**Analysing the Evolution of the Irish Population (1926-2023) A Data Analytics Perspective**

**MSC Data Analytics CA1**

*Author: Simone Finelli*

*Student ID: sba22524*

*E-mail:* [*sba22524@student.cct.ie*](mailto:sba22524@student.cct.ie)

*Github:* [*https://github.com/sifine/CA1-Population\_in\_Ireland/*](https://github.com/sifine/CA1-Population_in_Ireland/)

*MSc in Data Analytics*

*Programming for DA, Statistics for Data Analytics, Machine Learning for Data Analysis, Data Preparation & Visualisation*

*David Gonzalez, Marina Iantorno, Muhammad Iqbal, David McQuaid*

*CCT College*

*Dublin, Ireland*

**Abstract**

This report presents a comprehensive analysis of the Irish population from 1926 to 2023, focusing on its age and sex distribution. The study aims to understand **how the age composition of the population has evolved over time** and to explore the dynamics of **new births** on an annual basis.

The aging population is a global demographic challenge, and Ireland is no exception. By examining the age distribution over nearly a century, this report assesses whether the Irish **population is getting older** and the implications of this trend. A closer look at annual birth rates will shed light on the patterns and fluctuations in new births, allowing us to better understand the factors influencing demographic change.

Furthermore, this research provides methods to build a **population forecast** model to gain insights into how the population is likely to change in the future.

Through data visualization and statistical analysis, this report offers a **data-driven exploration of the Irish population's past, present, and future**, making it a valuable resource for anyone interested in demographic trends and population projections.

**Table of Contents**

[Statistics – Dataset summarisation 5](#_Toc150607921)

[Statistics – Discrete distributions 9](#_Toc150607922)

[Statistics – Normal distribution 11](#_Toc150607923)

[Data Preparation & Visualisation - Exploratory Data Analysis 14](#_Toc150607924)

[Data Preparation & Visualisation – Preparation for ML 22](#_Toc150607925)

[Machine Learning - Project Management Framework 25](#_Toc150607926)

[Machine Learning - Modelling 27](#_Toc150607927)

[Machine Learning – Evaluation and comparison 33](#_Toc150607928)

[Reference List 36](#_Toc150607929)

## Statistics – Dataset summarisation

The data we are working contains data regarding the population of Ireland divided by Year, Age and Sex. This can be easily visualized with a quick exploration of the dataset:

A screenshot of a computer

Description automatically generated

*Dataframe head*

With the original dataset we couldn’t proceed with a descriptive statistic as it contained redundant data given “All Ages” vs other ages and “Both sexes” vs “Male and Female” data. For this reason, some processing was made to gain valuable insights.

First thing we double checked that the sum of “Male” and “Female” population for each year is equal to the one of “Both Sexes”:

A graph of a number of people

Description automatically generated

*Population Trends Over the Years*

A graph of a number of people

Description automatically generated with medium confidence

*Population value by Years for Males, Females and Both Sexes*

The reason of previous analysis was to avoid having incorrect data for the “Both Sexes” aggregation as it will be heavily used through the assessment.

Please also note that the analysed dataset contains missing years, this is important to keep in mind as we are not currently synthetizing this data.

We can now gain descriptive information for male and female population:

A screenshot of a computer

Description automatically generated

*Male describe*

A screenshot of a computer

Description automatically generated

*Female describe*

To better compare this data, a boxplot was built:

A blue and orange rectangular shapes

Description automatically generated

*Population by Sex box-plot*

Looking at numbers and plot, we can say that from 1926 to 2023 the **average female population has been higher that the male** one with very similar min, 25% and 50% with respect to the male population but visibly higher 75% and max values.

We can thus say that, observing the data available to use, **Ireland has a slightly higher female population**. This may also relate to a higher longer life expectancy for women.

Instead of just looking at the population sex over years, we now want to get some information regarding the population age. This was possible thanks to a manipulation of the “Single Year of Age” column:

A screenshot of a computer screen

Description automatically generated

*Dataset filtering*

To have a good understanding of how the population age is changing over time, we want to compute (Bhandari P., 2020):

* Mean. It is the average age of the population; if is increasing over time, it may suggest that the **population is aging**, and the older generations are becoming a larger proportion of the population.
* Median. It is the middle age in a population when ages are sorted from youngest to oldest; if is increasing, it suggests that the population is aging, as the midpoint of ages is moving toward older individuals.
* Mode. It is the age that appears most frequently in the population; if it is increasing, it could indicate that there is a **growing cohort of people at a specific age**, which may be due to factors like increased birth rates.
* Standard deviation. It is a measure of the spread or dispersion of ages in the population, if it is high it indicates greater variability in ages, which may imply a **more diverse age structure in the population**.

With the data available we obtained the following data:

* *The mean is 34.95 years.*
* *The median is 30 years.*
* *The mode is 43 years.*
* *The standard deviation is 22.41 years.*

From those numbers we can say that, looking at the standard deviation, the population of Ireland appears to have a **moderately diverse age structure**, with a mix of younger and older individuals.

The presence of a mode at 43 years suggests the existence of a **significant age cohort**, which may be attributed to specific demographic events or trends.

One last question we want to answer is: *Ireland population is getting older?*

To be able to answer this question I also computed the age mean for few years in the last decade and those are the results I got:

* *Mean age in 2013 is 36.07 years.*
* *Mean age in 2018 is 37.25 years.*
* *Mean age in 2023 is 38.61 years.*

This does demonstrate that the population is getting older with a higher life expectancy.

Let’s also visualize this:

A graph of a number of people

Description automatically generated

*Weighted Population Trends Over the Years*

The graph shows the normalized population for different years by age. We can see that the age in the last decade (2013 to 2023) is becoming more and more skewed, moving from left to right as age is increasing in the last years.

## Statistics – Discrete distributions

The **Poisson distribution** is often used to model count data, such as the number of events or occurrences within a fixed interval. (Zach, 2020)

As we have data on population growth, we can model it as a Poisson distribution. Each year represents a fixed interval, and the number of births can be counted as events within that interval.

Given previous results, we want to better understand if aging is only due to a higher life expectancy or also due a decreasing number of births.

To configure the Poisson distribution, we want to answer the following problem:

*In Ireland there are, in average, several births a year. If we watch Ireland any random year from the last decade, what is the probability of having new births population less than the average?*

The problem is so configured:

* *Number of elements -> Number of births*
* *Specific place -> Ireland*
* *Time frame -> Year*

The first thing we calculate is the mean (λ) of the distribution, that is the average of new births from 1926 to 2023.

Once we got λ we derived that:

* The probability, in any random year from 1926 to 2023, to have **less than λ new births is 50%**
* The probability, in any random year from last decade, to have **less than λ new births is 95%**

This can also be seen from below graph:

A diagram of a normal distribution

Description automatically generated

*New Births Poisson Distribution*

Given that λ is 61764, the probability to have that same number of births from last decade is very low – indicating that natality is decreasing with respect to what it used to be.

If we further increase our dataset over several decades, we will notice that the mean of new births for a specific year would become closer and closer to the overall mean of new births. This means that with a larger dataset the two distributions plotted in before graph would come closer due to the **law of large numbers**.

Let’s try now to analyse the same problem with a different perspective while using a **Binomial distribution**:

*If we randomly choose 500 people what is the probability to encounter at least 10 newborns?*

We start to solve the problem for 2023:

* *Number of people -> Elements*
* *Newborns -> Attribute*
* *Within 500 people -> Limit*

For 2023 we got a 4% probability, but we also plotted a graph to show the results for other years available:

A graph of a number of years

Description automatically generated

*Probability by Year Binomial Distribution*

Inherently, last decade shows a significant drop in this probability with a YoY result not promising.

If in this case we are to increase the dataset, the distribution of encountering newborns would approximate to a normal distribution due to the **Central Limit Theorem**. This means that we would have a relatively consistent probability of encountering newborns across years, even though it may be a small percentage.

## Statistics – Normal distribution

The variable we want to analyse as a normal distribution is the value of population over time.

Before analysing the normal distribution, we must make sure that the variable is normally distributed using:

* **Q-Q Plot**
* **Shapiro-Wilk Normality Test**

A Q-Q plot compares the distribution of the data to a normal distribution by plotting the quantiles of the data against the quantiles of the normal distribution. If the data is normally distributed, the points on the plot will form a straight line. (Zach, 2021)

This is the result we got for our variable:

A graph of a graph with blue dots

Description automatically generated

*Q-Q Plot for Population over time*

The graph clearly shows that the quantiles try to form a straight line.

To double check the distribution we look at the Shapiro-Wilk Normality Test. This test measures the difference between the observed distribution and the expected normal distribution. It the resulting value is greater than 0.05, data might be normally distributed. (SPSS Tutorials, 2023)

From previous graph, we can see that our variable is normally distributed, and we can confirm this by plotting its **probability dense function**:

A graph with a line

Description automatically generated

*Probability Dense Function for Population over Time*

Given the normality of the variable we can say:

* The symmetry suggests that **fluctuations of population over time is balanced**, meaning that if there is a year were the population is above average there is also one year where it is below.
* There is no high dispersion as the standard deviation value is not big with respect to the mean. This means that **there are no years with extremely low or high population**.

This allows us to say that the **population is growing naturally** without any disruptive event creating spikes or deeps of population. Still this analysis doesn't take into consideration the fact that the population is growing because of aging previously analysed.

The variable used for the discrete distributions is the Newborn population over time and, according to Q-Q Plot and Shapiro-Wilk Normality Test, it does follow a normal distribution and could have been used for our analysis:

A graph with blue dots

Description automatically generated

*Q-Q Plot for Newborn population over time*

## Data Preparation & Visualisation - Exploratory Data Analysis

**EDA** is an approach to analysing data sets to summarize their main characteristics, often with visual methods. it is also used to understand the data and how to prepare it for the modelling of machine learning models. (IBM, 2020)

The data we are working on is about Irish population divided by Year, Age and Sex. This can be easily visualized with a quick exploration of the dataset:

A screenshot of a computer

Description automatically generated

*Dataframe head*

From the describe function we get:

A screenshot of a computer

Description automatically generated

*Dataframe description*

With this descriptive table we realise that numerical features available are only the "Year" and "VALUE", out of other features available we would be interested in the "Single Year of Age" one but now it is a string.

So far, we can say:

* Given that the min year is 1926, the max is 2023 and the mean is about 1996 we can say that for sure **we are missing some population data** for years from 1926 to 1996 (as already demonstrated before).
* The mean population is about 40k, this is the mean population **considering different ages and multiple years**.

We also want to look at single unique values for each column:

A screenshot of a computer

Description automatically generated

*Unique values of the dataset*

And we now can recall the following key information:

* We don’t have null values, but we are missing data for years between 1926 and 1996.
* “STATIC Label” and “UNIT” columns can be dropped as they only contain a single unique value.

After removing unwanted columns, we convert the "Single Year of Age" column into a numeric by creating a mapping between string and the numeric value.

If we look at the numerical order of the unique values, we are lucky enough to build a simple mapping function that is used to create the new column “Age”:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*String to int mapping function*

The first visualization we want to build is to show how the Irish population changes over time by age. If we analyse our objective, we realise that we need to build into the graph three different variables:

* *Population number*
* *Age*
* *Years*

As this could not be possible with a single 2D graph, we decide to do it using multiple graphs – each of them representing the value of the population by year for a specific age.

Given that we have 99 unique values for the age, it would be hard to build multiple. To simplify our job and make the result easier to read, we decided to use an interactive graph where a slider would allow us to easily switch graph from one age to the other:

A graph of a number of years

Description automatically generated with medium confidence

*Population by Age over time at Age 0*

A graph of a number of years

Description automatically generated with medium confidence

*Population by Age over time at Age 27*

This interactive visualization allows us to see that missing years are well distributed over time, meaning that we have some population data at least for each decade. This is very important as we will fill missing years in data preparation.

As some spikes are quite visible, we are also interested in understanding if we do have some outliers.

For this purpose, a box plot of population by age is used. Given that 100 ages are available, to avoid complexity we draw two graphs to split the ages just in two:

A screenshot of a computer

Description automatically generated

*Age 0-49 box-plot*

A screenshot of a graph

Description automatically generated

*Age 50-99 box-plot*

The graphs show us that outliers are presents for different ages, but we cannot still see if those are caused by specific years. This would be interesting to know as we need to decide if we want to keep or remove those.

Below functions allows us to flag specific rows as outliers thanks to IQR:

A computer screen shot of a computer code

Description automatically generated

*Outlier computation formula*

We can now plot a graph like the “Population by Age Over Time” but this time plotting the population by age over time. Filtering the colour by the outlier column we can hide outliers and easily spot any pattern to find those:

A graph of a number of people

Description automatically generated with medium confidence

*Population by Age over time at Year 2019*

A graph of a number of years

Description automatically generated

*Population by Age over time at Year 2021*

A graph of a number of years

Description automatically generated

*Population by Age over time at Year 2023*

Previous graphs show how outliers starts to form in year 2019 up to 2023 for ages 70 to 77. In this case outliers are due to the population getting older as previous analysed in the statistic section. For this reason, we prefer **not to remove outliers as they describe how the population is changing**.

Population by Year Over Age, Population by Age Over Year and Age boxplots were designed to obey Tufts Principles:

* (**Simplify**) The charts are simple and easy to understand. They use bar charts and boxplots to make data easily comparable. The use of a single colour avoids unnecessary complexity.
* (**Highlight the important features**) The first charts effectively highlight the changes in population over time and across different age groups. The height of the bars directly corresponds to the population, making it easy to compare different years and age groups thanks to the slider. Same principle applies to the other two.
* (**Maximize the data-ink ratio**) All ink in the plot presents data, maximizing the data-ink ratio.
* (**Appropriate scales**) The scales on both axes are linear, which is appropriate for this type of data. The y-axis starts at zero, accurately representing the values.
* (**Provide context**) The title of the chart provides clear context about what the charts represent. The axes are clearly labelled, providing necessary context for interpreting the bars.

Population by Year Over Age and Population by Age Over Year charts use the same **vibrant and bright** shade of blue that can be quite **eye-catching**. The same colour has been used as the graph is quite similar.

In contrast, boxplots use a light greenish yellow that was selected as the **complementary of the previous on the colour wheel**. This is because we want to create a contrast to show new data (outliers) without creating any **visual disruption**.

## Data Preparation & Visualisation – Preparation for ML

We are now ready to prepare our dataset for modelling and, from previous EDA, those are actions we can take:

* Filling data for missing years
* Normalizing the population data
* Drop the outliers.

We start by filling the missing data and, for this purpose, **Linear interpolation** is used.

Linear interpolation estimates missing values by drawing a straight line between the nearest available data points and filling in the missing values along that line. The reason we use this method is because it is suitable when the population changes relatively smoothly over time. (Cuemath, 2023)

To accomplish this, we first created a dataframe containing missing years in the Year column and ages from 0 to 99 in the Age one:

A screenshot of a computer code

Description automatically generated

*Dataframe creation containing missing years*

And then we concatenate it before proceeding with the interpolation:

A screenshot of a computer code

Description automatically generated

*Dataset interpolation*

Let’s recall that 58 years were missing from 1926 to 1996 while years available were only 39.

This means that most of the data now available is synthetic. During modelling phase, we will take this into account and decide if we want to consider all years or only years from 1996 to 2023.

Let’s visualize, as an example, how our population by age over time graph changes for age 0:

A graph of a number of years

Description automatically generated with medium confidence

*Population by Year over Age at Age 0*

A graph of a number of years

Description automatically generated with medium confidence

*Population by Year over Age at Age 0 - Interpolated*

The overall pattern is conserved but now we have many more outliers, especially for the range of age 0-49:

A graph of a number of squares

Description automatically generated with medium confidence

*Age 0-49 boxplot – Interpolated*

Even if more outliers are now available, it is interesting to see how those are different from the previous ones. In fact, we can say that most of the outliers are now coming from the last decade and older ages:

A graph of a number of years

Description automatically generated with medium confidence

*Population by Age Over Year at Year 2023 – Interpolated*

Again, I truly believe that those **outliers are not to be deleted** as the are they key pattern that determines how the age of population is changing and evolving over time.

Finally, we also normalize our dataset in case some models would benefit from it.

We decide to use **StandardScaler** as normalization method so that each value in the dataset will have the sample mean value subtracted, and then divided by the standard deviation of the whole dataset. We decide to use this method as it is one of the most used.

## Machine Learning - Project Management Framework

**CRISP-DM** (Cross Industry Standard Process for Data Mining) is a process model that serves as the base for a data science process. It contains descriptions of typical stages of a project, tasks related to each stage, and a description of the relationships between these tasks.

It is widely used due to its industry-agnostic and application-agnostic nature. It’s suitable for projects where the problem and the data are well-defined and understood. (EcemHazarhun, 2022)

**KDD** (Knowledge Discovery in Databases) is a data mining technique that was first used by Piatetsky Shapiro. It involves preparing and selecting data, cleaning data, obtaining prior knowledge about datasets, and interpreting solutions from observed results.

It is an iterative process in which evaluation metrics can be developed, mining improved, new data integrated and transformed to produce different and more appropriate results.

It’s suitable for projects where the goal is to discover new, potentially unexpected insights from the data. One of the significant differences between KDD and other frameworks is that KDD goes beyond IT applications and applying business context to the search. (Chehab M., 2020)

**SEMMA** (Sample, Explore, Modify, Model, and Access) is a set of sequential steps that guide the implementation of data mining applications. SEMMA is more focused on the technical aspects of a data mining project, making it suitable for projects where the main goal is to build a high-quality predictive model. (Azevedo A. and Manuel Filipe Santos, 2008)

Examples for each of those frameworks are:

* A **bank wanting to predict customer churn** would benefit from using CRISP-DM, as they have a clear problem statement and access to relevant customer data.
* A **healthcare research institute exploring a large database** of patient data to discover new patterns or relationships would benefit from using KDD.
* An **e-commerce company wanting to build a recommendation system** to suggest products to their customers would benefit from using SEMMA, as the focus is on sampling data, exploring relationships, modifying data, building models, and assessing their performance.

In this assessment we are looking at using a **supervised learning method**. This method is applicable when we have a target variable or outcome that we want to predict based on a set of features. In the case of forecasting, the target variable is usually a future value of a time series, and the features can be past values of the same time series.

## Machine Learning - Modelling

We are now ready to start training our models. Techniques we decided to use are:

* *Linear Regression*
* *ARIMA*
* *VAR*
* *XGBoost*

**Linear Regression**

Linear regression can be used for forecasting by treating the time steps as the independent variable and the quantity to be forecasted as the dependent variable. (Graphpad, 2023)

Let's assume we are only working with a single age. For that age we know the Irish population from 1926 to 2023. Each year is a **time step**, and it is our independent variable x while y, the actual population for that year, is our dependent variable.

If instead of the timestamp we use the population data of the previous year as the independent variable x, then we are talking about an autoregressive model that is exactly what we are going to use.

To better apply this model, we need to transform our dataset by **pivoting** it, this way each column represents a single time series based on the age:

A screenshot of a computer

Description automatically generated

*Dataset pivoting*

We can now define the parameters we will work on and start our multi-series forecasting:

* **y** is the time series data we want to forecast.
* **initial\_train\_size** is the number of initial observations to use for training.
* **steps** are the number of steps ahead to forecast at each iteration. We are going to set this as 5.
* **metric** is the metric to use for evaluating the forecasts. In this Through all the assessment, the mean absolute error is used.

A screenshot of a computer program

Description automatically generated

*Linear Regression lag tuning*

Previous code shows how we tried to find the optimal variation of the model by changing the lag. In the context of a forecasting problem, “lag” refers to the **step or steps back in time** we look to find patterns that help predict future values:

A graph with blue lines and text

Description automatically generated

*Average MAE vs Lag*

**ARIMA**

ARIMA stands for Autoregressive Integrated Moving Average and it means:

* **Autoregressive (AR)** -> The autoregressive part of ARIMA indicates that the evolving variable of interest is regressed on its own previous values.
* **Integrated (I)** -> The integrated part refers to the differencing of actual observations to allow for the time series to become stationary. This means that if the time series is non-stationary, it will be made so by differentiating it multiple times.
* **Moving Average (MA)** -> The moving average part indicates that the regression error is a linear combination of error whose values occurred contemporaneously and at various times in the past.

(Selva Prabhakaran, 2019)

Remember that a time-series is said to be stationary if statistical proprieties such as mean, and variance do not change over time.

Parameters that can be fine-tuned are:

* ***p*** *represents the number of time lags*
* ***d*** *is the degree of differencing (according to the stationary)*
* ***q*** *is the order of the moving-average model*

Modelling of this model for all ages is like the one for linear regression as shown by below code:

A screenshot of a computer program

Description automatically generated

*ARIMA modelling*

We didn’t proceed with the fine-tuning of this model as we wanted a better method to forecast time-series not only based on values for previous age but also for values of previous ages. This is accomplishable with VAR.

**VAR**

VAR stands for **Vector Autoregression**, and it is a model capable of capturing the linear interdependencies among the multiple time-series. (D’Amico, 2022)

To make sure that it is the right option for our problem, we need to make sure that there is a bi-directional relationship between multiple time-series. This is possible thanks to the **Granger causality test** below reported:

A screenshot of a computer screen

Description automatically generated

*Granger Causation Matrix function*

Given that the result of cells (p-value of the test) in the matrix are mostly less than 0.05, we can conclude that the **time-series are highly correlated**.

Before proceeding with the modelling, we now need to check if our time-series are stationary. To do so, **ADF test** is used:

A screenshot of a computer program

Description automatically generated

*ADF Test function*

Over 90% of time-series were **non-stationary** and they were differentiated 4 times in total to make them stationary.

When building the model, we realized that the dimension of our matrix was too big, receiving a “*non positive definite*” error. This, unfortunately, means that we have too many time-series (100 ages) we are trying to correlate to build the forecasting model.

A subset of ages was then used to build the model, considering ages multiples of 10:

A screenshot of a computer

Description automatically generated

*Age filtering*

Now, after differentiating the data, we could train the model by also finding the best lag possible based on the average MAE of each time-series:

A screenshot of a computer program

Description automatically generated

*VAR modelling and results on different lags*

**XGBoost**

Last method used is XGBoost, it is a powerful and efficient machine learning algorithm that belongs to the family of **gradient boosting methods**. It is particularly well-suited for regression and classification problems but can be adapted for time series forecasting as well. (xgboost developers, 2022)

After preparing the data a first version of the model was trained.

Given that MSE obtained was quite good, a fine-tuning technique was then used to find best values for:

* **Learning Rate** to control the step size during the optimization process.
* Max Depth to control the complexity of the individual trees in the ensemble.
* **Estimators** to determine how many weak learners (trees) are combined to form the final model.
* **Lags**, the historical values of the target variable at different time points.

**Optuna** was used for fine-tuning, it is an open source hyperparameter optimization framework used to automatically search for the best hyperparameters of a given machine learning model:

A screenshot of a computer screen

Description automatically generated

*XGBoost with Optuna fine-tuning*

After more than 12 hours the best parameters were identified to reduce MSE:

A screenshot of a computer program

Description automatically generated

*Optuna best parameters*

## Machine Learning – Evaluation and comparison

The metric used to evaluate the models is the **Mean Squared Error (MSE)** that quantifies the difference between an estimator (forecast value in our case) and the actual or true value. (Glen S., 2023)

Reasons why we decided to use this metric are below reassumed:

* Since MSE squares the errors before averaging, it gives **more weight to larger errors**. This is useful as we want to prevent larger errors.
* MSE is a differentiable function, which makes it suitable for optimization **using a variety of gradient-based algorithms** like XGBoost.
* MSE provides a single number that summarizes the average squared error of the forecasts, making it **easy to compare** the performance of different models.

The results we obtained from different models can be observed looking at below graphs:

A graph with blue rectangular bars

Description automatically generated

*MSE Results for Different Models*

A graph of different models

Description automatically generated

*Squared Root MSE Results for Different Models*

Looking at the MSE and SRME it’s clear that the best model we obtained is the XGB.

To be completely fair the only two models that should be compared are the **Linear Regression** and the **XGBoost**. The reason why we decided to move from LR to Arima and VAR is because we didn’t want to use a simple Linear Regression model to solve a forecasting problem.

On one hand, ARIMA was promising better results thanks to the concepts of **autocorrelation** and **moving average**. Basically, thanks to the concept of lags, we would have had a better method to **correlate previous data from the time-series to future**. The main problem is that this method is suitable **only for single time-series**, forcing us to lose a lot of useful information about how different ages would correlate to each other. We ended up by just applying ARIMA to single time-series, **not bringing up any additional value** with respect to a simple linear regression. For this reason, we didn’t bother fine-tuning it.

VAR had the promise to exactly overcome ARIMA limitations and **work on a multiple time-series approach** to realize a single model. Again, this was the promise, but we got stuck on the dimension of our matrix caused by the different ages. We were able to the method but **only on a subset of ages**, making it unhelpful for our final evaluation. Also, the creation of stationary data completely caused the loose of boundaries imposed by the original standardization of the data.

XGBoost, ultimately, surprised us with both the **simplicity of the implementation** and its **great results**. Its goodness is because of:

* It is an **ensemble learning method**. It builds multiple models and combines them to produce improved results.
* It includes a **regularization parameter to prevent overfitting**, which is not present in most of the traditional algorithms.
* It takes into consideration **multiple lags** for the forecasting of future values.

Let’s now see the comparison between Linear Regression and XGBoost:

A blue rectangular bars with white text

Description automatically generated

*Linear Regression and XGBoost metric comparison*

**MSE for LR is about 0.124** with **XGB having a much better value of 0.047**.

On top of benefits of XGBoost that would explain why it performed much better than other methods, if we compare specifically to Linear Regression we have to remember that XGBoost took into consideration data from other time-series while training.

This means that Linear Regression only forecasts population data **looking at previous data for that population**, XGBoost does the same forecasting also looking at **previous data for different ages** (what we were trying to accomplish with VAR).

Let’s also visualize how LR and XGB forecasts compare with actual values for newborn population in the last few years:

A graph with lines and numbers

Description automatically generated

*Population Actual vs Forecast for LR and XGB*

We can clearly see that XGB is much closer to the actual values but **trying to follow the overall trends** instead of identifying spikes. LR, instead, is very good at **identifying spikes** but those are also overweighted, causing an increase error in the overall model performance.

## Reference List

Bhandari, P. (2020). Central Tendency | Understanding the Mean, Median and Mode. [online] Scribbr. Available at: <https://www.scribbr.com/statistics/central-tendency/>.

Zach (2020). An Introduction to the Poisson Distribution. [online] Statology. Available at: <https://www.statology.org/poisson-distribution/>.

Zach (2021). How to Use Q-Q Plots to Check Normality. [online] Statology. Available at: <https://www.statology.org/q-q-plot-normality/>.

SPSS Tutorials. (2023). SPSS Shapiro-Wilk Test - Quick Tutorial with Example. [online] Available at: <https://www.spss-tutorials.com/spss-shapiro-wilk-test-for-normality/>.

IBM (2020). What Is Exploratory Data Analysis? | IBM. [online] www.ibm.com. Available at: <https://www.ibm.com/topics/exploratory-data-analysis>.

Cuemath. (2023). Linear Interpolation Formula - Derivation, Formulas, Examples. [online] Available at: <https://www.cuemath.com/linear-interpolation-formula/>.

EcemHazarhun (2022). KDD vs CRISP-DM. [online] Medium. Available at: <https://medium.com/@ecemhazarhun/kdd-vs-crisp-dm-f7b8ea99640>.

Chehab, M. (2020). Knowledge Discovery Data (KDD). [online] Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/knowledge-discovery-data-kdd-a8b41509bff9>.

Azevedo, A. and Manuel Filipe Santos (2008). KDD, semma and CRISP-DM: A parallel overview. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/220969845_KDD_semma_and_CRISP-DM_A_parallel_overview>.

www.graphpad.com. (2023). The Ultimate Guide to Linear Regression. [online] Available at: <https://www.graphpad.com/guides/the-ultimate-guide-to-linear-regression>.

Selva Prabhakaran (2019). ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+. [online] Machine Learning Plus. Available at: <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>.

D’Amico, J. (2022). Vector Autoregressive (VAR) models in Stata. [online] Medium. Available at: <https://medium.com/@JDEconomics/vector-autoregressive-var-models-in-stata-b484fb1f0f27>.

xgboost developers (2022). XGBoost Documentation — xgboost 1.5.1 documentation. [online] xgboost.readthedocs.io. Available at: <https://xgboost.readthedocs.io/en/stable/>.

Glen, S. (2023). Mean Squared Error: Definition and Example. [online] Statistics How To. Available at: <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/>.