomposes the time series into trend, seasonal components (yearly, weekly, and daily), holiday effects, and error terms. It employs a Bayesian framework and utilizes Markov Chain Monte Carlo (MCMC) for parameter estimation.

Strengths: here are the following key factors which also addressed the challenges faced in above models:

Flexibility and Ease of Use: Prophet is designed to be user-friendly and requires minimal data preprocessing.
Automatic Handling of Seasonality: Prophet incorporates sophisticated algorithms to automatically detect and model various seasonal patterns in the data, including daily, weekly, and yearly seasonality. This eliminates the need for manual feature engineering and simplifies the forecasting process.

Handling of Holidays and Special Events: Prophet allows users to specify holidays and other special events that may impact the time series data. It includes these events as additional regressors in the model, enabling more accurate forecasts around these specific time periods.

Robustness to Missing Data and Outliers: Prophet is designed to handle missing data and outliers gracefully. It employs robust statistical techniques to impute missing values and mitigate the effects of outliers, ensuring more robust and reliable forecasts.

Scalability: Prophet is highly scalable and can handle large datasets efficiently. It leverages parallelization and distributed computing techniques to speed up the model-fitting process, making it suitable for analyzing large volumes of time-series data.

Interpretability: Prophet provides interpretable forecasts by decomposing the time series into trend, seasonality, and holiday effects. This decomposition allows users to understand the underlying patterns driving the forecasts and gain insights into the factors influencing future outcomes.

Uncertainty Estimation: Prophet quantifies forecast uncertainty by generating prediction intervals around the point forecasts. These prediction intervals provide a measure of the uncertainty associated with the forecasts, enabling users to make informed decisions based on their level of confidence.

In general the choice between Prophet, ARIMA/SARIMA, and exponential smoothing depends on the characteristics of the time series data and the specific requirements of the forecasting task. Prophet offers robustness to missing data, handles multiple seasonalities and holidays, and provides an intuitive interpretation of forecasts. ARIMA/SARIMA are well-suited for linear modeling of stationary time series with autocorrelation, while exponential smoothing methods are simple and efficient, making them suitable for short-term forecasting and datasets with trend and/or seasonality.

Prophet Model:

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data.

In Prophet, for example, the additive model assumes that the observed time series $y(t)$ can be represented as the sum of several components:

$$y(t)=g(t)+s(t)+h(t)+\epsilon t$$

Where:

- $g(t)$ represents the trend component, capturing the overall direction of the data over time.
- $s(t)$ represents the seasonal component, capturing recurring patterns or seasonality.
- $h(t)$ represents the holiday effects, accounting for one-off events or holidays that might impact the time series.
- ϵt represents the error term, accounting for random fluctuations or noise in the data.

By modeling each of these components separately and then combining them additively, the additive model aims to provide a more flexible and interpretable framework for time series forecasting.

Visualization & Results:

Without removing stationarity:

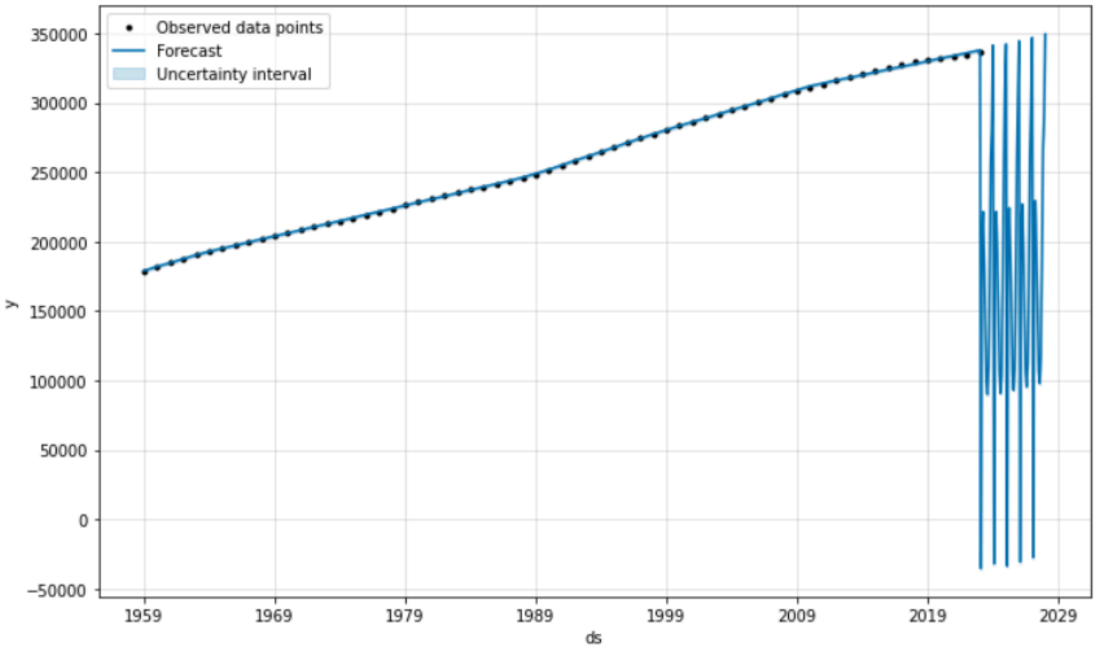
We have a trend that shows the long-term growth, shrink, or stagnancy of the data, trend_lower, and trend_upper is the uncertainty levels.

```
future = m.make_future_dataframe(periods=12 * 5,
                                  freq='M')
```

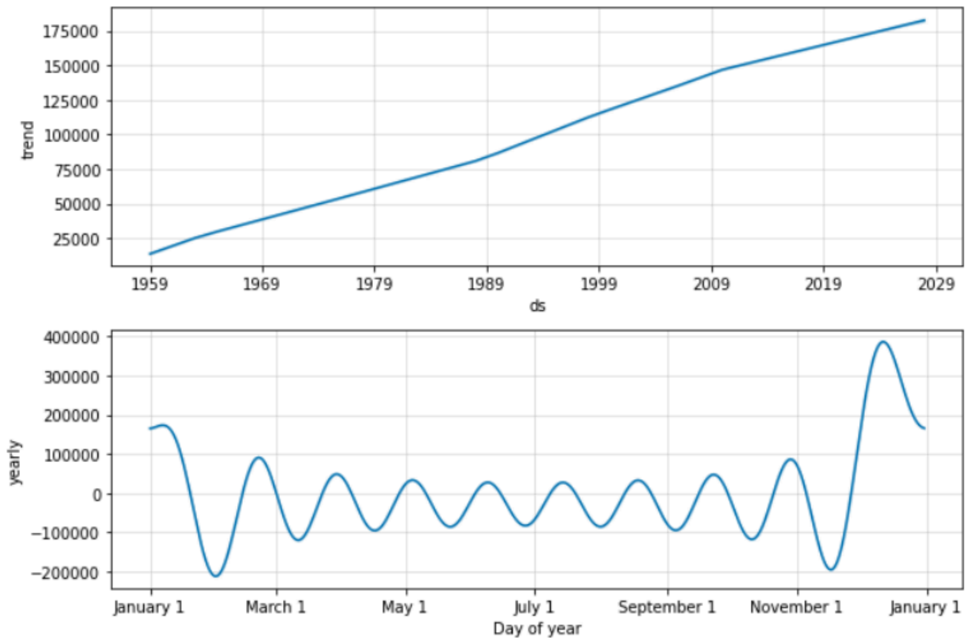
```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower',
          'yhat_upper', 'trend',
          'trend_lower', 'trend_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper	trend	trend_lower	trend_upper
120	2027-08-31	117852.895916	116773.200102	118919.435014	181954.757483	181264.167604	182576.471879
121	2027-09-30	183596.830838	182543.023131	184638.038156	182117.793019	181410.197360	182757.423526
122	2027-10-31	266055.229924	264943.039116	267202.380345	182286.263072	181543.101810	182944.737471
123	2027-11-30	289982.562826	288803.173846	291108.654064	182449.298608	181688.923271	183130.790671
124	2027-12-31	349359.371850	348219.259144	350403.376344	182617.768661	181836.183397	183316.787291

```
fig1 = m.plot(forecast, include_legend=True)
```



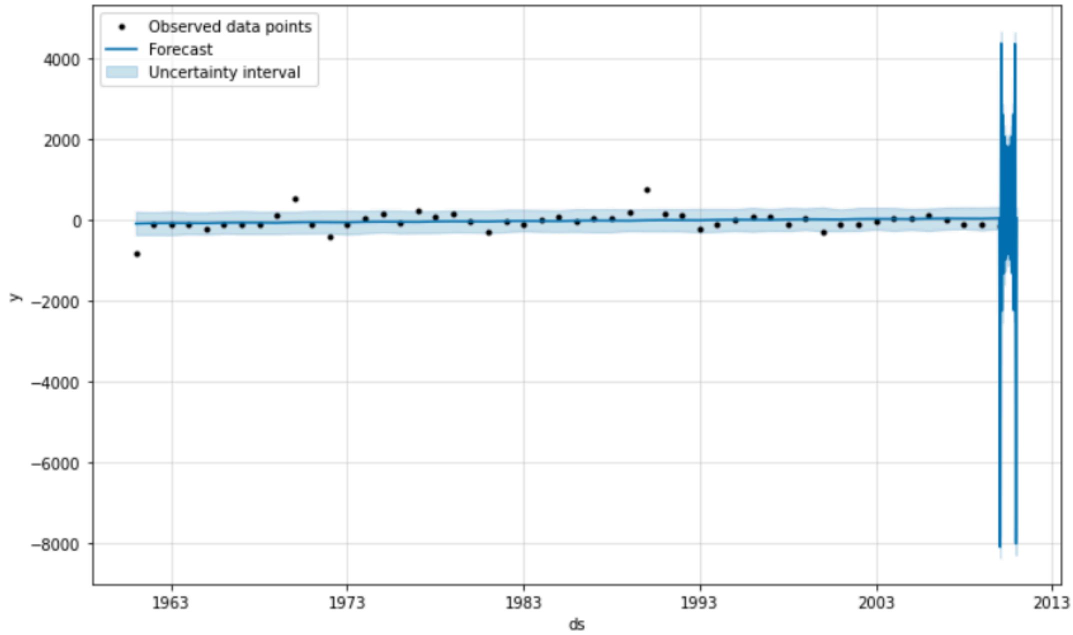

```
fig2 = m.plot_components(forecast)
```



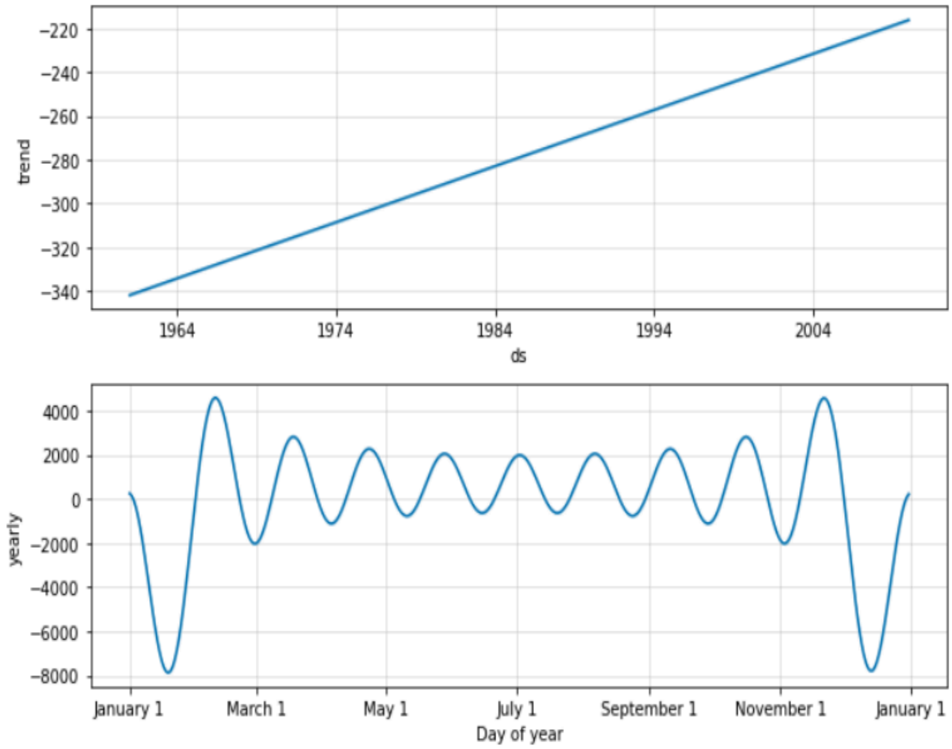
Removing Stationarity:

The below images show the trends and seasonality (in a year) of the time series data. We can see there is an increasing trend, meaning the **population has increased over time**. If we look at the seasonality graph, we can see that **December and January is the time with the most increase observed** in a given year.

```
fig1 = model.plot(forecast, include_legend=True)
```




```
fig2 = model.plot_components(forecast)
```



Key takeaways:

To compare the results of the Prophet model with those of other models, it's important to have the performance metrics (MAE, MSE, RMSE, and MAPE) for the other models. Lower values of MAE, MSE, RMSE, and MAPE indicate better performance.

Prophet Model:

- Root Mean Squared Error (RMSE): 3591.63

Other 3 Model Comparisons:

- ARIMA: RMSE = 2.614149e+07

- SARIMA: RMSE = 2.297991e+06

- Exponential Smoothing: RMSE = 1.012942e+06

- The Prophet model has the lowest RMSE value among all the models, indicating better performance in terms of accuracy.
- Exponential Smoothing has the second lowest RMSE value, followed by SARIMA and ARIMA, respectively.
- These results suggest that the Prophet model outperforms the other models in terms of forecasting accuracy based on RMSE

Insights or recommendations applicable in real-world scenarios:

- Identifying Trends and Patterns
- Resource Allocation and Infrastructure Planning
- Social Services and Public Policy
- Business Strategy and Market Segmentation
- Environmental Sustainability
- Emergency Preparedness and Disaster Management
- Investment and Economic Development
- Healthcare Planning and Disease Prevention

How the project Could be Improved?

Using Alternative Models:

- LSTM
- GBM (Gradient Boosting)
- State Space Models (DLM)
- CNN
- ARIMAX
- Ensemble of Different Forecasting Models
- Auto Encoders
- GANs
- Bayesian VAR(Vector Auto Regression)

Improvement of Existing Models:

Using Hybrid Models:

- **ARIMA with Exogenous Variables (ARIMAX):** Extends ARIMA by incorporating additional exogenous variables that may influence the time series.

- **Prophet with Additional Regressors:** Extends Prophet by including additional regressors that capture external factors influencing the time series**.**

Future Research Direction: The following directions represent opportunities for advancing population forecasting methods and addressing challenges in understanding and predicting population dynamic.

Hierarchical and Spatial Modeling: Develop models that can capture population dynamics at different geographic levels and incorporate spatial dependencies between regions.

Dynamic Population Modeling: Explore approaches that capture the effects of evolving social, economic, and environmental factors on population dynamics.

Integration of Multiple Data Sources: Investigate methods for integrating diverse data sources to improve the accuracy and timeliness of population forecasts.

Uncertainty Quantification and Risk Assessment: Develop techniques for quantifying and propagating uncertainty in population forecasts to support decision-making under conditions of uncertainty.

Long-Term Population Forecasting: Extend forecasting horizons to include long-term scenarios and projections, considering demographic trends over multiple decades or centuries.

Multi-Domain Integration: Explore interdisciplinary approaches that integrate population forecasting with other domains such as urban planning, healthcare, and environmental sustainability.

Ethical and Social Implications: Address ethical, social, and policy implications of population forecasting, ensuring transparency, fairness, and accountability.

Stakeholder Engagement and Co-Creation: Involve stakeholders in the design, development, and evaluation of population forecasting models to reflect diverse perspectives and priorities.

Real-Time Forecasting and Decision Support: Develop real-time forecasting systems to provide timely information for decision-making in areas like healthcare, education, and infrastructure.

Evaluation and Benchmarking: Establish standardized evaluation metrics and benchmarks to compare the performance of different forecasting methods and tools.