CONTENTS

INTRODUCTION

The datasets chosen for this exercise is from the World Bank specifically their World Development Indicator Data[1], Gender Statistics Data[2] and their Debt Statistics Data[3]. These datasets cover a huge range of socioeconomic indicators for countries at the macro level and are in the form of a time series from the 60's to the present time.

The aim was then to explore the relationship between Fertility Rates and wider macroeconomic indicators. Also I wanted to determine the most important feature set from the attributes calculated from that would enable the prediction of the median fertility rate for countries. The reason that medianmeasure is chosen is because it is robust to outliers and wold therefore serve to reduce noise in the target value in our predictive modelling.

The motivation for exploring and predicting fertility rates is because this is a good indicator of demographic changes and potential problems related to it. So low fertility rates in countries such as Japan are well known andhave caused a situation where there is a rapidly greying population with insufficient replacement leading to huge pressure on the economy, health system and by extension government expenditure. Such an inverse pyramid leads to questions such as how will care for such populations be funded? Where will the workers of the future come from? If there are insufficient workers what sort of impact would this have on the wider economy? Although the later questions are interesting but are not addressed here. The underlying cause for such questions such asthe fertility rate are explored. As it would help to understand and thus enable for appropriate policy decisions tobe made. The data processing flow is outlined in **Table 2.**

However, most of the time there are a lot of blanks in the data and the key challenge firstly was to pick interesting indicators that had sufficient amount of data. Initial exploration suggested that considering the time range from2000 – 2013 would yield the most complete data for the set of chosen indicators shown in **Table 1** below.

The way this was done was by extracting each indicator separately from the source files into Python and then calculating the percentage of blank to non-blank values. The indicators chosen have typically less than 10% of the data missing. This was the most acceptable compromise as most other indicators had missing information in the range of

80% for the given time range. Thus considering such variable would introduce a high degree of biastowards variables that had more data and hence would not need to be filled with zeros. Also filling such variableswith a large number of zeros would not lead to useful features.

Table 1: Selected indicators for analysis and source

Indicator Name	Source
Fertility Rate	Gender Statistics
Total Percentage of Labour Force that is female	Gender Statistics
GDP (current US$)	Word Development Indicators
GDP (current US$)	Word Development Indicators
GDP per capita, PPP (current international $)	Word Development Indicators
Health expenditure, total (% of GDP)	Word Development Indicators
Consumer price index (2010 = 100) (CPI)	Word Development Indicators
Population, total	Word Development Indicators
Gross National Income (GNI) (current US$)	Word Development Indicators
Total debt service (% of exports of goods, services and primaryincome) ->(Total Debt)	Word Development Indicators/DebtStatistics

The IMF databases mentioned collect over 480 indicators from all countries and not all of them are relevant forthis purpose. There was a preference for broad macroeconomic and social indicators which are easy to understand without much domain knowledge.

I wanted to investigate whether using summary statistics can be used in place of the whole time series and only a few of these variables can then be used to predict our value of interest then we have successfully reduced ourpotentially big data problem to a smaller data problem.

A few months ago, I stumbled upon an interesting article about how a country's shrinking population size affects its economy and the ability to generate wealth. Greece caught my attention, as the article mentioned the challenge of obtaining reliable statistical data to conduct a proper analysis. Another article highlighted the country's declining fertility rates. This sparked an idea in my mind to investigate the data provided by the Hellenic Statistical Authority (ELSTAT) which is the official statistic agency in Greece and check for consistency while seeking additional data resources from UNdata and Worldbank websites to enrich the analysis.

As I delved deeper into the data, I was confronted with several questions:

- Is it easy to get accurate data on birthrates and population?
- How can we make sure that the data we have is of good quality?
- What are some effective ways to display birthrate and population data over a period of time?
- Using machine learning algorithms, can we predict birthrate and population growth? If so, which technique is the best for this purpose?

PROJECT OVERVIEW

The scope of this project is to use Python programming to conduct an analysis on Greece’s population, utilizing data from multiple resources and addressing key questions related to the reliability and quality of the data obtained.

By implementing a range of machine learning models, the project aims to evaluate and compare their accuracy in predicting population growth over the next 30 years. Throughout my work, I hope to provide actionable recommendations and generate new insights that various stakeholders can make use of to tackle issues associated with population growth.

DATA

For this project I combined multiple datasets, overcoming the limitations of using a single dataset. In that way, I was able to cover a broader time period and draw analysis that would provide a more comprehensive view of population, enabling to explore different aspects such as population growth, gender, birthrate, stillbirths and births per month.

The first dataset used, was from the Hellenic Statistical Authority (ELSTAT) and specifically the «01. Births-Absolute number and rates (1932–2021)» dataset. This dataset contains information on the population of Greece broken down by gender, live births and stillbirths as well for years 1932–2021.

As I sought to augment my understanding of population growth, I explored another ELSTAT dataset that contained information on total population over the period from 2001 to 2021. After examining the dataset, it proved to be limited in its scope, particularly when considering the prospect of training a machine learning model.

Given the limitations of the aforementioned dataset, I needed to seek out a more extensive and detailed source of information. This research led me to the WORLDBANK website, were a more extensive dataset discovered including data from a longer time frame, spanning from 1960 to 2021.

At last, the project was enriched by another dataset provided by the UNdata website. There was information about births per month, enabling to perform more complex analysis and create visualizations that helped to better understand the underlying trends and patterns in population growth.

DATA CLEANING AND PREPROCESSING

Before proceeding further with the analysis, it is important to write a few things about data cleaning. When I started my data science journey, every article I read and every professional I was talking to, were pointing out the significance of data cleaning. Forbes magazine mentions that almost 80% of time is being spent in data cleaning. In this particular analysis, the statement referred to earlier was found to be accurate, as a significant amount of time was dedicated to data cleaning and wrangling.

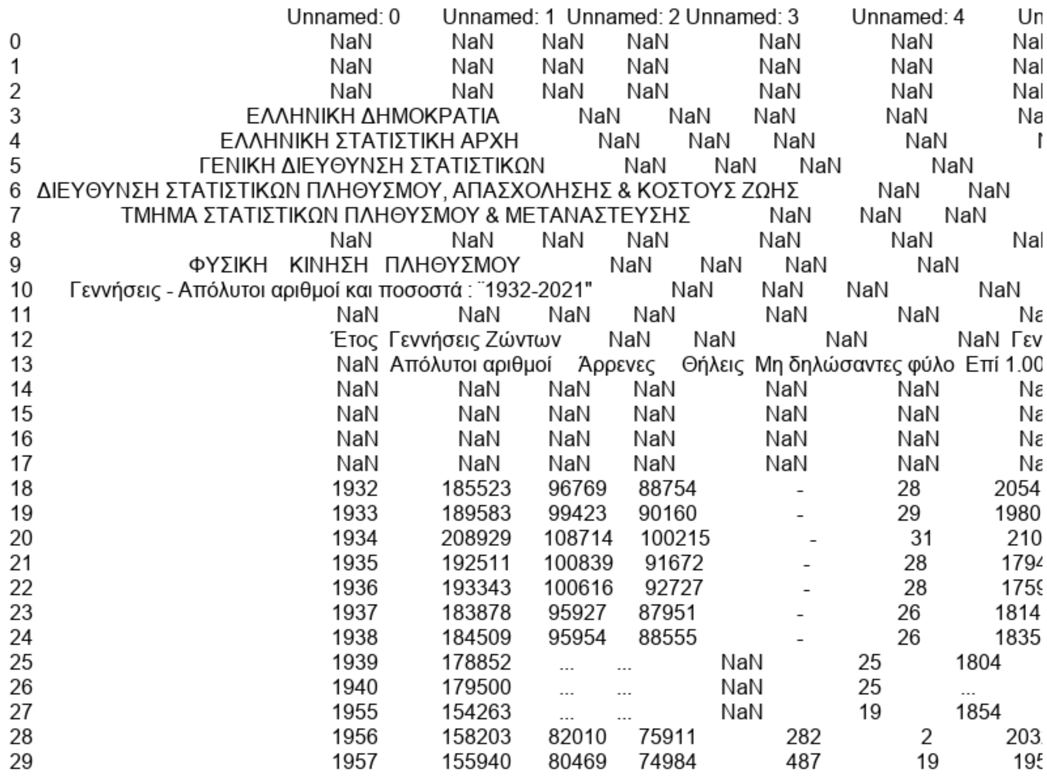
Data cleaning definition

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

Data wrangling definition

Data transformation is the process of converting data from one format or structure into another. Transformation processes can also be referred to as data wrangling, or data munging, transforming and mapping data from one "raw" data form into another format for warehousing and analyzing.

When dataset was first loaded it looked like this:



```
                                              Unnamed: 0        Unnamed: 1  Unnamed: 2 Unnamed: 3        Unnamed: 4        Un
0                                                    NaN               NaN         NaN        NaN               NaN               Na
1                                                    NaN               NaN         NaN        NaN               NaN               Na
2                                                    NaN               NaN         NaN        NaN               NaN               Na
3                                 ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ              NaN         NaN        NaN               NaN              Na
4                              ΕΛΛΗΝΙΚΗ ΣΤΑΤΙΣΤΙΚΗ ΑΡΧΗ              NaN         NaN        NaN               NaN
5                           ΓΕΝΙΚΗ ΔΙΕΥΘΥΝΣΗ ΣΤΑΤΙΣΤΙΚΩΝ              NaN         NaN        NaN               NaN
6  ΔΙΕΥΘΥΝΣΗ ΣΤΑΤΙΣΤΙΚΩΝ ΠΛΗΘΥΣΜΟΥ, ΑΠΑΣΧΟΛΗΣΗΣ & ΚΟΣΤΟΥΣ ΖΩΗΣ              NaN         NaN
7            ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΩΝ ΠΛΗΘΥΣΜΟΥ & ΜΕΤΑΝΑΣΤΕΥΣΗΣ              NaN         NaN        NaN
8                                                    NaN               NaN         NaN        NaN               NaN               Na
9                              ΦΥΣΙΚΗ    ΚΙΝΗΣΗ   ΠΛΗΘΥΣΜΟΥ              NaN         NaN        NaN               NaN
10     Γεννήσεις - Απόλυτοι αριθμοί και ποσοστά : ¨1932-2021"              NaN         NaN        NaN               NaN
11                                                    NaN               NaN         NaN        NaN               NaN              Na
12                                                   Έτος  Γεννήσεις Ζώντων         NaN        NaN               NaN              NaN  Γεν
13                                                    NaN  Απόλυτοι αριθμοί     Άρρενες     Θήλεις  Μη δηλώσαντες φύλο  Επί 1.00
14                                                    NaN               NaN         NaN        NaN               NaN               Na
15                                                    NaN               NaN         NaN        NaN               NaN               Na
16                                                    NaN               NaN         NaN        NaN               NaN               Na
17                                                    NaN               NaN         NaN        NaN               NaN               Na
18                                                   1932            185523       96769      88754                 -                28   2054
19                                                   1933            189583       99423      90160                 -                29   1980
20                                                   1934            208929      108714     100215                 -                31    210
21                                                   1935            192511      100839      91672                 -                28    179
22                                                   1936            193343      100616      92727                 -                28    175
23                                                   1937            183878       95927      87951                 -                26   1814
24                                                   1938            184509       95954      88555                 -                26   1835
25                                                   1939            178852         ...        ...               NaN                25         1804
26                                                   1940            179500         ...        ...               NaN                25          ...
27                                                   1955            154263         ...        ...               NaN                19         1854
28                                                   1956            158203       82010      75911               282                 2    203
29                                                   1957            155940       80469      74984               487                19    19
```

A considerable amount of effort was dedicated to convert the dataset into a usable format before being able to utilize it and provide any insights. The first step of the cleaning process included removing rows and columns that were blank in order to streamline the dataset and make it more manageable. Secondly, certain column names were renamed to make them more descriptive and understandable. Finally, all data were converted into numerical since it was originally stored as object types.

ANALYSIS METHODOLOGY

The data analysis process started first seeing the data in Excel after downloading from the IMF site to see who it was structured. The data is structured by Countries with all their associated indicators and then the time seriesdata. Based on the selection of the date range for this analysis from 2000-2013 I created a subset of this data with just this time range. This was read into Python using PANDAS[4] and all instances of the Indicators from this data for all the countries were extracted. This resulted in 10 data frame objects with all countries and a single indicator. After this I calculated completeness of the data as a percentage between complete and blank values. In addition to checking that all the indicators had Countries starting from Afghanistan to Zimbabwe giving 248 in total and the relevant time range. It was necessary to exclude aggregates as they would skew the data. This was done at the initial stage to prevent confusion later in the analysis. This then allowed easy calculation of the rowstatistics outlined in **Table 2** Step 1 and the column statistics shown in **Fig 6**.

The column statistics are considered separately. This is shown in the pair plot in **Fig 6**. Although **Fig 6**, is hard touse due to the large numbers of indicators considered. But it serves as a useful presentation of some key insights from the data. It also helps to contextualise the correlation and cluster maps. We see that for our attribute of interest which is the fertility rate the mean fertility rate is negatively correlated to almost all the indicators considered apart from Total Debt of a country.

Table 2: Shows Data Analysis flow and key decisions made and methods used.

Step 1: Extraction, Transform and Load	Step 2: Pre-processing	Step 3: Data Fusion	Step 4: Modelling & Visualization
1. Extract selected indicators from the relevant sources 2. Check amount of blank data if >10% pick a different indicator 3. Fill blank values with 0 4. Separate individual indicators for further processing 5. Delete first 34 rows as they consist of aggregations of the data such for OECD, Developing countries, only country wise data is considered 6. Calculate	7. Calculate column statistics for the indicators for the given time range 8. Data shows a variety in scale between indicators so test scaling methods maximum absolute scaling, min max scaling and standard scaling. 9. Produce IQR/Median plots with	11. Merge data based on Country Name and Country Code 12. Apply Maximum Absolute Scaling (MaxAbs) 13. From merged data calculate some additional features of interest such as means ratios of • Debt to GNI , • Population to GNI • Population to Labour Force %Female • CPI to Fertility Rate • GDP to GD per Capita • Health Expenditure to GDP • Health Expenditure to GDP growth • Health Expenditure to Fertility	15. The dataset is high dimensional with over 90 columns which makes visualization difficult therefore dimensionality reduction methods Tested the following: – PCA – ICA – FA – NMF – LLE – T-SNE 16. Assess Dimensionality Reduction methods by looking at explained variance, reconstruction error, and plots. 17. Use Gradient Boosting Regressor for feature selection 18. Use Gradient Boosting Regressor, Random Forest Regressor and Extra Trees Regressor with and without

country wise statistics from the time series such a: • Mean • Median, • Standard deviation, • Interquartile Range • % Change 1 Year • % Change 5 Year, • % Change 10 Year • % Change 13 Year	different scaling and choose best one 10. Produce columns statistics for time range and explore trends using [5]	Rate • Health Expenditure to Population • Fertility Rate to GDP growth • Fertility Rate to GDP per Capita • Fertility Rate to Labour Force %Female • Fertility Rate to CPI 14. Calculate 'Ahmed Score' for countries based on the above features	dimensionality reduction to predict median Fertility Rates and report Mean Square Errors (MSE) using [6]. 19. Use final data with all calculated statistics in previous steps and visualise in Tableau.

Since my aim is to predict median fertility rates I wanted to do so using summary statistics which reduce dimensionality of the data. Also it would be useful going forward as new data becomes available instead of havingto deal with a large volume of data we have a means to compress the data and get our answer.

Next step was to merge the data using the Country Name and Country Code column and again checking the length, the names of the countries, and completeness of the data. It is important to do such Quality Control (QC)at each stage in processing to catch potential problems before progressing further in the analysis. After mergingI saved the output as an Excel file to have a look the data, which showed that some columns had 'infinities' which are not caught by the PANDAS 'isnull' function so I used NUMPY[7] 'is.inf' and 'is.nan' functions to find the remaining problems in the data and fill them with zeros before progressing further. It is important to note thatthis would not have been as obvious from a dump of the data on the python console and this highlights the importance of having a variety of tools and leveraging their strengths in analysis. These extra blanks were causedby trying to calculate percentage values for example where one of entries was zero.

Then I investigated the need for scaling of data as shown in **Fig 3** and decided in this case the maximum absolutescaling worked best. The next step was to apply this scaling to the data and calculate the measure highlighted in**Table 2** Step 3. The reason I only calculated ratio of means is because there was not much variation when these measures were calculated with other statistics. Also, I designed the ‘Ahmed Score’ to combine these features into one score that could be used for ranking and visualization.

High dimensional datasets by their nature are difficult to visualize, so a useful method to visualize this datasetwas by calculating the column IQR and Median and plotting them as scatterplot. This effectively compresses thedimensions into a point using robust statistics which is then easier to visualise. **Fig 3** shows the IQR-Median plotof data and highlights the need for data scaling. This is because we see a few points dominating and the rest of the dimensions getting drowned out. To make the most of the data available and to make them comparable it was necessary to scale them shown in **Fig 3.** Out of all the 3 methods tested the Max Abs Scaler is chosen as it

exposes structures in the data not seen as clearly with the other two methods. It also highlights outliers such as Zimbabwe and Sao Tome et Principe which score poorly on most indicators considered.

Fig 5 shows a cluster map of the data which shows the detailed dependencies inferred from the data. This is tobe expected as the correlation matrix heat maps in **Fig 4 and 9** show a high level of correlation in both the scaledand unscaled data. The correlation is invariant with the choice of scaling and here serves as a good quality controlmeasure.

I wanted to be able to come with a single measure that would capture the relative importance of the means ratiomeasures calculated from the satistics (Table 2). The reason the Means Ratio of the indicators is used and not the others was because upon calculating the same measures with the other statsitics such as median there wasno additional variance gained and they were broadly similar so I stuck with one set of ratio measures. Equation 1 shows how this is calculated.

$$\text{'Ahmed Score'} = \text{arccosh} \\ (\log(\sqrt{}\text{abs}(\textstyle\sum \text{Features})))$$

Equation 1: Definition of Ahmed Score. It’s the logarithmic cube root weighting of summed features mapped asan angle to a hyperbolic plane. So bigger scores are better than lower ones. As Figure 1 shows it provides an easy way to rank countries and produces a bimodal distribution. This measure favours developing countries as shownin Figure 2.

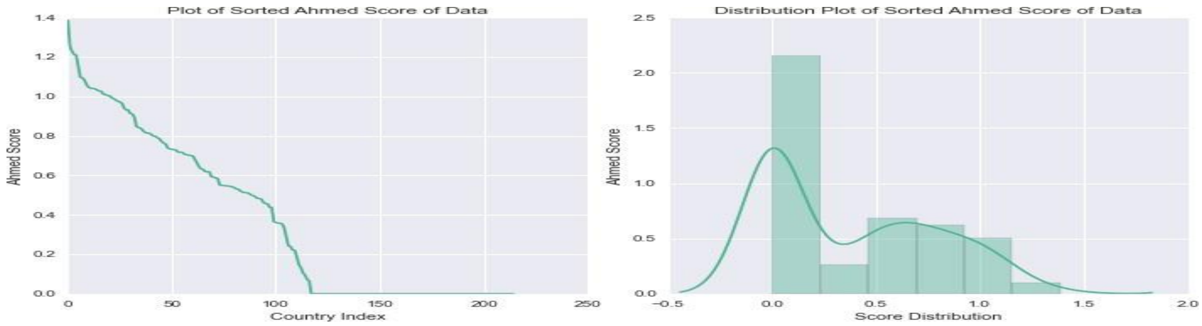


Figure 1: Left: Distribution of Ahmed Score greater than 0 from the data. Right: Sorted Ahmed Score plot of countries showing rankings produced.

The map in **Fig 2** shows how this can be visualised and the insights we can from it. We see that the median Fertility Rate is generally on the lower end compared to developing countries the usefulness of the score is captured in the size of the bubbles. To visualise the large number of ratios in this map would be difficult but one measure that captures them all makes it easier to see trends such as this score tends to favour developing countries. The reason for this could be that most of the developing countries have been undergoing recession for most of the time period considered so the indicators could have summed to zero. Also a lot of the growth over the same time period has been in emerging nations which would give them more positive scores. This is probably the underlying reason we observe the trends above.

Figure 2: Map Countries Coloured by Median Fertility Rates and Sized by Ahmed Score showing that this measure favours developing countries which makes sense because most of the measures used to calculate it revolve around fertility rate ratios which are lower in developed countries than developing countries shown smaller circles here. Generated using Tableau.

Since the data is high dimensional, dimensionality reduction was a key component of the analysis and inspirationwas taken from [8], [9] and the methods tested on this data were Principal Components Analysis (PCA), Independent Component Analysis (ICA), Factor Analysis (FA), Linear Local Embedding (LLE), t-distributed Stochastic Neighbour Embedding (t-SNE) and Non Negative Matrix Factorization (NMF).

The details of the dimensionality techniques are not discussed in detail here but references provided for those interested. But very briefly, PCA is an example of a linear dimension reduction technique that embeds data intoa smaller subspace[9]. ICA is used for revealing hidden factors that underlie the dataset[10]. Factor Analysis removes redundancy from the data with a smaller set of derived variables and these factors are fairly independent of the initial variables[11]. LLE and t-SNE are examples of nonlinear methods for dimensionality reduction[9] in contrast to PCA. NMF is another approach to reducing dimensionality that aims to find non negative matrices whose product will approximate the non-negative data[12].

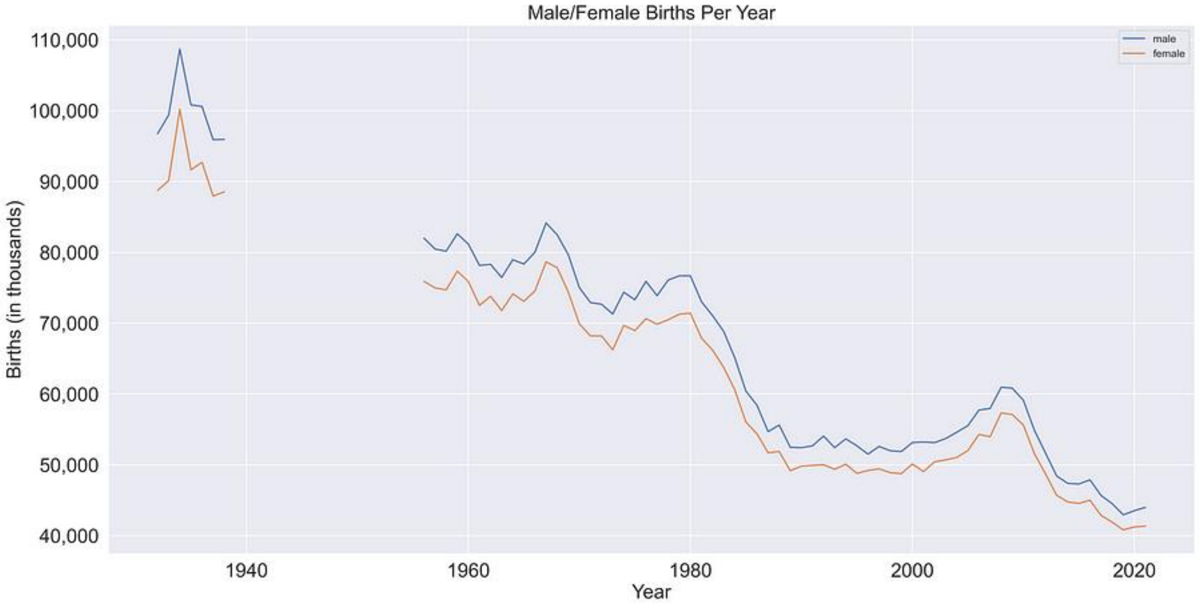
The first technique applied was PCA it was found that 3 components are sufficient to explain 99% of the variance in the data so all the other techniques are tested with 3 components and results presented in **Fig 7.** NMF on this data gave a very high reconstruction error so it was deemed unsuitable for application. The LLE produces a good projection of the data with a small reconstruction error rate but proved very difficult to integrate with regression methods so this is not considered further in the analysis but results of the testing are presented for completeness.

Regression methods are chosen because we are interested in a numerical value so classification methods are not appropriate for this type of problem. The methods chosen are ensemble methods such as Gradient Boosting Regression [13], Random Forest [14] and Extra Trees Regressor [15].

Ensemble methods are chosen because they have the benefit of being able to combine predictions of several underlying estimators which can contain combinations of strong and weak learners with the learning method, which improves the overall generalizability and robustness in contrast to using a single estimator. This is done by averaging like in Random Forests and Extra Trees or by boosting which minimizes the combined bias from the estimators like Gradient Boosting Regression. [16]

Prior to the application of the modelling techniques, the data had the target Median Fertility Rate removed and used as the target. All other statistics related to it such as IQR, Mean and Standard Deviation were dropped from the data.

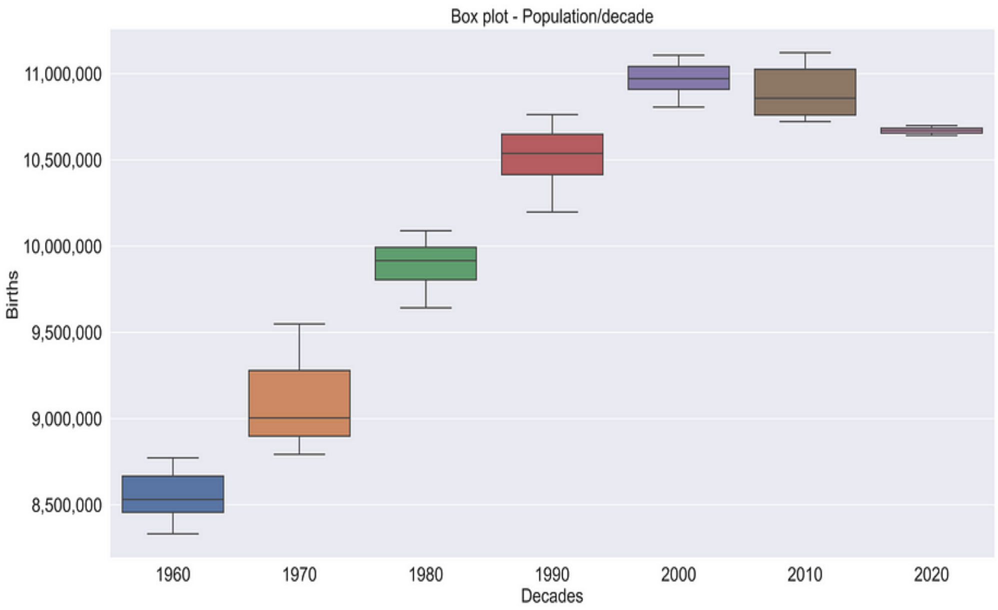
```
#To ensure proper visualization and accurate data analysis, it is
# recommended to plot and visualize the data before proceeding with
#null value processing. It is also important to check for any gaps in
#the visualization, as this can impact the accuracy and reliability of the
#analysis.
#CHECKING DATA CONSISTENCY IN TERMS OF YEAR SEQUENCE
#YEARS NOT INCLUDED IN DATASET
mylist = []
for i in item:
    if i not in x:
        mylist.append(str(i))
print (mylist)

Output:
['1930', '1931', '1939', '1940', '1941', '1942', '1943',
'1944', '1945', '1946', '1947', '1948', '1949', '1950',
'1951', '1952', '1953', '1954', '1955']
```



As you can see there is a notable lack of chronological order. I assume that this is due to the fact that Greek official authorities were unable to record diary related data during certain periods of Greek history. Specifically data collection was severely impacted during 1940–1944, where Greece was involved in World War II, (over 8% of the population died) and the subsequent Greek Civil War (1944–1949), which coincided some of the most turbulent times in Greek history.

DATA EXPLORATION AND VISUALIZATION

In the field of data analytics, there’s always a lot of back and forth involved. Before moving on to add more datasets and concatenating them to deepen our research, it would be wise to create some visualization in order to gain a clearer picture of the trends and patterns present within the data.



As I was exploring the processed dataset I thought it would be worthwhile depicting the year-on-year percentage change in birth rates in Greece. So after making the necessary code modifications I calculated the percentage change in birth rates.

```
## PERCENTAGE CHANGE IN BIRTHS
decades_change['pct_change'] =
((decades_change['births_decade'].pct_change())*100).round().map(str) + '%'
decades_change

Output:
decade births_decade pct_change
0 1960    1544786      nan%
1 1970    1435671     -7.0%
2 1980    1229539     -14.0%
```

```
3 1990    1020280     -17.0%
4 2010     951342     -7.0%
```

This analysis revealed a significant decline in births during specific decades. For example, in the 1970s, Greece experienced a 7% drop in births compared to the previous decade, while in the 1980s and 1990s, the decrease was even more significant, with drops of 14% and 17%, respectively. Moreover, the decade of 2000 saw a staggering 38% drop in births compared to the 1960s.

ENRICHING DATASETS

Following the completion of initial stages of data cleaning, preprocessing, and visualization on the first dataset, the focus shifted towards enriching it with information on population growth. This addition would require a comprehensive dataset that could cover a significant period of time.

A search for such a dataset led to the discovery of two potential sources of information, each with its advantages and limitations.

The first dataset explored, provided by ELSTAT, contained relevant information for years 2001–2021, a reasonable time frame for the study of population growth. However, it was limited in its scope, raising questions about the validity of the resulting analysis.

The second dataset, obtained from the WORLDBANK website, was far more extensive, providing a range of population growth values from 1960 to 2021. This comprehensive dataset offered a more complete and in-depth understanding of population growth patterns over a more extended period. However, the long time frame also meant that the dataset was more susceptible to errors, omissions, and inaccuracies, which could affect the results of the analysis.

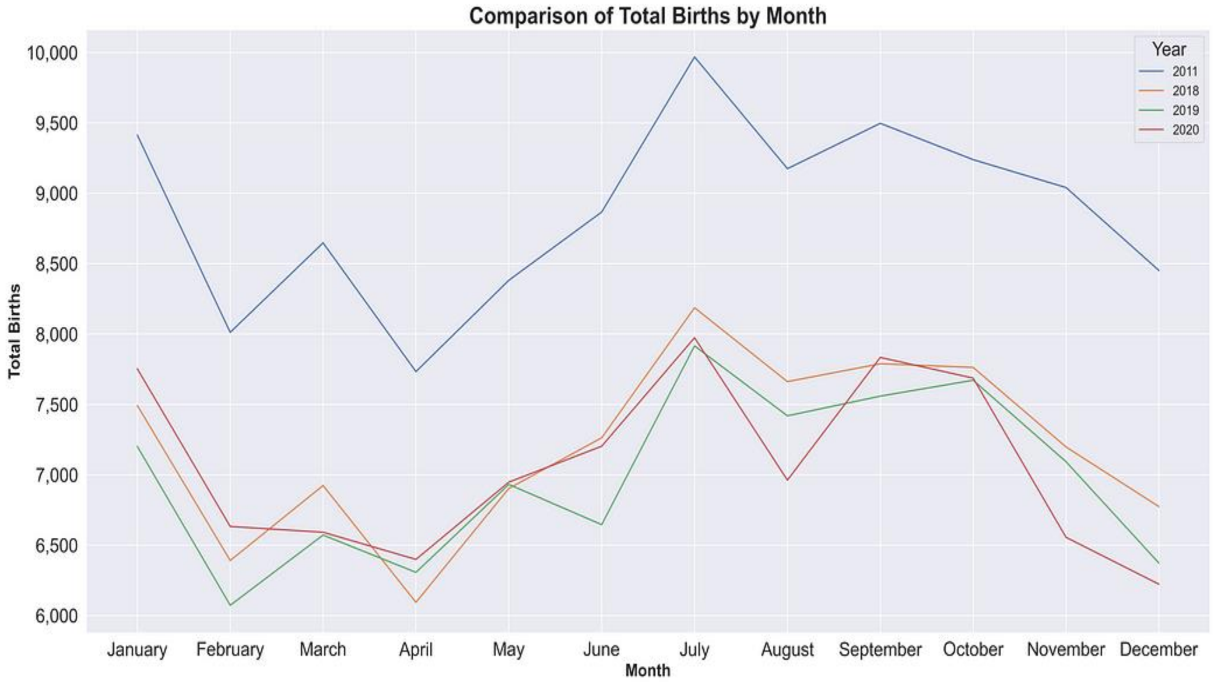
To determine which dataset would be most appropriate for the study, a critical comparison of their values was necessary.

One good way to compare the two datasets is by concatenating them and comparing their values and differences. In most cases, ELSTAT’s official dataset values are slightly lower than the values in the WORLDBANK dataset, with the calculated difference being shown in the “population_diff” column. For example, in the year 2001, the “worldbank_pop” value is 10.862.132, while the “elstat_pop” value is 10.835.989, resulting in a difference of -26.143 people. This pattern continues throughout the dataset, with the largest difference occurring in 2012, where the “worldbank_pop” value is 11.045.011, and the “elstat_pop” value is 11.086.406, resulting in a difference of 41.395 people. Quite a significant difference, right?

After this comparison it was decided to withdraw the ELSTAT dataset from further consideration and use the more extensive WORLDBANK dataset.

```
 #  WORLDBANK   VS   ELSTAT

  year    worldbank_pop   elstat_pop
0 2001     10862132        10835989
1 2002     10902022        10888274
2 2003     10928070        10915770
3 2004     10955141        10940369
4 2005     10987314        10969912
5 2006     11020362        11004716
6 2007     11048473        11036008
7 2008     11077841        11060937
8 2009     11107017        11094745
9 2010     11121341        11119289
10 2011    11104899        11123392
11 2012    11045011        11086406
12 2013    10965211        11003615
13 2014    10892413        10926807
14 2015    10820883        10858018
15 2016    10775971        10783748
16 2017    10754679        10768193
17 2018    10732882        10741165
18 2019    10721582        10724599
19 2020    10698599        10718565
20 2021    10641221        10678632
```

Moreover, a third dataset was sourced from the UNdata site, providing information about births per month.



It was observed that there were recurring patterns in birth rates that were linked to certain months of the year. For instance, the birth rate was highest during the summer months, with July and August registering the highest number of births, which could be attributed to factors such as warmer weather and increased social activity. Additionally, birth rates tended to be lower during the winter months, with December and January recording the lowest number of births.

BUILDING THE PREDICTIVE

As mentioned in the beginning of the article one of the main objectives of this project is to utilize different machine learning models, value their accuracy and predict future values for the next 30 years.

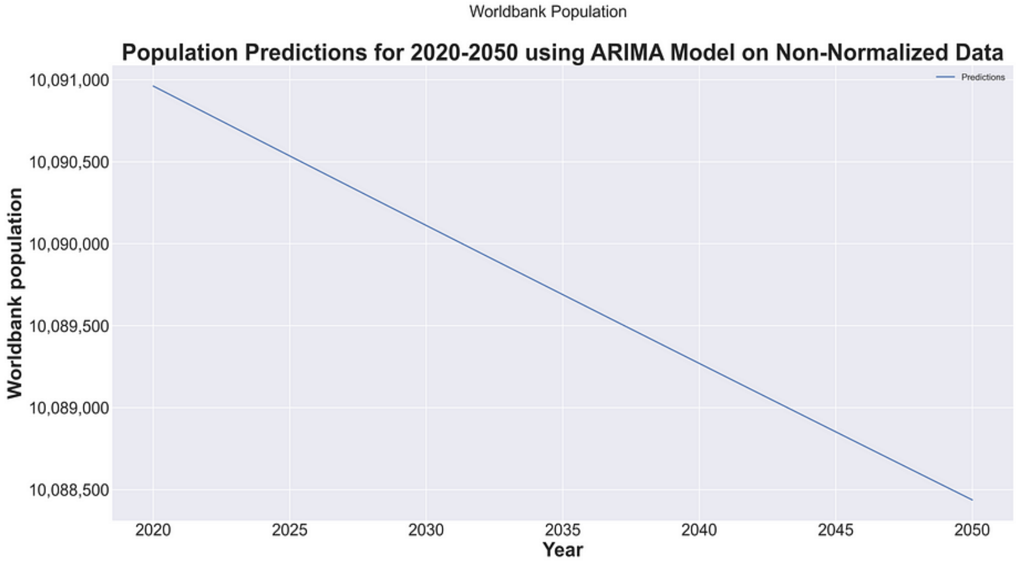
Since we do have information on the population growth of a country over time, it seems that ARIMA and SARIMAX models are the most appropriate for predicting future population growth patterns. ARIMA models are effective when the time series data has a trend or seasonality, as it can capture both. SARIMAX is a time-series forecasting model that includes parameters for trends, seasonality, and exogenous variables.

1.ARIMA model before normalizing dataset:

Firstly, we visualized predictions using ARIMA MODEL without normalizing the dataset:

```
MAE: 1561478.9033188259
RMSE: 1793899.2457239418
R-squared: -2.8469637435125428
```

As it is profound the results were not very satisfying at a



ll.

2.ARIMA model after normalizing dataset:

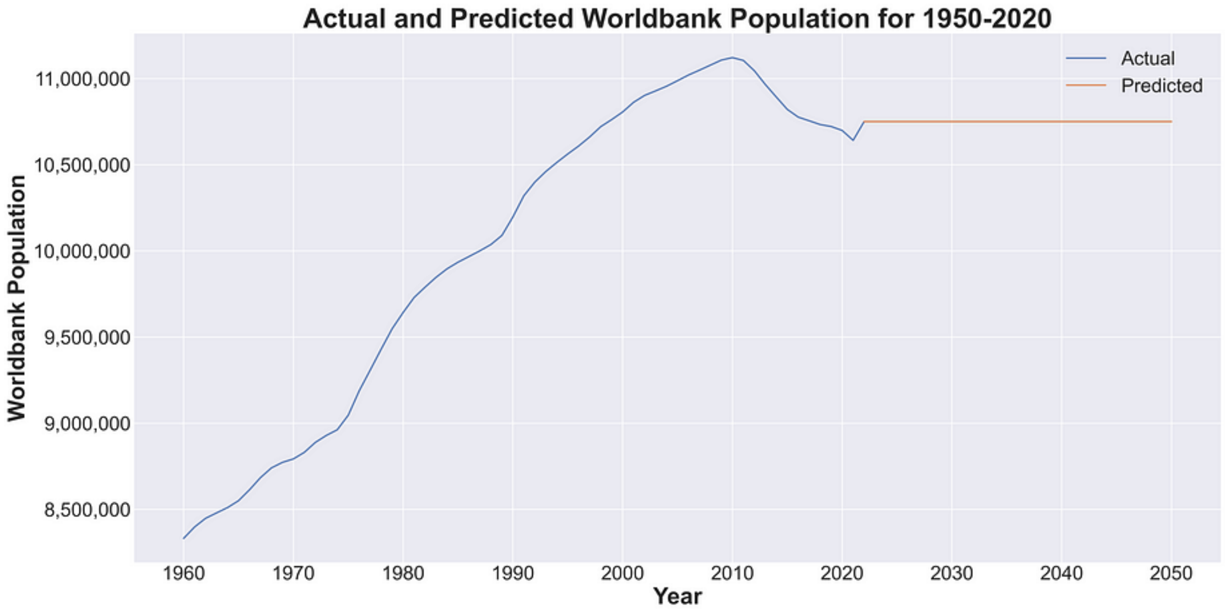
Although normalizing the data is not strictly necessary for fitting an ARIMA model in Python, it is generally a good practice to normalize or scale them for better accuracy of predictions.

Let’s check if there is improvement in metrics after data normalization:

```
RMSE: 52140.74660368626
MAE: 39432.355658901855
R-squared: -0.46113177189983
```

The results were better but not very satisfying.

ARIMA Model on Normalized Data



3.ARIMA model after normalizing and making data stationary:

Seeking to improve the predicting model’s accuracy I found out an additional reason that our dataset should be normalized. ARIMA model assumes that the input time series is stationary, which means that the mean and variance of the series are constant over time.

In previous section we normalized the dataset without checking if time series is stationary.

Now let’s check it time series is stationary:

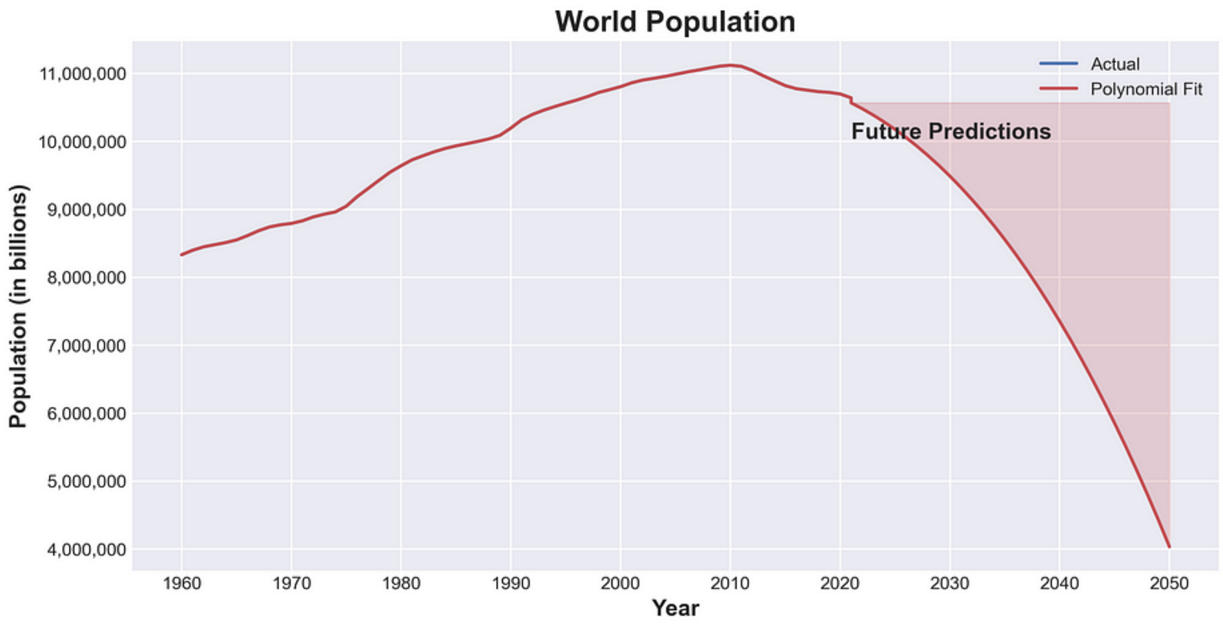
```
## TCHECKING IF DATA IS CONSTANT
# Check if mean is constant
pop_mean = pop['wordbank_pop'].rolling(window=len(pop), min_periods=1).mean()
if np.allclose(pop_mean, pop_mean.mean()):
    print("The mean of the time series is constant.")
else:
    print("The mean of the time series is not constant.")

# Check if variance is constant
pop_var = pop['wordbank_pop'].rolling(window=len(pop), min_periods=1).var()
if np.allclose(pop_var, pop_var.mean()):
    print("The variance of the time series is constant.")
else:
    print("The variance of the time series is not constant.")

Output:
The mean of the time series is not constant.
The variance of the time series is not constant.
```

Now we will try and compare two different techniques for making data stationary.

```
# Fit a 3rd degree polynomial to the data
x = np.array([i.year for i in pop.index]).astype(float)
y = np.array(pop['worldbank_pop']).astype(float)
polynomial_fit = np.polyfit(x - x[0], y, 3)
trendline_polyfit = np.polyval(polynomial_fit, x - x[0])
# Detrend the data using the polynomial fit
```

```python
detrended_polyfit = y - trendline_polyfit

# Calculate the MAE, RMSE, and R-squared for the polynomial fit
mae_polyfit = mean_absolute_error(y, trendline_polyfit)
rmse_polyfit = np.sqrt(mean_squared_error(y, trendline_polyfit))
r2_polyfit = r2_score(y, trendline_polyfit)

# Print the metrics for the polynomial fit
print("Polynomial Fit Metrics:")
print("MAE:", mae_polyfit)
print("RMSE:", rmse_polyfit)
print("R-squared:", r2_polyfit)

# Define the window size for the moving average
window_size = 5
# Calculate the moving average
y = np.array(pop['worldbank_pop']).astype(float)
trendline_moving_average = np.convolve(y, np.ones(window_size)/window_size, mode='valid')

# Detrend the data using the moving average
y_adj = y[window_size-1:-window_size+1]
trendline_moving_average_adj = trendline_moving_average[window_size//2-1:-window_size//2]

# Calculate the MAE, RMSE, and R-squared for the moving average
mae_moving_average = mean_absolute_error(y_adj, trendline_moving_average_adj)
rmse_moving_average = mean_squared_error(y_adj, trendline_moving_average_adj,
squared=False)
r2_moving_average = r2_score(y_adj, trendline_moving_average_adj)

# Compare RMSEs and print which method is more accurate
# Print the metrics for the moving average
print("\nMoving Average Metrics:")
print("MAE:", mae_moving_average)
print("RMSE:", rmse_moving_average)
print("R-squared:", r2_moving_average)

# Compare RMSEs and print which method is more accurate
if rmse_polyfit < rmse_moving_average:
    print("\nThe polynomial fit method is more accurate.")

else:
    print("\nThe moving average method is more accurate.")

import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(20, 10))

import matplotlib as mpl
ax.yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}'))

# Plot the data and trendline
plt.plot(pop.index[window_size-1:-window_size+1], y_adj, label='Polynomial Fit')
plt.plot(pop.index[window_size-1:-window_size+1], trendline_moving_average_adj,
label='Moving average')
plt.xlabel('Year')
plt.ylabel('Wordbank population')
plt.legend()
plt.show()
Output:
Polynomial Fit Metrics:
MAE: 46037.9548235934
RMSE: 59332.77389479269
R-squared: 0.9957916553091768

Moving Average Metrics:
MAE: 55969.73333333302
RMSE: 62792.28293485068
R-squared: 0.9946139927077985
```

```
The polynomial fit method is more accurate.
```



4. SARIMAX model:

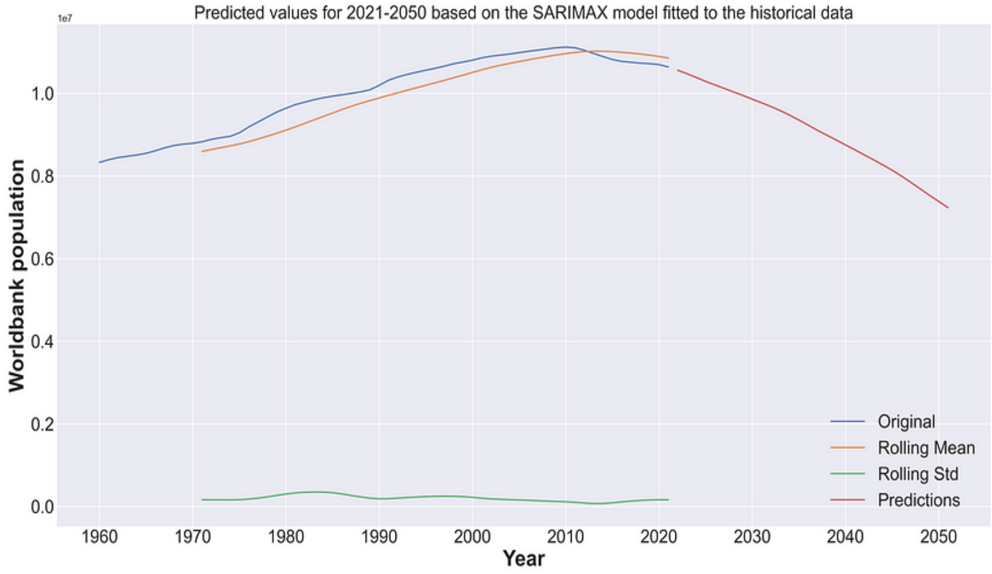
Another alternative scenario for predicting future values would be try to predict using seasonal ARIMA (SARIMAX).

```
# Create a rolling mean and standard deviation of the data
rolling_mean = pop['worldbank_pop'].rolling(window=12).mean()
rolling_std = pop['worldbank_pop'].rolling(window=12).std()

# Plot the data and rolling statistics
fig, ax = plt.subplots(figsize=(20, 10))
ax.plot(pop.index, pop['worldbank_pop'], label='Original')
ax.plot(rolling_mean.index, rolling_mean, label='Rolling Mean')
ax.plot(rolling_std.index, rolling_std, label='Rolling Std')
ax.legend()

# Make the data stationary by taking the difference between consecutive values
```

```python
stationary = pop['worldbank_pop'].diff().dropna()

# Fit the SARIMAX model to the stationary data
model = SARIMAX(stationary, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
result = model.fit()

# Make predictions for future values
start = len(stationary)
end = len(stationary) + 29 # Predict for 29 years (2021-2050)
predictions = result.predict(start=start, end=end, dynamic=False)

# Convert the stationary predictions back to the original scale
predicted_values = [pop['worldbank_pop'].iloc[start] + predictions[0]]
for i in range(1, len(predictions)):
    predicted_values.append(predicted_values[i-1] + predictions[i])
predicted_values = pd.Series(predicted_values, index=predictions.index)

# Plot the original data and predicted values
ax.plot(predicted_values.index, predicted_values, label='Predictions')
ax.legend()

# Set the plot title and subtitle to indicate the machine learning method used
title = "World Population Projection using SARIMAX Model"
subtitle = "Predicted values for 2021-2050 based on the SARIMAX model fitted to the historical data"
plt.title(title, fontsize=20)
plt.suptitle(subtitle, fontsize=16)
plt.xlabel('Year', fontsize=16)
plt.ylabel('Worldbank population', fontsize=16)
plt.show()

# Remove missing values from stationary
stationary = stationary.dropna()
```

```
# Check length of stationary
if len(stationary) < end:
    end = len(stationary) - 1
    start = end - 4

# Make predictions
predictions = result.predict(start=start, end=end, dynamic=False)

# Calculate accuracy metrics
mae = mean_absolute_error(stationary[start:], predictions)
rmse = mean_squared_error(stationary[start:], predictions, squared=False)
r2 = r2_score(stationary[start:], predictions)

print("Mean Absolute Error (MAE):", mae)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R2):", r2)
```

```
Mean Absolute Error (MAE): 13577.950093531384
Root Mean Squared Error (RMSE): 16009.9691
```



We applied 4 machine learning predictions using ARIMA model the first three times and SARIMAX model the fourth time. Each prediction was evaluated based on three performance metrics: MAE, RMSE, and R-squared. The first prediction (ARIMA-non normalized dataset) and the second one (ARIMA-normalized without making data stationary) did not fit the data well as indicated by negative R-squared values. The third prediction (ARIMA-normalized stationary data), had the best performance with the highest R-squared value and lowest MAE and RMSE values. The fourth model –SARIMAX-, had the lowest MAE and RMSE values but still did not fit the data well according to the negative R-squared value. Overall, the third model is the best choice for this dataset.

CONCATENATING DATA FRAMES

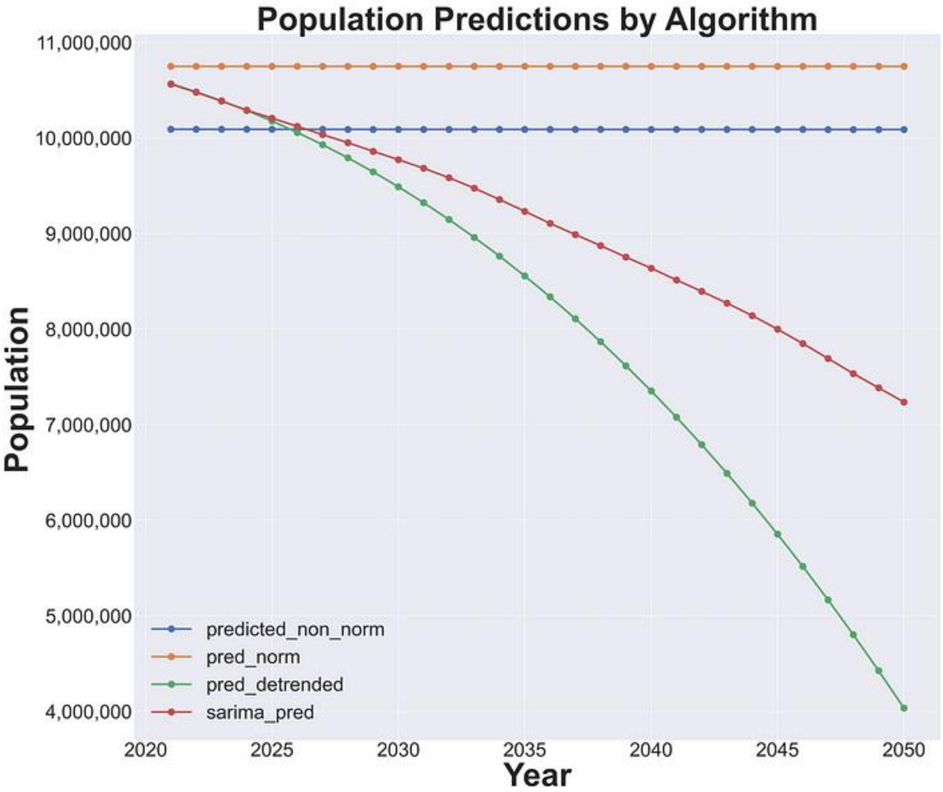
Here is a new dataframe containing all results from the four different predictive model applications, predicting population growth for years 2021–2050:

```
Output:
year   predicted_non_norm  pred_norm    pred_detrended  sarima_pred
2021     10,090,876        10,750,114   10,567,861     10,562,881
2022     10,090,791        10,750,114   10,482,580     10,477,377
2023     10,090,706        10,750,114   10,389,122     10,385,668
2024     10,090,620        10,750,114   10,287,286     10,291,249
2025     10,090,535        10,750,114   10,176,869     10,206,581
2026     10,090,450        10,750,114   10,057,668     10,122,107
2027     10,090,366        10,750,114   9,929,480     10,035,083
2028     10,090,281        10,750,114   9,792,102     9,950,584
2029     10,090,196        10,750,114   9,645,333     9,860,334
2030     10,090,111        10,750,114   9,488,969     9,773,272
2031     10,090,027        10,750,114   9,322,808     9,682,787
2032     10,089,942        10,750,114   9,146,647     9,584,198
2033     10,089,858        10,750,114   8,960,283     9,474,578
2034     10,089,774        10,750,114   8,763,514     9,356,472
2035     10,089,690        10,750,114   8,556,137     9,232,627
2036     10,089,606        10,750,114   8,337,950     9,106,152
2037     10,089,522        10,750,114   8,108,749     8,989,451
2038     10,089,438        10,750,114   7,868,333     8,872,936
2039     10,089,354        10,750,114   7,616,498     8,753,679
2040     10,089,270        10,750,114   7,353,042     8,636,808
2041     10,089,186        10,750,114   7,077,762     8,514,152
2042     10,089,103        10,750,114   6,790,455     8,394,635
2043     10,089,019        10,750,114   6,490,919     8,271,750
2044     10,088,936        10,750,114   6,178,952     8,140,933
2045     10,088,853        10,750,114   5,854,350     7,999,152
2046     10,088,769        10,750,114   5,516,911     7,848,876
```

```
2047    10,088,686     10,750,114   5,166,432     7,692,864
2048    10,088,603     10,750,114   4,802,710     7,534,222
2049    10,088,520     10,750,114   4,425,544     7,385,355
2050    10,088,437     10,750,114   4,034,730     7,236,674
```

As you can see the most accurate machine learning model of ARIMA technique calculates a predictions of total population of 4.034.730 people for year 2050. It’s disheartening to see such a substantial decrease.

Here’s the visualization of the above dataframe:



RESULTS

Fig 8 shows Feature Importance of the calculated features on the Median Fertility Rate, from here we can offer an explanation that the reason we might be observing a negative correlation between Fertility Rates and TotalDebt in **Fig 6** is probably because the most important determinants seem to be GDP measures, Health Expenditure with respect to the Population, the Percentage of Labour Force that is Female and the amount by which price of things rise every year measured by CPI.

A possible explanations could be that if a Country has a high debt burden a lot of the Government expenditure would have to go towards servicing debt rather than on productive things of benefit to the economy such as healthcare, infrastructure and social policy which probably reduces opportunities for women and thus encourages them to probably marry and raise families. This is in contrast to more developed nations where better opportunities for women mean that career considerations might be affecting attitudes towards raising families as there is a significant cost involved personally and professionally for women. This is well captured mythe custom measure designed for this dataset dubbed the ‘Ahmed Score’.

The modelling suggests that we can successfully take summary statistics of the time series data and predict the median Fertility Rate of countries accurately. The models shown in Table 3 with different processing all show accuracy of over 95%. Since three different models are used with similar accuracy we can have us confidence inour results otherwise we would have expected much greater variation and lower accuracy from the models and also say that Regression Models especially the ensemble methods are highly successful on this problem.

It is important to note that dimensionality reduction worked very well on this data. We found that PCA with 3 components was sufficient to explain 99% of the variance in data and subsequent modelling showed that our accuracy does not drop below 95% and the difference is less than 5% than if the full dataset is used. This is an important finding that we can reduce our 90 dimension data into a small dataset and still make very accurate prediction of our value of interest. Since a wide range of methods are tested and combined we can see that ourmodels are robust. With the exception of t-SNE no other processing yields final model results lower than 95%. Even then the result is only marginally lower than 95% accuracy but is still within the 5% window from the results of the full data set. The 5% can be considered as an acceptable limit on the variability of the results. However, PCA and ICA are the best methods for reducing dimensionality of this data. There also seems to be an upper limit as to how many dimensionality techniques that can be chained are actually useful. We see that with 4 methods combined we achieve around 96% accuracy which is lower than that gained by PCA and ICA. It was expected that combining PCA and FA would be quite effective since PCA is a special case FA. But we find that this combination produces the most variability in the accuracy across the 3 models. While applying ICA to the PCA and FA combination seems to stabilize the accuracy across methods. Thus we can hypothesise that if one wants the mostaccurate possible result with data dimensionality reduced on this problem PCA and ICA are good choices. If accurate and stable results across methods are required then the PCA, FA and ICA methods can be applied prior to modelling.

	ICA Comp1	PCA Comp1	FA Comp1	PCA + FA Comp1	PCA + ICA Comp1
0	HealthExp_toFertRate	RowMean_GNI	RowIQR_GNI	RowMean_GNI	%Chg5yrs_FertilityRate
1	%Chg13yrs_GDPgrowth	Debt_toGNI	%Chg13yrs_TotDebtService	%Chg13yrs_CPI	%Chg10yrs_GDPgrowth
2	%Chg13yrs_GDPperCap	RowIQR_Population	%Chg1yrs_TotDebtService	RowIQR_CPI	RowMedian_GDPperCap
3	%Chg1yrs_FertilityRate	%Chg5yrs_Populatio n	RowStd_GNI	RowStd_CPI	%Chg13yrs_CPI
4	%Chg5yrs_FertilityRate	%Chg1yrs_LabForFemale	RowMean_GDP	RowMedian_CPI	RowStd_GDPperCap
5	%Chg1yrs_HealthExp	%Chg13yrs_Population	RowMedian_GNI	RowMean_CPI	FertRate_toLabForFem

6	%Chg13yrs_CPI	%Chg1yrs_Population	RowMean_GNI	%Chg10yrs_LabForFemale	RowMean_HealthExp
7	Pop_toGNI	%Chg10yrs_Population	%Chg5yrs_GNI	%Chg5yrs_LabForFemale	RowIQR_TotDebtService
8	RowMean_GDP	%Chg13yrs_LabForFemale	RowMedian_TotDebtService	%Chg1yrs_LabForFemale	%Chg1yrs_GDPperCap
9	Debt_toGNI	%Chg1yrs_CPI	%Chg1yrs_GNI	%Chg1yrs_CPI	%Chg1yrs_GNI
10	%Chg1yrs_GDP	RowMedian_Population	%Chg10yrs_TotDebtService	%Chg13yrs_LabForFemale	%Chg5yrs_LabForFemale

Table 4: Shows top 11 features from the first components of PCA , ICA, FA, PCA + FA and PCA + ICA. **Table 4** gives us additional insight into the features that the dimensionality reduction methods have identified as being important. This is to complement **Fig 8** and the full list is given in **Table 5**. We see PCA and FA find moreglobal features in the data such as GNI while ICA finds more local features such Health Expenditure to Fertility Rate means ratio and %Changes in GDP. Since we see that PCA and ICA projected data have similar accuracy it is interesting to note the differences in the feature importance that these 2 methods return and gives us a differentway to think about target attribute. This is useful because in **Fig 4** we see that there is high level of correlation and **Fig 5** shows there are some structures in the data, the dimensionality reduction methods allows us to determine which variables are significant within the structures we are observing.

SUGGESTIONS FOR FURTHER WORK

Although this is only observed in this data, **further work** could explore whether this dimensionality reduction processing flow has the same effect across even more models and on different datasets. A graphical modelling approach to this data could also another extension to this work.

COMMENTS ON SOFTWARE AND APPROACH

This analysis as noted has been conducted in Python with the pandas[4], numpy[7], seaborn[5], matplotlib[17],and scikit-learn[6] libraries. The map presented is generated using Tableau. Since this was a fairly large analysis task, it was helpful to split into stages. The first python script dealt with pre-processing such importing, cleaning, filling blanks, calculating statistics, merging and scaling data. The second script was the analysis stage which read in the excel file generated in the previous step and calculated ratio of means, custom scores and enabled testing of dimensionality reduction techniques before they were applied. An excel file of the final merged data with allthe features are output to be visualised in Tableau at this stage. The third script was the application of the dimensionality reduction techniques deemed useful in the previous step with their associated parameters. The fourth script was the machine learning part where the data with and without dimensionality reduction along withthe various combinations were split into training and test set to be used for predictive modelling.

CONCLUSION

In conclusion, it can be said we have successfully shown that median Fertility Rates are related to macroeconomics indicators such as GDP, CPI, GNI and Total Debt of a Country etc. Also, it is noted that medianFertility Rate of a country can be successfully predicted from features derived from time series data of these indicators exclusively. A custom measure to incorporate a range of features has been found to be surprisingly, an important predictor of our value of interest.

It is hoped that being able to predict trends in demographics changes can help countries prepare for the futurewith adequate social and economic policies. This can be with regards to increasing migration to combatting greying population, greater automation within the economy and also developing legislation which promotes a culture.