# **Predicting Ireland’s Immigration and Emigration Count Using a Supervised Machine Learning Algorithm**

Author: Emma McCready

Student ID: sba23001

MSc in Data Analytics

Postal address: Meenagowan, Lettermacaward, Co. Donegal

E-mail: [sba23001@student.cct.ie](mailto:sba23001@student.cct.ie)

Github link: <https://github.com/EmmaMcC1802/CA1.git>

**1. Abstract**

*Population migration can be difficult to predict as it is influenced by a wide range of socioeconomic factors. This analysis aimed to predict Ireland’s net migration based on the Irish unemployment rate by identifying an appropriate machine learning model to be used in migration prediction. It was found that a random forest regressor provided the most accurate prediction. The results of this analysis suggested that the model wasn’t perfect even after hyperparameter tuning, and so unemployment rate may not be enough to predict Ireland’s migration.*

**2. Introduction**

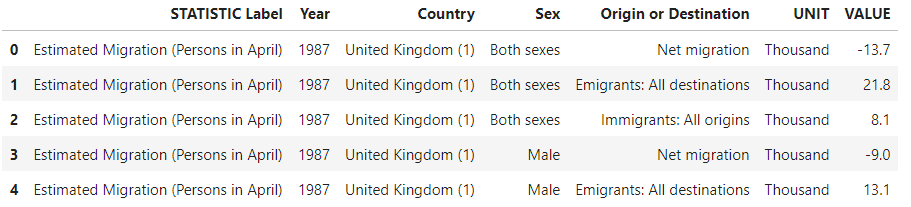
The purpose of the Central Statistics Office (CSO) is to collect, analyse and make available statistics about Ireland’s people and society, across a range of areas including construction, health, welfare, the environment and economy. Most crucially for the purpose of this analysis, the CSO maintain a record of population stocks and migration flows on an annual basis. They compile data and disseminate it for statistical purposes, where it can be readily downloaded from the CSO’s website (Ireland, Central Statistics Office 2023). This allows for multiple datasets can be compared to each other to check for correlation. With a particular interest arising from the volume of young people emigrating at the year of this report, this analysis focuses on the migration to and from Ireland. Such an analysis can be facilitated by the use of Python. With the application of machine learning methods, a machine can be trained to be able to make predictions. This highlights the importance of the work of data collectors like the CSO.

**3. Methodology**

**3.1 Dataset**

Data was obtained from the CSO and a dataset containing information on the migration count to and from Ireland for the published for 1987 to 2023 was selected for this analysis (Central Statistics Office 2023a). The migration data is collected from mid-April to mid-April of the following year, and the year listed in the dataset refers to the latter year when the data is published. For example, the 2023 rows refer to migration data collected from mid-April 2022 to mid-April 2023. This dataset contains information on the total migration split by sex for each country of origin or destination country. The dataset contains 2664 rows of data and 7 columns (Fig. 1), with the migration value provided for each type of migration (emigration, immigration, net migration) split by sex (male, female, both sexes).

**Fig. 1.** The first 5 rows of the dataset before data cleaning.

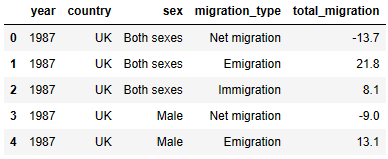


**3.2 Data preprocessing**

Exploratory Data Analysis (EDA) was carried out to develop understanding of the dataset. A standard method for EDA was followed and is outlined in Section 1 of the accompanying codebook, but EDA was performed at all steps of the analysis. Initially, EDA provided insight into how the data should be cleaned for further analysis. Columns that were deemed redundant were dropped and some column headings were renamed. Similarly, the values under the ‘Country’ and ‘Origin or Destination’ columns were shortened and/or acronymised. Fig. 2 provides a glimpse at the dataset after cleaning.

The missing values were explored to garner additional information into the type of missing data and where the values were missing by moving them into a separate dataset and exploring it.

Boolean masks were used to filter the data throughout the analysis. Specific areas of the dataset were explored to investigate areas which may not directly form part of the overall analysis as part of the EDA. In these cases, line plots were used to look at migration over time, boxplots were used to assess the distribution of migration values and identify outliers. Histograms and bar charts were also used to visualise the data. The Seaborn library was used for all plotting.



**Fig. 2.** The first 5 rows of the dataset after data cleaning.

**4. Results and Discussion**

**4.1 Exploratory Data Analysis**

The EDA process was cyclical and was adjusted throughout the analysis as more information was discovered about the dataset. This aided in identifying areas where the data could be cleaned and prepared for analysis. While doing research, another method of EDA was identified called ‘ydata-profiling’ which will be considered for future analyses to speed up this step. However, given the relatively small size of the dataset used in this analysis with the limited number of features, it was appropriate to do EDA manually.

**4.2 Data Preparation**

For ease of exploration and to aid in using the dataset, some data tidying and preparation was performed in Section 2 of the codebook. Firstly, redundant columns that provided no useful information for the analysis were dropped (i.e., 'STATISTIC Label', 'UNIT'). The UNIT column specified that the units for the VALUE column were in thousands (See Fig. 1). Thus, it was considered to multiply the VALUE column by 1,000 to reflect this. However, later, when visualising the data, it was realised this was additional data ink which goes against Tufte’s 4th principle of graphical excellence (See Section 4.3 of this report on Data Visualisation).

The column headings were renamed to make them easier to call with functions, and to better reflect the values they contained. For example, the column title ‘Origin or Destination’ implies that the data it contains is binary (i.e. either ‘Origin’ or ‘Destination’), but it actually lists the type of migration that the data row refers to (i.e. whether it was emigration, immigration or net migration). Therefore, the column was renamed to “migration\_type”.

The countries included in this dataset were quite long and contained additional text in some strings which wasn’t deemed necessary to this analysis, and so they were shortened. In some cases it was deemed appropriate to acronymise them with widely accepted terms (i.e., for the UK and USA). However, it was not considered appropriate to shorten all countries, as in some cases it may not have been clear to the reader (e.g. “Aus” for Australia may refer be misinterpreted as Austria).

The dataset spanned from the years 1987 to 2023. Exploring the data revealed that there were multiple missing values, particularly for before 2008. As the dataset is of a moderate size with only a few features, it was possible to conclude through exploring the dataset that the values were Missing Not at Random (MNAR). Prior to 2008, migration data wasn’t categorised into the list of countries it is now, namely for migration to and from Australia, Canada, and EU 15 to EU 27. Prior to this, the migration data for these countries were classified under “Other countries”. Thus, the category for ‘Other countries’ pre- and post- 2008 are not comparable, and so it was deemed appropriate to split the dataset here, and to focus all analysis on migration data from 2008 onwards.

A function was defined to filter data, which when called, used Boolean masks to create a subset of data based on the desired country, sex and migration type. This helped improve the efficiency of the code. The function was validated by creating the same subset of data manually, and after this was used in the creation of all subsets of data.

**4.3 Data Visualisation**

Throughout EDA, data was visualised to get a better understanding of the data. Seaborn, which is built on matplotlib, was the primary library used for visualising data. Tufte’s Principles were followed to assist in this goal, both to make the data more interpretable for the analyst when preparing the data, but also for the reader. These principles are outlined in Table 1.

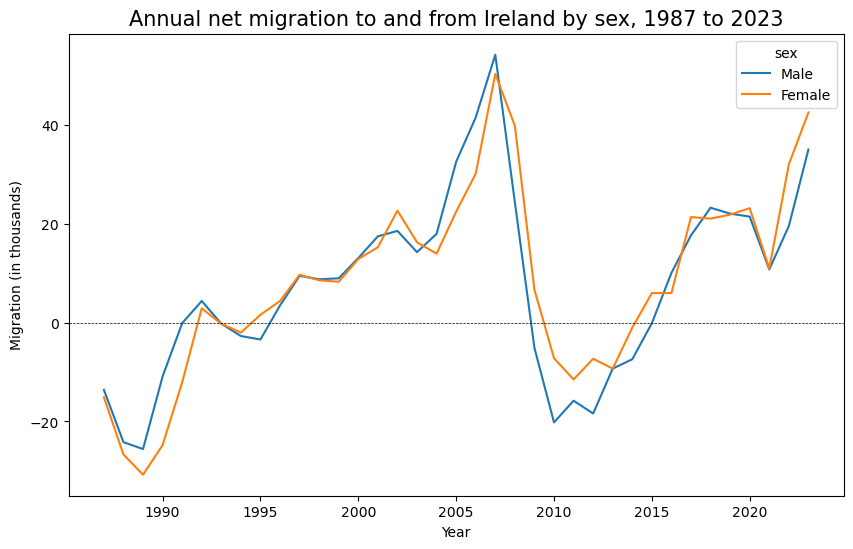
For graphical integrity, care was taken to ensure visualisations were not misrepresented. With respect to subplots (for example Fig. 4 of this report), the axis scales were kept constant to aid in easy interpretation and to not to mislead the reader. Moreover, it was ensured that there was adequate spacing if subplots were too close together. Where possible, a 3:2 aspect ratio was applied to all plots by default. While Tufte frowns on horizontal reference lines, a horizontal line was added in graphs for net migration at y=0 to highlight to viewers when there was a negative migration total. However, this line was made slimmer so it wouldn’t be a distraction or use too much data ink. The plots were all labelled with axes titles and plot titles, and legends were used where required. In choosing a colour scheme for visualisations, the documentation for the Seaborn package for assistance in choosing the colours. The default colour scheme in Seaborn was used as this is in line with what viewers are most accustomed to, and it’s optimised for representing categorical data. Moreover, the default ordering of the colours is distinct, which will aid in the viewer’s interpretability. The use of gendered colours when representing males and females was avoided. However, it was decided to the default colour schemes in this regard also, as orange is not usually associated with females. Despite blue being associated with males, it is recognised as a generic and standard colour used in plots.

With regards graphical excellence, these principles were also employed throughout, and care was taken not to add any additional and unnecessary features to plots. For example, when plotting histograms, by default Seaborn adds the kernel density estimation (KDE) to the plot. The KDE estimates the probability density function of a random variable, but this information can be generally inferred from the histogram itself, so the KDE was removed from histograms during visualisation.

**Table 1.** Tufte’s Principles for data visualisation, relating to graphical integrity and excellence.

|  |  |
| --- | --- |
| **Graphical Integrity** | **Graphical Excellence** |
| 1. The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities measured. | 1. Above all else show the data |
| 2. Clear, detailed, and thorough labelling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data. | 2. Maximise the data-ink ratio |
| 3. Show data variation, not design variation. | 3. Erase non-data ink |
| 4. In time-series displays of money, deflated and standardized units of monetary measurement are nearly always better than nominal units. | 4. Erase redundant data-ink |
| 5. The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data. | 5. Revise and edit |
| 6. Graphics must not quote data out of context. |  |

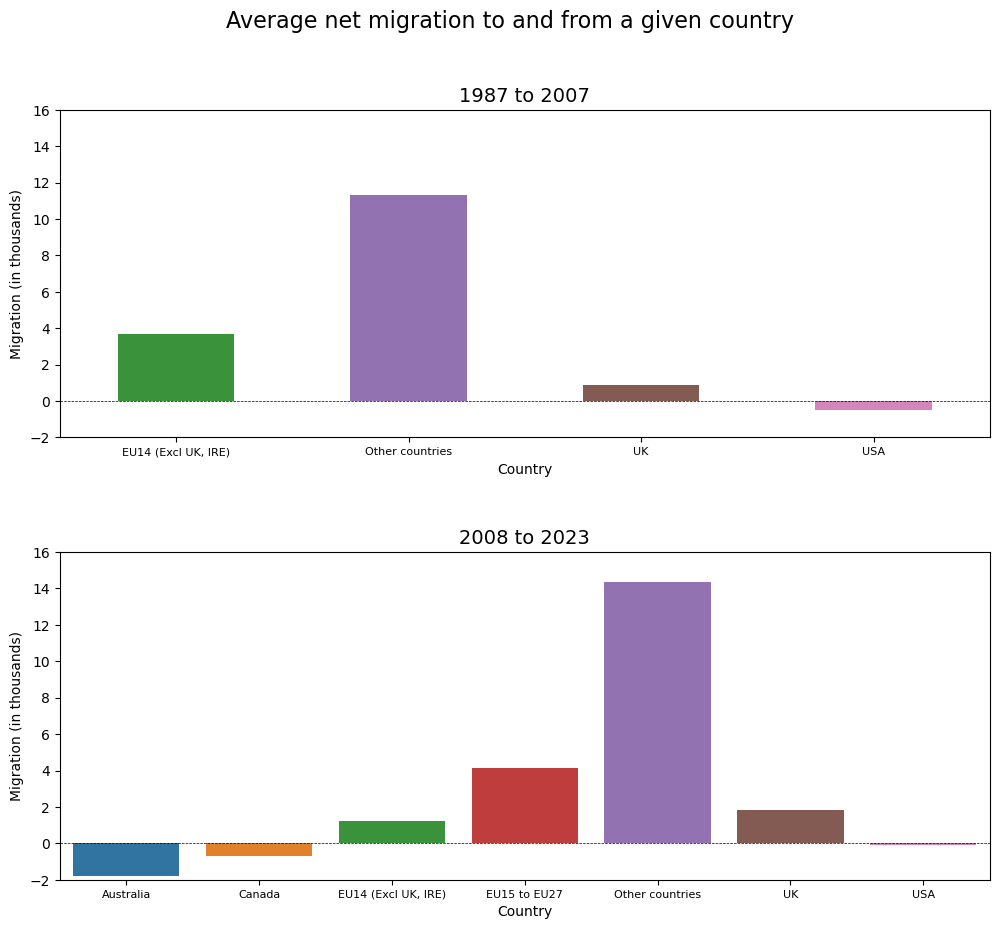
**Fig. 3.** Ireland’s net migration over time.



**4.4 Overview of Migration, 1987 to 2023**

The first goal of this analysis was to develop an understanding of migration patterns in Ireland in order to plan the best approach for integrating a machine learning model. It’s clear from Fig. 3 that this fluctuates year-on-year. A negative value on this plot indicates where there was more emigration from Ireland than immigration. The destination country/country of origin for this migration is shown in Fig. 4. The information was split into these two year ranges for the reason mentioned in Section 4.2, about how this data was collected and classified into different countries prior to 2008. This is interesting to see, particularly for countries where there’s more people emigrating from Ireland to, like the USA, Australia and Canada.

**Fig. 4.** Ireland’s net migration, grouped by the country of origin/destination country.

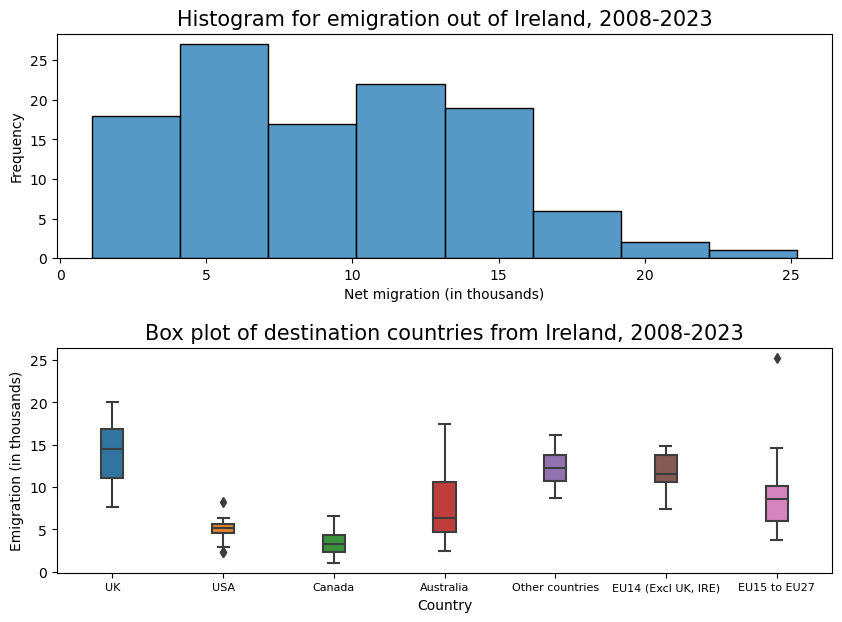


4.5 Statistical Analysis

For the analysis in this report, it was decided to focus on only a subsection of the data, from 2008 to 2023. A closer look was taken at the dataset, specifically with regard to emigration. The histogram shown in Fig. 5 (A) was plotted, and the right-skew was a surprise as the mean and median is similar for this dataset (9.206 and 9.250 respectively). It was assumed that this must be a result of outliers or extreme values, and so the boxplot in Fig. 5 (B) was plotted, confirming suspicions.

**Fig. 5.** (A) Histogram of the emigration total for each country.

(B) Boxplot of the emigration total for each country.



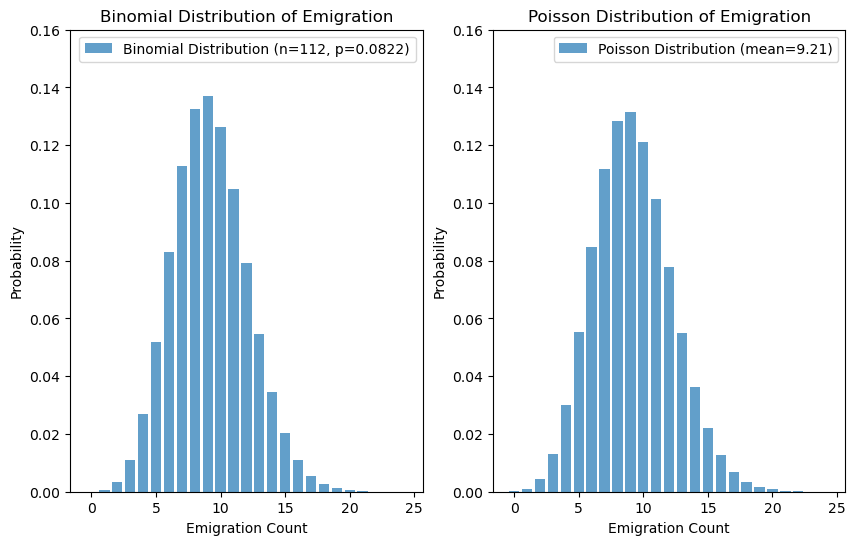
**(A)**

**(B)**

The Binomial and Poisson distribution for emigration totals have been plotted in Fig 6, which are helpful to predict all the possible probabilities of a particular emigration value in this context..

These datasets are relatively small, but an interesting feature of the Binomial distribution is that

**Fig. 6.** The Binomial and Poisson distributions for the emigration data for 2008 to 2023.



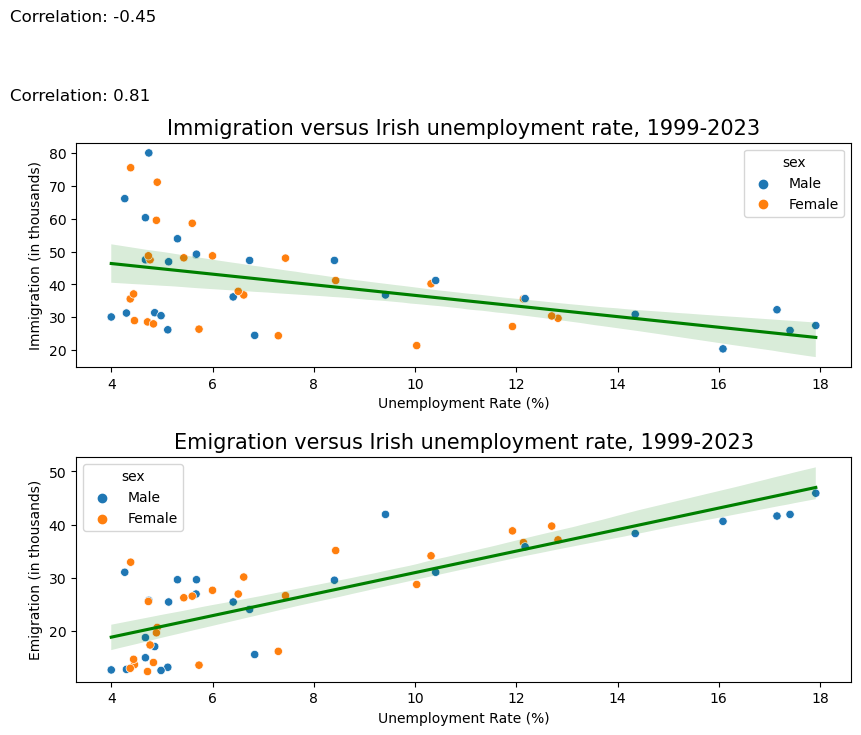
**4.6 Machine learning**

A solid project management framework is required for all data science projects to ensure efficiency in terms of time and resources. They are beneficial for providing a structured and systematic approach for data scientists to follow. One such project management framework is that of Cross-Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM is a popular framework consisting of only six phases (business understanding, data understanding, data preparation, modelling, evaluation, and deployment), which are represented as a cycle and so are iterative. Because CRISP-DM is flexible and easily adapted, it can be used for data science projects of all size, but it is renowned for designing and implementing data mining projects. The CRISP-DM framework was implemented in this analysis, as first the project’s objectives and requirements were defined, then EDA was performed, the data was prepared for modelling, models were built and then evaluated, and now with this report the framework is complete. Throughout this analysis, the project moved backwards and forwards in the steps of the CRISP-DM framework.

Supervised machine learning methods were applied in order to predict the migration, as the dataset used is labelled which the machine uses to earn. Further data from the CSO was used to supplement the dataset with the rate of unemployment (Central Statistics Office 2023b). How this data was prepared and integrated is detailed in Section 5 of the codebook. The unemployment data was filtered to obtain the average annual rate from April of a particular year to March of the following year, in accordance with the period covered by the main dataset. However, data was only available to cover 1999 to 2023, which limited the dataset size used in the machine learning methods.

Initially, the goal was to predict migration based on the Irish unemployment rate, and a basic linear regression analysis was performed to assess the relationship of these two variables (Fig Z). As expected, emigration exhibited a linear relationship with the unemployment rate. As the unemployment rate rose, emigration also increased increased, with an R2 (correlation) value of 0.81. It was expected that immigration would decrease with the unemployment rate, but the linear regression didn’t reflect this, with an R2 value of -0.45 and 0.81 for immigration.

Fig Z. Scatterplots showing the relationship between the two types of migration (emigration and immigration) and unemployment rate. The line of best fit and associated confidence bands are represented in green.



Owing to the disappointing results, it was decided to include other parameters into an actual machine learning model. As the target value is numerical, regression models were used. One-hot encoding was used to add binary values to the ‘sex’ and ‘migration\_type’ columns so that these could be included in the analysis. As the data is skewed due to outliers and extreme values, the data was pre-processed using min-max scaling. The fit and performance of five different models were evaluated (See Table 2 for the results). It was concluded that the Random Forest Regressor fit the model the strongest, and so the model underwent hyperparameter tuning using GridSearchCV() to find the optimal number of estimators and the maximum depth of trees. This was important, as adding more estimators to the ensemble can improve the model's predictive performance, but can be computationally-demanding. The maximum depth of trees refers to how long each decision tree grow, but this needs to be controlled to prevent overfitting. For this analysis, a maximum depth of 100 and 100 random estimators were found to be optimal.

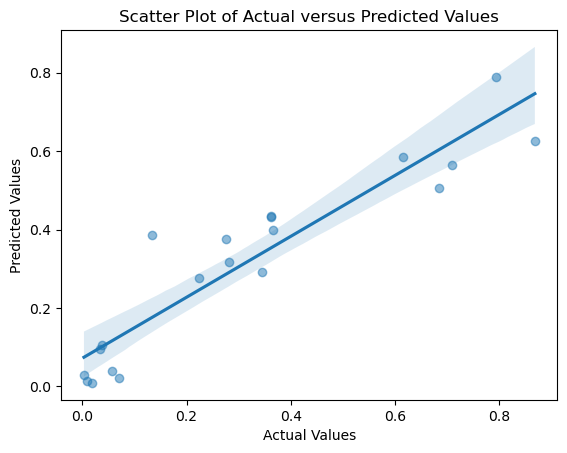
**Table 2.** Evaluation of the fit of machine learning methods applied to the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **R2 (training set)** | **R2 (test set)** | **Root Mean Squared Error** | **Mean Absolute Percentage Error** |
| Linear regression | 0.3921044 | 0.4891022 | 0.1977052 | 360.7682124 |
| Ridge Regression | 0.3891232 | 0.4743532 | 0.2005386 | 403.0211287 |
| Lasso Regression | 0.3408335 | 0.3993978 | 0.2143604 | 560.2011447 |
| Decision Tree Regressor | 1.0000000 | 0.7692657 | 0.1328639 | 85.1876140 |
| Random Forest Regressor (pre-hyperparameter tuning) | 0.9723276 | 0.8884682 | 0.0923741 | 92.4583043 |
| Random Forest Regressor (post-hyperparameter tuning) | 0.9605161 | 0.8600728 | 0.1034671 | 91.9288365 |

To evaluate fit of the models to the both the training and the test sets, the R2 was calculated, which is measured on a scale of 0 to 1, where 1 indicates a perfect match. The linear regression models (including Ridge and Lasso) tended to perform poorly, whereas the decision tree and random forest regressors fit very well. The Root Mean Squared Error (RMSE) for the test set was used to measure the average prediction error of the model, and the lower the RMSE value the better. The models all performed similarly, with the random forest regressor showing the most potential. Lastly, the Mean Absolute Percentage Error (MAPE) was used to assess the accuracy of predictions. The lower the MAPE the better, with decision tree and random forest regressors again proving to be better ML methods for this analysis. The random forest regressor was identified as the optimal model on account of the higher R2 value for the test set and the lower RMSE.

A scatter plot of the actual versus predicted values was generated to further assess the performance of the chosen ML model which can be seen in Figure A. While the model appears to be performing reasonably well, there are a number of outliers outside of the confidence intervals of the line of best fit, which indicates variability in the data. The ML analysis in this study was limited by a small dataset and the analysts inexperience. It’s unknown if the “year” feature should’ve been removed from this ML model. Performance could have been improved by removing outliers when preparing the dataset, but given how small size of the dataset, this was not considered.

Fig A. A visualisation of the performance of the random forest regressor model.



**4.7 Programming**

In terms of programming paradigms, Python is a multi-paradigm language, but the two most popular are object-oriented programming and functional programming. The study in this was based on imperative programming. The analyst took a step-by-step approach, and tracked progress of the analysis using variables. Functions were used to help define the variables, as the same function was being declared repeatedly to filter the dataset to extract specific features, and so the use of the function improved the efficiency of the code and minimised error. Print statements were used extensively to confirm that variables were correct throughout. Loops and conditionals were also used where they could be to improve the efficiency of the code, to iterate through the rows of the dataset, but the use was limited by the analyst’s inexperience. However, the imperative paradigm was beneficial given the analyst’s lack of experience, as the stepwise approach meant errors in the code were quickly located and resolved. Thus, while imperative programming was extremely beneficial to have control over the code, the implementation of more functional programme for repetitive tasks like visualisations or machine learning methods would improve the efficiency further. This is clear from some cells in the codebook where the code became very complex, and is very hard to read and interpret.

**5. Conclusions**

This dataset was visualised and presented in this report to aid readers in understanding, but the plots were discussed minimally as this is not the focus of the analysis. Additional visualisations and discussion can be found in the codebook which supplements this report.

This study found that a random forest regressor model was the most accurate machine learning method to when predicting annual population migration based on unemployment rate. However, this model was not a perfect fit. The outcome suggests that just unemployment rate may not be enough to predict Ireland’s net migration, and other factors should be considered and used to train a machine learning model, such as the GDP and cost of living. **6. References**

Central Statistics Office (2023a) *Estimated Migration (Persons in April)* [dataset], PEA18, Central Statistics Office, Ireland, available: https://data.cso.ie/table/PEA18 [accessed 25 Oct 2023].

Central Statistics Office (2023b) *Seasonally Adjusted Monthly Unemployment* [dataset], MUM01, Central Statistics Office, Ireland, available: https://data.cso.ie/table/MUM01 [accessed 7 Nov 2023].

Ireland, Central Statistics Office (2023) available: https://www.cso.ie/en/ [accessed 25 Oct 2023].