**Introduction**

As part of Project Ireland 2040 the National Development Plan sets out the Government’s over-arching investment strategy and budget for the period 2021-2030. It is an ambitious plan that balances the significant demand for public investment across all sectors and regions of Ireland with a major focus on improving the delivery of infrastructure projects to ensure speed of delivery and value for money (NDP Paper Ref). An essential component to ensure the delivery of the increased ambition set out in the National Development Plan 2021 – 2030 is an efficient, productive and sustainable construction sector. A build report was published in 2020 and it detailed a number of encouraging trends in the sector, such as increasing construction apprenticeships (BUILD 2022 Paper). Trends relating to the construction sector in Ireland will be analysed in this report along with international statistics to act as a comparison. Specifically, construction costs, labour input in the construction sector and construction sector productivity levels will be analysed. Other key economic indicators are also referenced for the purpose of enhancing the development of prediction models for the aforementioned construction sector statistics.

The structure of the paper is as follows; Data Extraction & Manipulation, Statistical Insights, Machine Learning Methods, Final Recommendations & Conclusions, and Limitations.

**Data Extraction & Manipulation**

Background

Raw data was extracted from multiple sources for the purpose of this research paper. The main source for the construction sector information was Eurostat. Key statistics related to each individual country were sourced from the relevant country’s statistical database. For the purpose of drawing international comparisons, the other countries included in this report are; United Kingdom, Germany, France, the Netherlands, the Eurozone (27 countries) and the Euro Area (20 countries). These four countries were selected for inclusion for comparative purposes as:

1. They have broadly comparable climate conditions to Ireland as mid/northern European countries (notwithstanding the south of France) and their building construction methodologies are broadly comparable also.
2. They are economically comparable on a GDP/GNP per capita basis to Ireland.
3. Their construction labour costs are broadly comparable to Ireland.

(Constr Costs Paper)

The Eurozone and Euro Area countries serve as an aggregate measure across the continent to compare against.

*Eurostat*

Eurostat acted as the primary source for the construction information. Eurostat is the statistical office of the European Union with the declared mission to provide high-quality statistics and data on Europe (Eurostat website). The datasets obtained from Eurostat are defined in the below table.

|  |  |  |
| --- | --- | --- |
| **Title** | **Definition** | **Link** |
| Production in construction - monthly data | Volume index of production (Index = 2015) | [Link here](https://ec.europa.eu/eurostat/databrowser/view/STS_COPR_M__custom_6124978/default/table?lang=en) |
| Construction producer prices or costs, new residential buildings - quarterly data | Output price index in construction - in national currency (Index = 2015) | [Link here](https://ec.europa.eu/eurostat/databrowser/view/STS_COPI_Q__custom_6125040/default/table?lang=en) |
| Labour input in construction - quarterly data | Employment (number of persons employed) (Index = 2015) | [Link here](https://ec.europa.eu/eurostat/databrowser/view/STS_COLB_Q__custom_6124890/default/table?lang=en) |

*Ireland*

The key economic indicators relating to Ireland came from a number of sources. The main source used was the online database for the Central Statistics Office (CSO). The CSO is Ireland’s national statistical office with the purpose to impartially collect, analyse and make available statistics about Ireland’s people, society and economy (CSO website). The Banking & Payments Federation Ireland (BPFI) statistics on mortgage drawdowns was also used. The datasets obtained relating to key Irish economic indicators are defined in the below table.

(**INSERT TABLE)**

An API data query on JSON-stat formatted data was also submitted on the ‘Indices of Total Production in Building and Construction (Base 2015=100)’ dataset. This was done as there was no information available for Ireland’s construction production field in the dataset obtained from Eurostat.

*United Kingdom*

*Germany*

*France*

*Netherlands*

*Eurozone (27 countries)*

*Euro Area (20 countries)*

Methodology

*Data Manipulation*

The datasets obtained and used in this paper required in depth manipulation and cleaning in order to form a robust analysis. Some of the problems encountered include; the actual data points not being in the first row of the relevant Excel sheet, very wide data (more columns than rows) with each column representing quarters for each year & uneven non-missing data for the countries included. In order to overcome these data deficiencies functions were written in Python and applied to data that were shaped in a similar format. For example a ‘clean\_eurostat\_df’ function was created which allowed for the manipulation of the datasets obtained from Eurostat in a format that is comprehensive in a Python environment for analysis. Steps included; removing the first tens rows, where information such as; time frequency, business trend indicator and unit of measure are included. This information is not relevant for the data points to be analysed.

A frequent issue with the time series data obtained for the purpose of this research paper is the misaligned time frequencies between datasets. The datasets for each country were decided to be in a quarterly format as this allows for monthly information to be easily converted, by taking each third month of the year. This manipulation was performed on multiple datasets (**GIVE EXAMPLES).**

Another frequent issue/decision to be made with the data obtained was to select a relevant starting point on which to begin analysis. The Eurostat tables had uneven starting points for each country’s statistics that were obtained. The decision was made to have the analysis originate from **XXX**, as this is when Ireland’s data entries began at this time point for most datasets included.

Exploratory Data Analysis for Final Datasets (Country Level)

Full final datasets were created for each of the aforementioned countries by joining on each of the relevant statics by the respective starting time points. This section summarises the structure of each final dataset constructed at a country level. Shape, summary statistics, relationships between variables and missing information for each of the country datasets is reported below.

*Ireland*

*United Kingdom*

*Germany*

*France*

*Netherlands*

*Eurozone (27 countries)*

*Euro Area (20 countries)*

**Statistical Insights**

**Machine Learning Methods**

The primary purpose of this section is to train supervised models to accurately predict different targets which comprise the datasets. Machine learning is a type of artificial intelligence whereby an algorithm or method extracts patterns from data (**Thoughtful**). Two machine learning methods are explored. One of the explored methods is a supervised learning machine learning algorithm utilised for the accurate prediction of the production levels in the construction sector in Ireland. According to (**AUTHOR NAME OF PRACTICAL ML)**, supervised learning is a category of algorithms classified under the supervised learning category focus on establishing a relationship between the input and output attributes, and use this relationship speculatively to generate an output for new input data points. The methods explored to build a predictive model to forecast future productivity levels in the construction sector are; the Gradient Tree Boosting or Gradient Boosted Decision Trees (GBDT) method, Ridge Regression, Lasso Regression, Elastic Net algorithm and a Neural Network. Each models performance on testing data, feature importance’s and forecasts for construction productivity in Ireland will be reported.

Supervised Learning Method

*Gradient Boosting Regression Tree Algorithm*

Gradient boosting is described by Friedman, 1999 as the construction of additive regression models by sequentially fitting a sample parametrised function (base learner) to current ‘pseudo’-residuals by least-squares at each iteration. The ‘GradientBoostingRegression’ function in used from the ensemble library in scikit-learn. This estimator builds an additive model in a forward stage-wise fashion; it allows for the optimisation of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function (**SCIKIT-LEARN**). The advantages of using a gradient boosting regression tree (GBRT) algorithm, and a reason for exploration in this paper is that the approximation accuracy and execution speed is substantially improved with the incorporation of randomisation. At each iteration, a random subsample of the training data is drawn, used to fit the base learner and compute the model update for the current iteration (Friedman, 1999). The idea is that the algorithm combines weak and robust learners, by adding each weak learner to the model to correct the prediction errors created by previous models (**Ecofriendly Concrete Compressive Strength**).

The ‘GradientBoostingRegressor’ function is used from the ensemble library of scikit-learn in Python. Hyperparameter tuning is performed in order to fit the model with the highest prediction accuracy. The hyperparameters to be optimised are the loss function, learning rate, number of boosting stages and the maximum depth of the individual regression estimators. The learning rate and the number of boosting stages are the most crucial hyperparameters for the GBRT model. Therefore, it was decided to include these and two other hyperparameters in the grid to be optimised in order to fit a robust model. GridSearchCV is used resulting in the following optimal hyperparameters for the GBRT model. The learning rate was set to be 0.01, the loss function was set to be based on absolute error, the maximum depth of the individual regression estimators was to be set at 9 and the number of boosting stages was to be set at 500. These parameters were added to the GBRT model fit on the training and testing data producing an R-squared of 94% and a root mean-squared error of 17.3714 on the testing data.

*Ridge, Lasso & Elastic Net*

Ridge Regression

Ridge Regression was introduced by Hoerl & Kennard (1970) is one of the penalisation or regularisation methods that reduces overfitting by considering collinearity among regressors. With ridge regression the model coefficients are shrunk towards zero but are never zero, resulting in more prediction accuracy at the cost of a small increase in bias. The principal reason that ridge regression was included as part of this research paper was to address the problem of multi-collinearity among the regressors, brought about by high correlations. Ridge regression also informs model developers which features are more significant in determining the defined target variable than others.

The ’Ridge’ function is utilised from the linear\_model library of scikit-learn in Python to fit a linear least squares model with L2 regularisation. Hyperparameter tuning is performed on one parameter, namely, the alpha parameter which is the constant factor that multiplies the L2 term, which controls the regularisation strength (**SCIKIT-LEARN**). Using the GridSearchCV function to search through possible alpha parameter values ranging from 0.1 to 25 resulted in using an alpha value of 0.1, optimised on the r-squared value. Setting the alpha parameter for the ridge model on the training data produced an R-squared of 97.4% and a root mean-square error of 11.3336 on the testing data.

Lasso Regression

Least Absolute Shrinkage and Selection Operator (Lasso) regression, introduced by Tibshirani (1996), is similar to Ridge regression in its efficacy of dealing with many prediction variables. It is a regression method analysis method that performs both variable selection and regularisation in order to enhance the prediction accuracy and interpretability of the statistical regression model (Emmeet-Streib, Dehmer, 2019). Lasso differs from Ridge in that it deals with variable selection and shrinkage of the parameter. Lasso shrinks some regression coefficients toward zero and others to exactly zero. Therefore, a Lasso regression model automatically creates a subset of variables on which a defined target is predicted on, while also shrinking other variables toward zero.

The ‘Lasso’ function is utilised from the linear\_model library of scikit-learn in Python to fit a linear model trained with L1 prior as a regulariser. A similar hyperparameter tuning exercise is performed on the lasso model as the ridge model, with the alpha parameter tuned. Alpha, in the context of Lasso regression, is the constant term that multiplies the L1 term, controlling the degree of sparsity of the estimated coefficients. The same alpha coefficient is returned as the ridge model which is set to 0.1. A lasso model fit on the training data with alpha set to 0.1 results in an R-squared coefficient of 97.4% and a root mean-square error of 11.3693. This is not to dissimilar to ridge regression.

Elastic Net

Zou and Hastie (2005) introduced elastic net to extend the Lasso by improving some of its limitations, with a particular emphasis on variable selection. Elastic net produces a regression model that is penalise with both the Ridge regression penalty term (L1) and Lasso regression penalty term (L2). The Ridge regression component generates a sparse model by shrinking some regression coefficients exactly to zero and the Lasso regression component removes the limitation on the number of selected variables, encouraging grouping effect and stabilising the Ridge regularisation path (Park, Konishi, 2015). An Elastic net model was included for completeness in order to assess which model (Ridge or Lasso) has a better prediction accuracy for productivity levels in the construction sector in Ireland.

The ‘ElasticNet’ function is utilised from the linear\_model library of scikit-learn in Python to fit a linear regression model combined with L1 and L2 priors as regularisers. There are two hyperparameters to be tuned with the Elastic Net, namely, the alpha and l1\_ratio parameters. The alpha term is the constant value that multiplies the penalty terms. The l1\_ratio term is defined as the Elastic Net mixing parameter, in an interval range of 0-1. If the l1\_ratio is set to 0 the penalty is an L2 penalty (Lasso). If the l1\_ratio is set to 1 the penalty term is an L1 penalty (Ridge). GridSearchCV is used to fine tune the hyperparameters based on r-squared. The resulting values with the best score set alpha to be equal to 0.1 and the l1\_ratio to be 0.98. This means that the Elastic Net model is very closely aligned to the Ridge regression model. An Elastic Net model is fit on the training data resulting in an R-squared coefficient of 97.4% and a root mean-square error of 14.8018 on the testing sample of the data.

*Neural Network*

According to IBM, Neural Networks are a subset of machine learning and are at the heart of deep learning. The name and structure is inspired by the human brain, mimicking the way that biological neurons signal to one another. Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above a specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network (**IBM**).

ANNs have been widely used for time series forecasting, showing good performance in predicting stock market data. (**LIN ET AL**), applied an ANN to predict option prices relating to the Taiwan stock index, (**Mohamed et al)** used neural networks to forecast the stock exchange movements for the Kuwait Stock exchange. While the purpose of the ANN developed as part of this research paper is not related to stock price movement, the same principles can be applied for predicting productivity levels in the Irish construction sector.

An important feature of neural networks is the ability to learn from their environment, and, through learning to improve. The major advantage of neural networks is that they are data driven and do not require restrictive assumptions about the form of the basic model (**Ciobanu, Vasilescu, 2013**).

A two-layer neural network is developed using Keras’ sequential model. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. The Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. Each layer is ‘dense’, meaning all the neurons in the previous layer are connected to all the neurons in the next layer. Activation functions are included in the input layer and the first layer to transform an input signal into an output signal which in turn is fed as input to the next layer in the stack (**Sharma**). An optimisation algorithm is also defined in the network to inform Keras how the network will learn. The number of neurons in the input and first layer, the activation function and the optimisation algorithm to train the network on are all the hyperparameters tuned using GridSearchCV. The number of neurons to be tested range from 50-250 increasing by 50 units each time. The activation functions to be tested are ‘relu’, ‘sigmoid’ and ‘tanh’. The optimisation algorithms tested are ‘adam’, ‘rmsprop’ and ‘sgd’. Following running GridSearchCV on the two-layer neural network created through Keras sequential model the number of neurons was set to 50, the activation function was set to ‘relu’ and the optimisation algorithm was set to ‘adam’ based on mean-squared error. Setting these as the parameters for the neural network resulted in a training RMSE of 17.706 and a testing RMSE of 20.03. The training and testing loss histories and the training and testing loss mean-square errors can be found in the **appendix**.