Submitted in part fulfilment for the degree of BSc

Improving Equation Discovery Techniques for Time Series Forecasting

Luke Richardson

16 June 2021

Supervisor: Dimitar Kazakov

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Dimitar Kazakov, for guidance and support during this project, at all hours of the day. I would especially like to thank David Griffin, who really stepped in during an unforeseen circumstance and picked up the workload. Without him, it is likely that this project wouldn't be close to the standard it is today.

Most importantly, I would like to thank my parents. All the work that exists in this report as well as the rest of my degree is dedicated to them, and it's all completed in the hope of making them proud. Hopefully, that has been achieved.

STATEMENT OF ETHICS

This project was conducted in accordance with ethics guidelines set out by the University of York. The work is meant solely as academic research and any use by individuals or groups either commercially or personally, outside academia, should use it at their own risk. There are no legal issues in this project as all data has been downloaded from publicly available sources and no other personal data has been gathered, used of stored during this project.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Executive Summary

Time series data is a series of points which are indexed in time order. Because of the temporal nature of time series data, attempts are often made to extract patters and trends as models in order to predict what values will be in the future. An area of interest when modelling time series data is macroeconomics, as an accurate forecast of these variables can have implications not only in macroeconomics but also in the wider financial sector.

The aim of the project was to improve the existing techniques of equation discovery for time series forecasting. The previous tool, LAGRAMGE [1], with this functionality was developed in the late 1990s, meaning some machine learning techniques have since been developed and the field of time series forecasting has moved further towards automated computational approaches rather than human inferences. This resulted in the techniques and technology improving. Attempts at improving or redesigning this have struggled with poor optimisation or is unavailable as an open-source project and hidden behind a paid license. A project report by a student at the University of York [2] laid the foundations for a new implementation of an equation discovery tool written with extensibility in mind making the subject more accessible. This was however merely an introduction in which the tool lacked features. The brute force approach to the tool prevented running on a number of equations past a certain depth and made it non-feasible to model more than a handful of variables at once. This created motivation to make improvements on the previous approach and evaluate the forecasting performance and efficiency of the new methods.

Two approaches were taken. The first exploring new methods of traversing through the search space of equations generated with an informed search, previously done via brute force. The second, adding constraints to how the equations are generated. The benefit of exploring these two approaches is that it shows feasibility and opens the door for further analysis and extension of these methods in future work.

To measure forecasting performance of the implementations, baseline forecasts were made on three macroeconomic time series. These time series were the CPI, GDP and unemployment rates in the United States. The method used was ARIMA, which is well-known for time series forecasting. The forecasts were also run on the previous equation discovery tool developed by a previous student [2]. This allowed the results to be compared with a baseline method and the

previous equation discovery method which forecasted a set of US macroeconomic variables. Due to the more recent datasets available, it allowed the assessment of predictive accuracy of the methods over a period of sharp, impulsive moves in the data caused by the Covid-19 global pandemic.

The results that came out of the experiment showed a level of predictive accuracy that most of the time was equivalent to or an improvement of the baseline techniques. Observing, visually, the forecasts from the two approaches taken clearly showed the ability for the models to pick up on short-term moves, especially when the models were trained on a reduced dataset excluding periods of post-World War economic recession and growth. When obtaining the results, the informed search method proved to be an effective method of reducing the number of equations that needed to be evaluated in the search space, in a domain in which the space can be vast.

It was concluded that the main objectives of the project were achieved and issues with the implementations of the methods were identified. An alternative to the LAGRAMGE tool was developed and the resulting program is easier to maintain and extend. A method has been developed which can more efficiently discover more complex equations, allowing more complex problems to be modelled, with the appropriate machine learning techniques that were missing in the previous LAGRAMGE tool. The versatility of the tool has also been explored by modifying the nature of the restraints on the search space, using a probability distribution. This shows potential use in other domains.

This project has no legal, professional, ethical, social and commercial issues.

1 Introduction

The field of macroeconomics dates to 1936, originally developed by John Keynes [3]. People have since been attempting to accurately predict the future of macroeconomic values and governments rely on this information in order to make policy decisions. The recent spike of interest in machine learning opens the door for new techniques and technology to execute this task.

Although there has been recent interest in the field of equation discovery, the current methods are still outdated. Also, the quality of application in certain domains is still unevaluated. The previous methods in the field of macroeconomic forecasting have been implemented using the LAGRAMGE tool, built in 1997 [1]. Any attempt to improve on the LAGRAMGE has left room for further development as the search space of possible models and equations forcing makes it unfeasible to find equations to the depth required to get the complexity required in some domains. Restricting the search space of equations using a declarative bias has provided huge improvements in effectiveness [4]. Despite this, the search space is often still vast and further restraints or methods of traversing the search spaces in more efficient ways should be explored. The techniques used in this project, although focusing on macroeconomic forecasting, can be applied elsewhere [5]. As this project is aimed at computer science students, macroeconomic terms have been defined in the background.

The aim is to create an alternative to LAGRAMGE on the back of modern Python [6] libraries, extending a previous attempt [2] with an informed search. The result should be a program that is easier to analyse, maintain, port and extend. The tool should be able to forecast macroeconomic parameters accurately enough to compete with modern day methods. We shall evaluate the results in terms of predictive accuracy and time complexity. Another aim is versatility, specifically in terms of defining the declarative bias, allowing the tool to be extended into other domains.

2 Background and Literature Review

This chapter provides an outline of the relevant literature. There will be a discussion of literature relating to key areas in the project and then comparison will be made between sources, assessing the strengths and weaknesses of each of these sources.

2.1 Macroeconomic Parameters

Macroeconomics is the field of economics concerning performance structure and behaviour of an economy. Macroeconomists study parameters such as GDP, unemployment rates [7]. Historic data of these macroeconomic parameters are often provided as time series data on a quarterly or monthly basis. There are many statistical techniques [8], techniques involving neural networks [9] and equation discovery techniques [2] to model these parameters. Once modelled, they can be forecasted – predicting future values. These forecasts are used extensively in industry and government [10]. If accurate, they can be useful when considering policy decisions [11] and predicting future moves in the stock market [12].

Gross Domestic Product (GDP)

Gross domestic product or GDP is a measure of the size and health of a country's economy over a period of time (usually quarterly or annually) [13]. It is also used to compare the size of different economies at a different point in time. When taking the expenditure approach, GDP can be represented by the equation:

$$GDP = C + I + G + NX \quad (2.1)$$

- C, consumption – Private-consumption expenditures by households and non-profit organizations
- I, investment - Business expenditures by businesses and home purchases by households
- G, government spending - Expenditures on goods and services by the government
- NX, net exports - A nation's exports minus its imports (can be written $(X - M)$) [14]

Unemployment rate

"Unemployment rates are calculated, in accordance with international guidelines, as the number of unemployed people divided by the economically active population (those in employment plus those who are unemployed)". [15] This is measured on a monthly basis in the Euro area and the UK.

CPI

Consumer price index (CPI) measures changes in the price level of a weighted average market basket of consumer goods and services purchased by households [16]. It is estimated with the formula:

$$CPI_t = \frac{C_t}{C_0} * 100 \quad (2.2)$$

CPI_t is consumer price index in current period. C_t is cost of market basket in current period. C_0 is cost of market basket in base period.

2.2 Macroeconomic Modelling

Stephen K McNees stated that “the task of modelling an economy is obviously quite a formidable one” [8]. Despite this being true, it has not stopped people trying. Methods exist such as human crafted conditional models built by forecasting services such as Chase and Wharton Econometric Forecasting Associates in the late 70s, and statistical models such as vector autoregression (VAR) and autoregressive integrated moving average (ARIMA) [8].

As macroeconomic variables can be affected by not only endogenous variables, but also external variables such as policy decisions and political forces [8]. It has been a point of discussion over the years if these factors should be considered to improve accuracy of a model used to forecast macroeconomic variables. McNees takes an excursion into the philosophy of social science when discussing the human adjustments made to the models.

Although stating that adjusting these mechanically generated macroeconomic models in order to incorporate external variables is not inherently “unscientific”, he does state that some of the adjustments can be made for “capricious or unscientific reasons” [8]. An example of this can be seen in the ‘velocity miracle’. In which a model was overwritten as a result of forecasters coming under criticism for ‘undue pessimism’. They kept their model optimistic which resulted in some of the largest nominal GNP forecast errors recorded. Not only is there an issue in unscientific adjustments being made, but the justification may also be flawed. McNees suggests that documentation of the model modification is required to make this method reliable however, this would be difficult for large models and additional information may annoy and/or confuse decision makers.

This is not an issue with statistical models as this assumes all variables are endogenous, removing skill of the model user from the predictive performance of the models [8]. The reproducibility makes these statistical methods a good baseline against new machine learning techniques.

When looking at the predictive accuracy of VAR and ARIMA models on macroeconomic forecasting, the results in [17] show that, for short-horizon (one and two-step ahead) forecasts, ARIMA outperformed the VAR forecasts. The VAR forecasts did however beat the ARIMA forecasts on all 'four-quarter-ahead' forecasts. When finding short-horizon forecasts, then ARIMA is clearly a more accurate baseline. On the contrary, if out of sample predictions range further than merely a couple of quarters into the future, another method such as a flavour of VAR may need to be considered. Interestingly, linking back with the discussion had in McNees [8], when looking at the root mean squared errors (RMSE), the VAR and ARIMA forecasts kept up with the constant adjustment models for up to six months in the future. This suggests that the unconditional models are a good baseline for short-horizon forecasts. Lupoletti and Webb assess the effectiveness of combining structural (produced by commercial forecasting services) and VAR forecasts using the ratio method. In the results section of the paper, it is proven a useful exercise, as on twelve of the sixteen examples, the combined forecasts are more accurate (including ties) [17]. This is something to consider with results from the equation discovery methods as it may be a useful exercise to attempt to combine with structural forecasts.

2.3 ARIMA

ARIMA is a commonly used technique utilized to fit time series data and forecasting [18]. ARIMA is an acronym of autoregressive integrated moving average, which is a generalized version of the ARMA (autoregressive moving average) technique. The whole premise of ARIMA is that information from the past values alone can be used to predict future values. Autoregression and moving average are two standalone methods for modelling stationary time series, and ARMA (and ARIMA) take a combination of the two as a ratio in order to find a better fit to the data. It was stated by Nelson [19] that ARIMA is a suitable benchmark for measuring accuracy of a forecast.

ARIMA takes three values, ***p***, ***d*** and ***q***. ***p***, ***d*** and ***q*** represent the orders of AR, Difference (how many times the dataset was differenced) and MA, respectively. The notation for these values is (***p, d, q***) for example (1, 1, 3). The general formula for ARIMA is:

$$\hat{y}_t = \mu + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} - \theta_1 e_{t-1} - \cdots - \theta_1 e_{t-q} \quad (2.3)$$

In equation 2.3, $\hat{y}_t$ is the differenced time series being calculated, μ is the intercept, $\varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p}$ are the lagged values and $\theta_1 e_{t-1} - \cdots - \theta_1 e_{t-q}$ are the lagged errors.

Before the formula can be used, the data must first be tested for stationarity. Stationarity is a condition the data must meet for a time series model to be built that can be used to predict future values, it means that the statistical properties of a time series do not change over time. In order to satisfy this condition, the data must meet the following properties:

- Mean is constant.
- Homoscedasticity: same variance at different points in the time series.
- Auto-covariance between two data points.

Once stationary data is obtained, the ***d*** parameter value will be known. The ***p*** and ***q*** parameters still require values. These can be determined by analysing the autocorrelation function plot (ACF) for the AR (***p***) parameter and the partial autocorrelation plot (PACF) for the MA (***q***) parameter [20]. Once all parameters are defined, the ARIMA model can be fit to the training data. The model can be fit using a mixture of conditional sum of squares and maximum likelihood estimation. The quality of fit of the models (if more than one ARIMA model has been fit to the data) can be compared using Akaike information criterion (AIC) and Bayesian information criterion (BIC). The trained ARIMA model can then be used to forecast the values and these forecasted values can be compared to the actual values. The difference between the forecast and the actual values can be measured using various metrics such as mean absolute deviation (MAD), mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) [21].

2.4 Equation Discovery

The method of equation discovery is an area of machine learning that automatically discovers quantitative laws. These laws are expressed in the form of equations, in collections of measured data [22]. It takes a set of system variables, and a table of observations and predicts a model M. This model is formulated as a set of ordinary algebraic or differential equations defining the target variable [22].

Although the method of equation discovery can be applied to any problem if it can be modelled by an equation, the key area of interest for this project lies in macroeconomic forecasting. There is a piece of literature by Kazakov and Tsvetomira [23] that details an experiment on modelling various macroeconomic parameters in the Euro area with ordinary equations. A follow up piece of work by Georgiev and Kazakov [24] takes the previous method and learns a set of ordinary differential equations (ODE) to model the same macroeconomic parameters on the same Eurozone data used in [23].

These two papers both use an equation discovery tool named LAGRAMGE, a piece of software that is detailed in [4]. LAGRAMGE is useful due to the features it presents. In both [23] and [24], the authors made use of the availability to add a declarative bias in order to reduce the potentially huge hypothesis space that comes with equation discovery. In LAGRAMGE, this is done by defining a context-free grammar (CFG) to restrict the hypothesis space.

In some cases, the grammar can be a representation of a previously defined model. This initial grammar would be revised, and the tool would be run to find a model that has a lower error than the initial model. This technique is called equation revision and it could be useful in the problem domain of this project due to the availability of manually generated models for macroeconomic parameters.

As the number of equations that can be generated is infinite, LAGRAMGE allows a depth limit to be set by the user to limit the search. Todorovski [4] shows an example problem of an aquatic ecosystem in which the use of the grammar stated in the paper reduces the complexity of the search space by a factor of 10^{35} compared to the complexity when a universal grammar is provided to the tool. This shows that the declarative bias is important in obtaining more optimal results as the lower complexity allows more equations to be evaluated. The minimal description length (MDL) principle can also be used in order to make the search more feasible, as this penalises equations with added complexity.

Although equation discovery for the problem set regarding macroeconomic forecasting is the primary subject of this project, it is not the only application. Todorovski in [4] detailed four problems (three synthetic and one real-world) in which the LAGRAMGE tool was applied. All the problems involved dynamic systems thus differential equations were sought by the tool. Two of the problems in Todorovski’s experiments were either out of reach (complexity-wise), or simply had not been modelled successfully from existing equation discovery systems [4]. The results from the real-world problem - phytoplankton growth in lake Glumsoe - showed that the equations discovered not only made sense from an ecological point of view, but also gave accurate short-term predictions for phytoplankton growth.

The work done by Georgiev and Kazakov – like Todorovski [4] - used LAGRAMGE’s ability to discover differential equations in order to apply it the macroeconomic modelling problem set. The results showed that the ODE models provided a good match to the previously discovered ordinary equations in [23]. With some additional functions applied, the ODEs performed better for all three variables being modelled. Implementing the ability to find differential equations is something worth trying to incorporate into this project.

Todorovski later went on to reinforce the usefulness of the LAGRAMGE tool in applying the equation discovery and revision method to real-world problems when he published a paper on revising an earth ecosystem model of the carbon net production [5]. As opposed to the previous experiments which used equation discovery, this study took a pre-defined model, transformed it into an initial grammar, then extended the grammar in order to revise the equations in the hope to find a more accurate model. The results of the experiment compared the RMSE of the initial CASA (existing model of the Earth's biosphere) NPPc (net production of carbon by terrestrial plants) model to the revised CASA-NPPc models that came out of the equation revision method. The revised models show an error reduction against the initial CASA-NPPc model of almost 20% in the best case. The experts who developed the CASA model regard this result as non-trivial [5].

In John Terry's recent project [2], when discussing future work and potential improvements on his modelling tool, he suggests incorporating additional information into the models in order to conform to policy decisions and external variables. The ability for a domain expert to be able to use the tool in order to revise the current models is important as this showed to be effective in Todorovski's work on equation revision for an Earth ecosystem in [5].

A recent publication by Brence, Todorovski and Džeroski [25] shows the effectiveness of using a probabilistic context-free grammar (PCFG) as declarative bias in equation discovery. The soft constraints give a probability distribution on the space of possible equations. It was demonstrated that the use of a PCFG leads to more efficient equation discovery in the problem set of the report.

2.5 Search Algorithms

As the approach of equation discovery is one which currently requires the generation of lots of equations creating a large search space, run time and complexity can be a problem. An improvement in navigating the large search space is to use a search algorithm. Search algorithms can be classified in many ways but the classification that applies more to the problem set at hand is informed and uninformed search algorithms.

An uninformed search is one in which the algorithm has no knowledge of the current state to a goal state. Informed search on the other hand is one which has additional information which can be used to estimate the distance from the current state to the goal state, also known as a heuristic [26]. One category of an informed search is a best-first search. Examples of best-first searches are A* search and greedy-best-first search. Examples of uninformed searches are depth-first and breadth-first [27]. Within the category of informed searches there are different algorithmic complexities and different levels of simplicity for implementation. The search which has the simplest use of a heuristic is a GBFS. This search visits the state with the smallest heuristic. This will be useful in equation discovery where the heuristic is the accuracy of the model. The issue with the GBFS is that it is not optimal. The A* search on the other hand is optimal, given that the heuristic is admissible [28].

For a heuristic to be admissible, it must never overestimate the cost of the path to the goal state from any given state. This means the heuristic must be less than or equal to the distance from the goal state [26]. If the heuristic is an overestimate, then an algorithm like A* could overlook the optimal solution in some cases. A search algorithm is complete when it is guaranteed to find a solution (from start state to goal state) if one exists. A search algorithm is optimal when it will always find the best solution if a solution exists.

GBFS is neither complete nor optimal, unlike A* which is both complete and optimal [21]. The reason that GBFS, although worse than the A* search by these metrics, may be a better choice for the problem of equation discovery is because in general it uses less memory than A*. Because A* has a space complexity of $O(b^m)$, as it stores all generated nodes in memory [29]. When compared to the GBFS which has a space complexity that can be polynomial, it can make GBFS more suitable to problems with a large search space. Equation discovery, when generating equations from a context-free grammar can very quickly build large search spaces which would cause memory problems.

The reason that these techniques apply so well to the equation discovery problem space is because they do not require a complete graph to do the search as they only look at the nearest neighbours of a state, which can be calculated as the search is running.

2.6 Linear Regression

Linear regression takes an input vector $X^t = (X_1, X_2, \dots, X_p)$ and wants to predict a real-valued output Y. The linear regression model has the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \quad (2.4)$$

The βj's are unknown parameters or coefficients and the X_j's are quantitative inputs. Typically, we have a set of training data (X_1, y_1) ... (X_N, y_N) from which we estimate the parameters β. The method relevant to this project is *least squares,* where the coefficients, $\beta = \left(\beta_0, \beta_1, \dots, \beta_p\right)^T$, are selected in order to reduce the residual sum of squares (RSS) to a minimum. The equation for the RSS is:

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j)^2 \quad (2.5)$$

In equation 2.5, y_i is the response for datapoint i, x_i is the vector of predictor values for i and $x_{i,j}$ is the value of predictor j in datapoint i. Equation 2.5 uses the distance of each data point from the regression line to calculate the best coefficients for the model. The β_0 in equation 2.5 is known as the intercept or bias coefficient. If a coefficient is fitted with the value 0, it means that the influence of the variable is removed from the model.

2.7 Source Comparison

Starting with Fildes and Stekler's paper on the state of macroeconomic forecasting [10]. This paper was published in 2002 so although not a recent publication, it still discusses many of the techniques that are still employed today such as the VAR methods discussed in [8] and [17]. This piece of research provides the breadth due to the different data it covers. It provides a comprehensive assessment of many different techniques across multiple economies over varying time periods.

In a slightly different direction, Panetta's work [11] assesses the relationship between the stock market and macroeconomic forces. Although this work looks at data in the Italian area only, it still provides a strong argument that the stock market is affected by macroeconomic factors. This helps to explain some of the motivation behind the desire to model the macroeconomic parameters.

McNees' work [8] on comparing the US macroeconomic forecasts provides some great historic background in the field. The discussion of the changeover from human crafted conditional models to unconditional statistical techniques is useful for providing motivation to model the macroeconomic parameters from scratch and to potentially revise the current models we have. Although the paper is written in 1986, so the data sets used to assess the accuracy of the alternative techniques is more limited than the data that is available today. The alternative techniques mentioned are still valid as a benchmark to assess model accuracy in recent pieces of work [2].

Building on McNees' discussion of human crafted conditional models in comparison to the statistical techniques, Webb and Lupoletti wrote a piece of work on improving the accuracy macroeconomic forecasts with contributions from a VAR model [17]. Although this paper was published prior to McNees' work [8], the background in McNees' work gave good context when reviewing this work. This paper details not only the VAR discussed prior but also introduces the ARIMA (autoregressive integrated moving average) method. The paper compares the VAR forecasts to the ARIMA techniques as well as the more complex BVAR method discussed in [8]. Over the period 1970 to 1983, the VAR model produces forecasts that are competitive with the well-known commercial forecasters. This further enhances the usefulness of a VAR model for a benchmark to assess predictive accuracy when defining a model. These results are potentially down to the small, limited data set that was available at the time of writing. It would be a more if a comparison could be made with more recently published data providing a larger range of training data for the VAR and ARIMA models.

Both Kazakov and Tsenova's equation discovery paper [2] and Georgiev and Kazakov's learning ODE's paper [19] are perhaps the two pieces of work relating to this project the most. They use an identical dataset from the Euro Area from 1971-2007. The parameters they are attempting to model are the inflation, output growth and nominal interest. The methods are similar as they both use the LAGRAMGE equation discovery tool. Georgiev and Kazakov's provides an extension to the initial work by defining a grammar which allows differential equations to be discovered.

Todorovski's use of LAGRAMGE for modelling real-world problems such as the phytoplankton growth in lake Glumsoe [4] and revision of an Earth ecosystem model [5] demonstrates the application and flexibility of the tool. In the ecosystem model, he demonstrates the way the tool can be used by domain experts to improve previous models, which is something that could potentially prove useful in the macroeconomic space. Despite these studies operating in a different domain to the one I am concerned with in this project, it adds motivation to building an equation discovery tool as this can be applied – if developed successfully – to many different areas of problems.

3 Methodology

3.1 Success Criteria

1. Creation of an alternative to LAGRAMGE with Python libraries, extending a previous attempt. This will be achieved by extending the methods used in Terry's work [2] to incorporate informed search and different declarative bias.
2. The program is easy to analyse, maintain and extend. The use of Python and modern-day IDEs allows the tool to be analysed and extended much easier. It will also be significantly easier to set the tool up in order to obtain results in a custom domain.
3. The tool accurately forecasts macroeconomic parameters. The tool will be run and generate several equations, which will be fit to the data and the best candidates for each macroeconomic parameter will be used to forecast them with a one-step-ahead forecast. Multiple baselines will be developed in order to assess the predictive accuracy of the forecasts generated by the tool.
4. The tool finds results more efficiently than the previous method. The tool will be assessed on how many equations are evaluated at runtime, against the previous implementation.
5. Versatility of the tool. This will be achieved by modifying the declarative bias into a PCFG, thus demonstrating the ability to alter the declarative bias in order to restrict the search space.

3.2 Dataset selection

One of the decisions which largely impacts the project is dataset selection. The three considerations were the quality, quantity and availability. When considering the quality of the data, how the data set will have been impacted by policy decisions was considered. In the design phase, three data sets were considered: Eurozone, United Kingdom and United States macroeconomic data. For these three sets of data, accessibility is not an issue as they are all instantly available on the internet. The Eurozone data can be harvested from the European central bank [30], UK data is published by the Bank of England [31] and the US data can be easily accessed through the Federal Reserve of St Louis [32].

The Eurozone data was the initial choice when setting up the project. This would have allowed me to assess the results from this project against the results obtained by Kazakov, Georgiev and Tsenova when using the Lagramge tool in [23] and [24]. Because Eurozone data includes data from an array of different countries, this means that the dataset over the years will have been largely affected by policy decisions. McNees' work [8] discussed the difficulties of forecasting data which has been impacted by policy decisions. One approach would be developing a method of factoring in these policy decisions to the model but because currently, the Eurozone is comprised of nineteen countries, this would be unachievable within the scope of the project. Even with the aid of an expert in the field, this may be an unrealistic goal. The Eurozone dataset used in the papers mentioned previously is outdated by fourteen years, dating up to 1971. Fourteen years of data that would not be incorporated into the training of the models would be more than a negligible amount to leave out. The other option would be to use the latest publication of Eurozone data. Although not directly comparable to previous papers, this would offset the issues raised by McNees when comparing models that are trained on a range of different datasets, as the data used in this project would merely be an extension of that used in [23] and [24]. It could be argued that a more accurate forecast would be produced from aggregating models from each different country that is part of the Eurozone instead of forecasting the Eurozone data as a whole. This would be significantly more work and may not prove any more useful for assessing the predictive accuracy of the equation discovery tool than if I were to use another dataset. Also, upon assessing the data, there are some gaps present. If this data set were to be selected, a decision would be made whether to interpolate the missing data or leave as is, resulting in less training data.

On the contrary, the Bank of England published ‘A millennium of macroeconomic data’ [31] which holds an extensive record of macroeconomic data for the UK. This data dates back several hundred years, for GDP data it holds values dating back to the year 1238. Using this data set would reduce the variance of the models estimates due to the larger set of values that can be used as training data. The data set is also very complete so it would remove the need to deal with missing values in the data like I would be required to do in the Eurozone data. This dataset contains multiple macroeconomic parameters in a single csv file, making the data is accessible. The dataset is extensive in the number of aggregates, completeness of data and the frequency of the data. Despite this, the previous work that has been researched during this project has generated models on the Eurozone and US datasets. Although there is a lack of robust techniques for comparing models that are trained and predict different datasets, results can be reproduced on the UK dataset using previous methods [2] for a comparison to be made. The dataset dates up to 2017 for the aggregates of interest. This means the data includes values over the period that a large policy decision was voted for by the British public (exiting the EU). Although this vote will have impacted the economy in some capacity, the data does not include the period when the actual exiting of the EU took place (early 2020), and it also does not include the period where the Covid-19 pandemic will have taken effect on the economy.

The US dataset does not date back as far as the UK dataset does, however it is more complete than the Eurozone data. This dataset dates to the 1940s for the aggregates that will be forecasted in the scope of this project, all the way up to current day. These aggregates are published both on a quarterly and monthly basis. Using the US dataset would allow me to make comparisons to Terry's results [2]. It would mitigate any issues that may occur when comparing two different models trained on different datasets, as much of the data used would be identical.

Upon assessing the different datasets in detail and considering several factors, the US and UK datasets have shown to be the most viable in the scope of the project. All datasets mentioned are equally as accessible, however the US and UK datasets are more complete than the Eurozone data. The Eurozone area has also been subject to large political changes over the time period that the data covers. These political changes could make non conditional predictive accuracy difficult. New countries joining the bloc and countries leaving the bloc could vastly affect the macroeconomic values. Using an aggregation of country specific models that are part of the Eurozone to build a wider model is more effective than modelling the Eurozone data available.

3.3 Dataset challenges

One challenge in the US dataset is the previous year's data. Due to the Covid-19 pandemic that was declared in March 2020, the macroeconomic variables were significantly affected due to the requirement for the US population to stay at home and some states to completely shut down any businesses. One may argue that this is out of scope of what a model should be able to achieve when as a global pandemic is mostly unforeseen in the data leading up to the event; having said this, these extreme datapoints will in fact be included. Another challenge with the data is the fact some of the more recent datapoints for the macroeconomic variables are only estimates; thus, requiring filtering. Due to the sparsity of the US dataset, issues may arise involving a lack of trainable data, in particular when splitting into train:valid:test ratios. However, due to this k-cross fold validation may prove useful in order to reduce model overfitting. Conversely, oversampling methods or interpolation (i.e. simple feedforward interpolation of time series data) of the data may also prove useful in overcoming these ailments.

When selecting data from each macroeconomic parameter, it must be taken into consideration that some datasets will be seasonally adjusted vs non-seasonally adjusted and other datasets may measure the change in a variables value vs the absolute value. This is because seasonally/non-seasonally adjusted data and stationary/non-stationary data will directly impact the effectiveness of the ARIMA models [33]. The impacts these features have can be mitigated by keeping the existence of seasonal adjustment in data consistent, however this does create an added step of complexity, and one in which the best-practice can differ between two resources.

3.4 Declarative bias selection

The declarative bias used will be a context-free grammar (CFG). This was shown in the original Lagramge paper by Todorovski. A range of CFGs will be used in this project and the one which produces the best models will be selected. The grammar used in Kazakov and Tsenova's paper [23] which was also seen in Terry's work [2] will be a starting point, allowing initial results of this project to be compared directly to the work done in [2] and [23]. This grammar will then be modified, to potentially generate some ODEs and to test if the informed search method outperforms the models found when generating ODEs in Georgiev and Kazakov's work in [24].

The issue with using CFGs in equation discovery is that the technique can easily overfit the training data by generating an unnecessarily complex equations [25]. This is a problem which the addition of a 'travel cost', which penalises the more complex equations containing a higher number of terms, in an informed search such as A* attempts to combat. Another approach towards the declarative bias that can be taken is using a probabilistic context-free grammar (PCFG) to generate the search space. As previously covered, the use of a PCFG can provide more efficient equation discovery. By nature, this will prevent the problem of equations becoming too complex and overfitting to the training data, as a probabilistic grammar can control the risk of overfitting by assigning higher probability to non-complex terms in the grammar. The effectiveness of this method in equation discovery was recently evaluated in Brence, Todorovski, Džeroski [25], however the effectiveness of the approach has not yet been evaluated in the macroeconomic domain. An example of a PCFG that would relate to the problem set we are dealing with, where more complex terms are assigned a lower probability can be seen in Figure 1.

```
E -> E Plus Term [0.4] | Term [0.6]
Term -> 'const' [0.4] | 'const' Times 'v' [0.4] | 'const' Times LT Times LT [0.05]
            | 'const' Times 'v' Times LT [0.05] | 'const' Times LT [0.1]
LT -> 'np.sin(' 'const' Times 'v' Plus 'const )' [1.0]
Plus -> '+' [1.0]
Times -> '*' [1.0]
```

Figure 1 - An Example PCFG

4 Design

4.1 Data description

The dataset contains 879 datapoints for both CPI and unemployment rate, which are sampled on a monthly basis. The dataset also contains 293 datapoint for GDP, which is sampled on a quarterly basis. These datasets have been aligned on the date column and compounded in the same csv file. The compounded dataset ranges from 01-1948 to 03-2021. This covers a period of post war which contains a lot of booms and recessions, as well as covering the great recession of 2008. The dataset also covers the months in which the Covid-19 pandemic was declared as well as the following months.

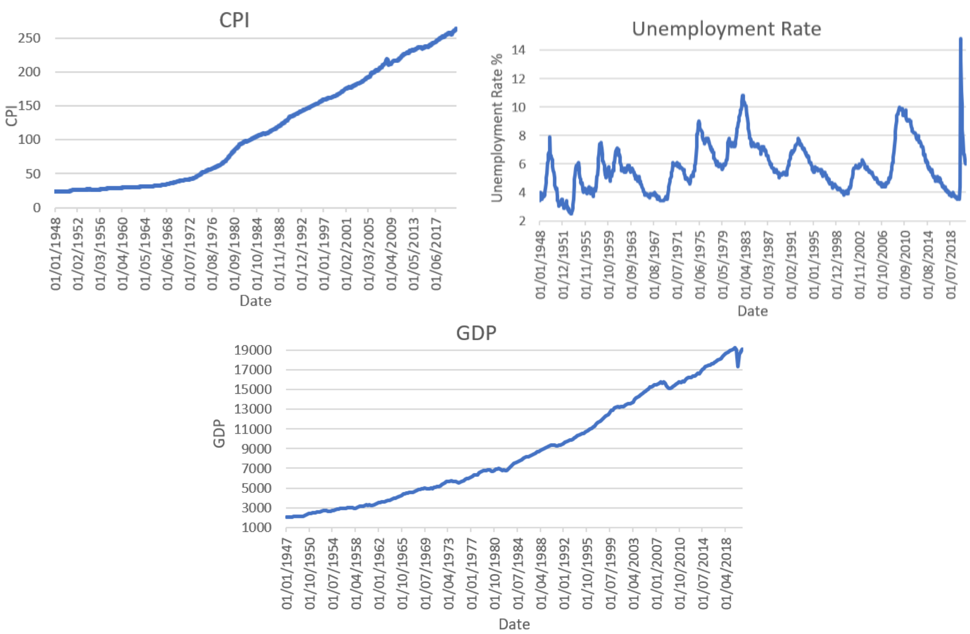

Figure 2 - Plots of Macroeconomic Parameters

4.2 Baseline model: ARIMA Usage

The reproducibility of the ARIMA method makes it a great comparison to any results obtained with the equation discovery tool, as custom ARIMA models based on relevant data can be utilised without having to compare to forecasts predicted from only macroeconomic data spanning 1980-1985. The availability of an extra 40 years of data should produce a more accurate model and provide a better baseline, allowing comparison with equation discovery defined models; this remains consistent with Terry [2].

In order to determine that stationarity of the data in a reliable way to ensure accurate forecasts, there are a few concrete methods, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) augmented dickey-fuller (ADF) test and the Phillips-Peron (PP) test [34]. These methods are known more formally as unit-root tests. The ADF test assumes a null hypothesis that the-time series data is non-stationary, if this null hypothesis is rejected with a high level of certainty (critical level), then it suggests that the data is stationary. On the contrary, the KPSS test assumes that the data is stationary as a null hypothesis. Rejection of the null hypothesis suggests that the data in non-stationary. The KPSS test can also check for stationarity of a series around a deterministic trend. If a series is stationary according to the KPSS test but is non-stationary according to the ADF test, it suggests that the time series is stationary around a deterministic trend and so is easy to model this series and produce accurate forecasts. It is for this reason that in the case of an ADF test result that displays non-stationarity of the data, a follow up KPSS test will be run to ensure that the data is not stationary around a deterministic trend. If both tests show non-stationary data, the data will then be differenced, and the tests will be applied again. The PP test is an adaptation of the ADF test which is robust to homoskedasticity and autocorrelation. I will not be using the PP test as it was shown in Davidson and MacKinnon [35] that the Phillips–Perron test performs worse in finite samples than the ADF test. As financial time series datasets are often small as data is taken quarterly or monthly, ADF is the better option here.

CPI data will be used to demonstrate how the ARIMA method is applied, further results for the remaining macroeconomic variables will be discussed in the results section of the report. As shown in Figure 3, the ADF statistic for the CPI data time series is 2.05 which is higher than even the critical value at a 10% confidence level and the p-value is > 0.1.

```
ADF STAT: 2.053085
p-value: 0.998738
Critical Values:
        1%: -3.438
        5%: -2.865
        10%: -2.569
```

Figure 3 – ADF test results

```
KPSS Statistic: 4.058667401606447
p-value: 0.01
num lags: 21
Critial Values:
   10% : 0.347
   5% : 0.463
   2.5% : 0.574
   1% : 0.739
```

Figure 4 - KPSS test results

To be certain, a KPSS test was run, and the results further reinforced non-stationarity in the CPI time series. These results can be seen in Figure 4, displaying a KPSS statistic value of 4.05 and a p-value of 0.01, significantly less than the 10% critical value of 0.347. In order to accept the null hypothesis, which assumes the data is stationary around a deterministic trend, the p-value would need to be greater than the critical values of the significance level appropriate.

The non-stationarity of CPI time series data is obvious from the trend of the data as seen in Figure 2. In order to better demonstrate why a statistical test such as ADF or KPSS must be used, the results for the analysis of US unemployment rate are demonstrated. When plotting the unemployment rate time series in Figure 2, there is no clear signs of non-stationarity. There is potentially some seasonality in the data however it is difficult to be certain. The results from an ADF test give a p-value of -.0014 and a statistic of -3.99, less than the 1% critical value of -3.438. Therefore displaying, with a 1% significance level, that the data is stationary, and the null hypothesis can be rejected.

```
mean1=4.752218, variance1=1.408092,
mean2=6.748288, variance2=1.705100,
mean3=5.806143, variance3=3.604194
```

Figure 6 - Mean and Variance of Data Splits

```
ADF STAT: -21.276079
p-value: 0.000000
Critical Values:
        1%: -3.438
        5%: -2.865
        10%: -2.569
```

Figure 7 - ADF Test Results, Differenced

```
mean1=0.007877, mean2=-0.001706, mean3=0.003425
variance1=0.066753, variance2=0.037983, variance3=0.439098
```

Figure 5 - Mean and Variance of Data Splits, Differenced

For concreteness, a KPSS test was also run which flagged up some interesting results. The test gave a KPSS statistic of 0.199 and a p-value of 0.016 which is lower than the 10% significance level critical value required to accept the null hypothesis. Contrary to the ADF test, this suggests that the time series data is not stationary. This provides some scepticism on the stationarity of the data and in order to determine if the data is stationary, other techniques can be used. To test homoscedasticity, data can be separated, and the mean and variance can be calculated in order to check for consistency across all periods in the data [36]. A true stationary set of data would have near-constant mean and variance at all time periods. When the data is split into three chunks, the mean and variance is not constant, and the range of values is great enough to suggest that the data is not stationary, and that the data should be further differenced. These values can be seen in Figure 4. The PACF and ACF plots of the data can also be analysed in order to check for autocorrelation. These plots can be seen in Figure 10.

Upon differencing the data once, you can clearly see that the results of the ADF test are much more convincing and the mean and variance throughout the data remain somewhat constant when compared to the previous values in Figure 6.

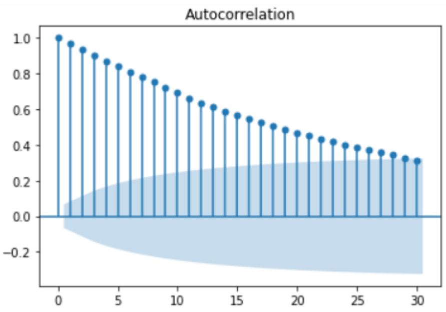

Figure 8 - ACF Plot of Non-differenced Data

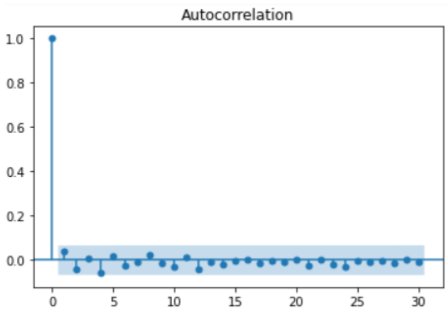

Figure 9 - ACF Plot of Differenced Data

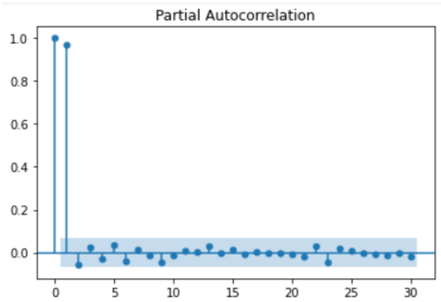

Figure 10 - PACF Plot of Non-differenced

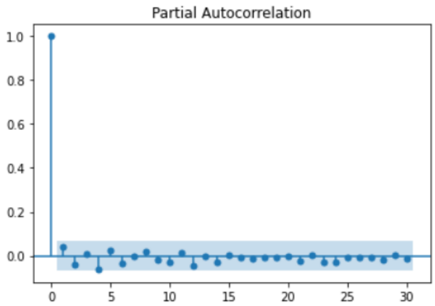

Figure 11 - PACF Plot of Differenced

The KPSS test also shows stationarity on the differenced data. This suggests with a good level of statistical significance as well as judgement from viewing the plotted data, that the differenced data is now stationary.

Furthermore, in order to select the ***p*** (AR) and ***q*** (MA) terms for the ARIMA model, the PACF and ACF plots are used. As shown in Figure 11, the first lags in the ACF and PACF plots are significant, the following lags are not. This suggests using an ARIMA model of (1,1,1). This aligns with Nau's documentation [33] on selecting AR and MA terms for the model, stating that ", the sum of p and q will generally be no larger than 3" and that a ***p*** or ***q*** value larger than 3 should almost never be used.

4.3 Heuristic Selection

In order to implement an informed search through the equations to optimise the brute force approach to equation discovery, a heuristic must be selected. A candidate for a heuristic, given that states are equations, is the error on the equations when modelled against the data. Although an equation with an error of 0 would be seriously overfitting the data, we can use this as a goal node as it will result in the search minimising the error and in the case of overfitting, a previous equation can be selected as the top ten scoring equations will be stored in the population-based approach. In terms of which error metric to use, MAD, MSE, RMSE and MAPE are all useful. As MSE is more impacted by larger errors, it may be the best option as this will show how well the model picks up changes in the trend of the data, therefore providing a good estimate of the effectiveness of an equation when fit to the data. Whether the heuristic is admissible is up for debate as this depends on the actual cost to a goal node which is defined using the complexity of the equations. The error could have several standard deviations subtracted from it in order to provide underestimation to a reasonable probability. It is however the case that informed searches such as A* can still find the best results without an admissible heuristic, however this does mean that if A* was to be implemented, it would not be optimal.

Another choice is what values will represent the distances between states (travel cost). The theory of minimum description length (MDL) can be implemented and the increase in complexity of the equations from one state to another could be measured, reflecting in an increase of the distance between the states. Complexity can be defined as the number of terms added as well as the type of terms that are added i.e. polynomial or trigonometric. Because A* finds $f(n) = g(n) + h(n)$, where $g(n)$ is travel cost and $h(n)$ is heuristic value, the heuristic and the travel cost should be within the same order of magnitude otherwise one will dominate the other. In order to adhere to this, the travel cost and heuristic values should be scaled.

4.4 Linear Regression usage

Ordinary least squares (OLS) is well implemented by multiple libraries available in the python language. This means that implementing from scratch will not be a requirement, saving time and providing an efficient implementation that is designed with run time in mind. OLS also treats the data as a matrix and uses linear algebra to estimate optimal values for coefficients. Linear regression can also be applied to higher order polynomials, these higher order polynomial terms can be added in order to get a better fitted model in some cases. In Python, this can be implemented using libraries such as SciPy [37]; namely using the *curve_fit* method. There are other methods of linear regression that introduce shrinkage parameters in order to penalise larger coefficients. These methods are known as ridge regression and lasso regression (L1 and L2 penalty resp.). These methods take an ‘shrinkage ’parameter in order to minimise overfitting [38]. These methods will prove useful if the data overfits substantially. Aside from this, the project will use OLS in order to prioritise efficiency and minimise the number of variables that can be altered (i.e. shrinkage parameter).

Unlike the previous work [2], the algorithm will be slightly different in this project due to the implementation of an informed search in the tool. The algorithm itself iterates over each of the macroeconomic parameters (CPI, GDP, Unemployment rate), generates a set of equations from the CFG that is used as a declarative bias. Once the equations are generated, they will be fitted using OLS linear regression with higher order polynomial terms added. The MSE will then be measured of the equations. A greedy best-first search will then be run on the discovered equations. The nodes will be the equations, the heuristic will be the MSE and the values on the edges of the graph will be how many more terms the equation has than the origin equation from the previous node. As the equations are fitted, their respective MSE values will be calculated against unseen data and the top ten equations regarding MSE will be stored. MSE is being used again here in order to select equations that can react to sharp changes in the data. This is a population-based method which allows selection and expansion of equations other than the best scoring equation. This is useful as the top scoring equation may be overfit or have other issues which make it unsuitable for selection or expansion.

In the LAGRAMGE tool, there exists no method of reducing overfitting via splitting the data into a train:valid:test ratio. This also opens the door for cross-validation to be implemented, although this comes with more computational overhead, the informed search method should vastly reduce the computation needed to get the results. As the period of data used for training will contain the post second World War recessions and recoveries [39], it may cause the equations generated to be overfit to these movements in the data. Because of this, the set of training data has been reduced so that it is equal to the size of both the validation set and the testing set. This means 55% of data discarded, and the remaining data is split equally between the train:validation:test respectively.

4.5 Previous Tool Usage

The previous equation discovery tool built by John Terry [2] was able to be run on the data used in this project. Although Terry published the results when using the tool on a similar dataset, rerunning the tool on the same data that will be used in this project allows for a reliable direct comparison in accuracy of results and performance. This was, things like run time can also be measured using profiling, as this was an area of the tool that was causing some problems previously when fitting the equations that had been generated to a higher depth. The results from running Terry’s equation discovery tool will be directly compared to the adaptations of the equation discovery tool which uses an informed search, as well as the version of the tool which uses a PCFG. These comparisons will be made using forecasting accuracy. The improvement in time and space complexity will also be measured between the versions.

5 Results and Evaluation

5.1 Full Dataset

CPI

Method	MAD	MSE	RMSE	MAPE
ARIMA	4.519084	26.78020	5.174959	0.018440
Equation Discovery	9.353469	134.3767	11.5921	0.039112
GBFS Equation Discovery	8.101712	121.6289	11.02855	0.034477
PCFG Equation Discovery	9.95259	122.6219	11.07348	0.040963

Table 1 - Results of CPI Forecasts on a Full Dataset

GDP

Method	MAD	MSE	RMSE	MAPE
ARIMA	756.6549	835648.7	914.1382	0.041752
Equation Discovery	1011.121	1236877	1112.15	0.059405
GBFS Equation Discovery	524.2712	471619.5	686.7456	0.03042
PCFG Equation Discovery	538.9758	449991.2	670.8138	0.030844

Table 2 - Results of GDP Forecasts on a Full Dataset

Unemployment

Method	MAD	MSE	RMSE	MAPE
ARIMA	4.410025	24.57246	4.957061	0.914858
Equation Discovery	1.859409	5.33473	2.309704	0.293749
GBFS Equation Discovery	1.94525	5.071845	2.252076	0.373011
PCFG Equation Discovery	1.997259	5.345245	2.311979	0.387263

Table 3 - Results of Unemployment Forecasts on a Full Dataset

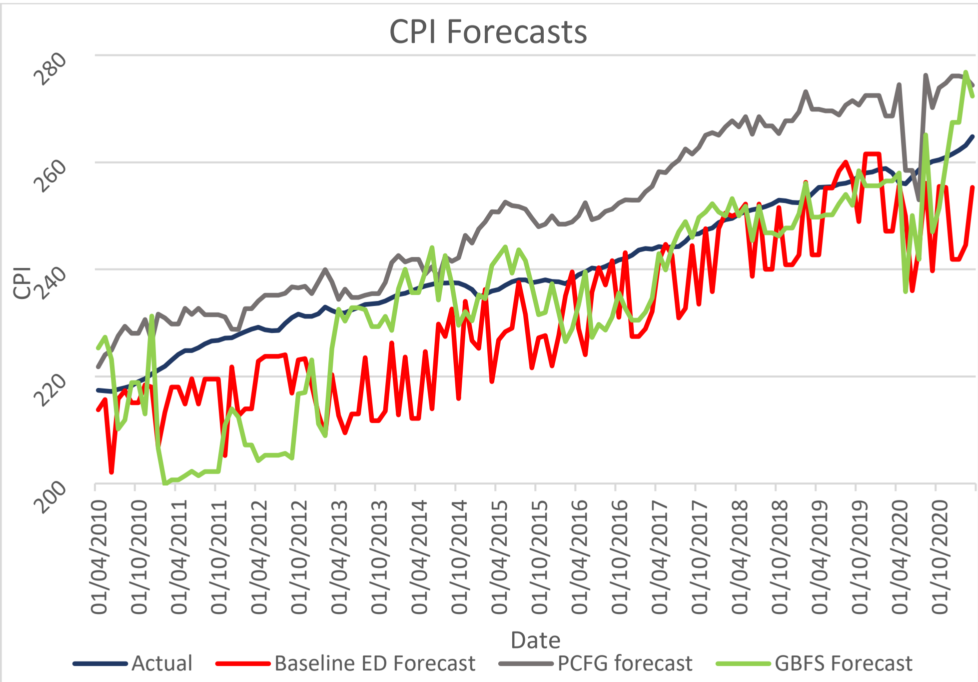

Figure 12 - Chart of Forecasts (Full Dataset) Against Actual Data

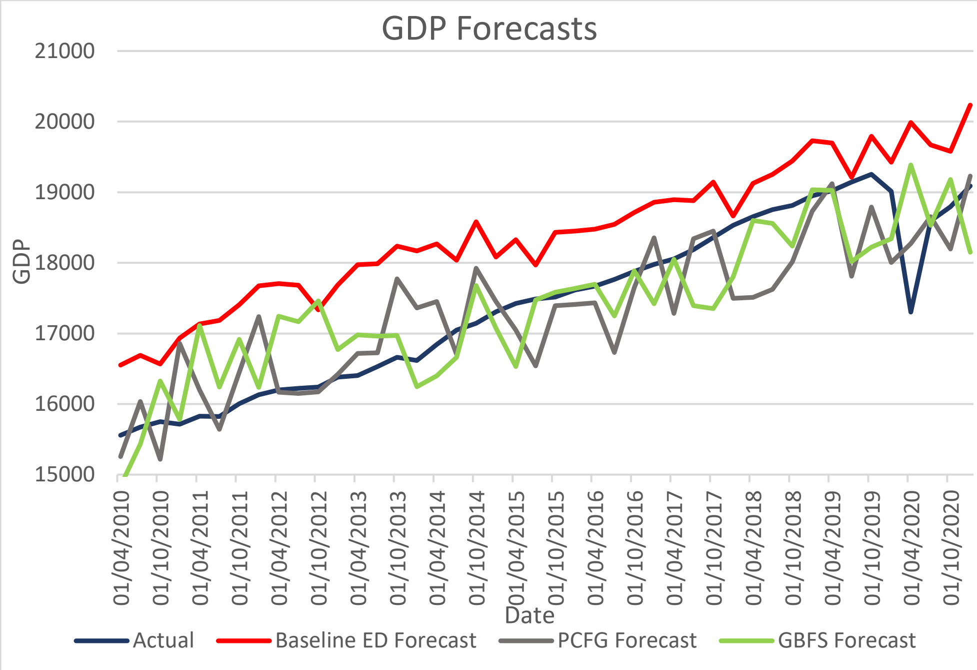

Figure 13 - Chart of GDP Forecasts (Full Dataset) Against Actual Data

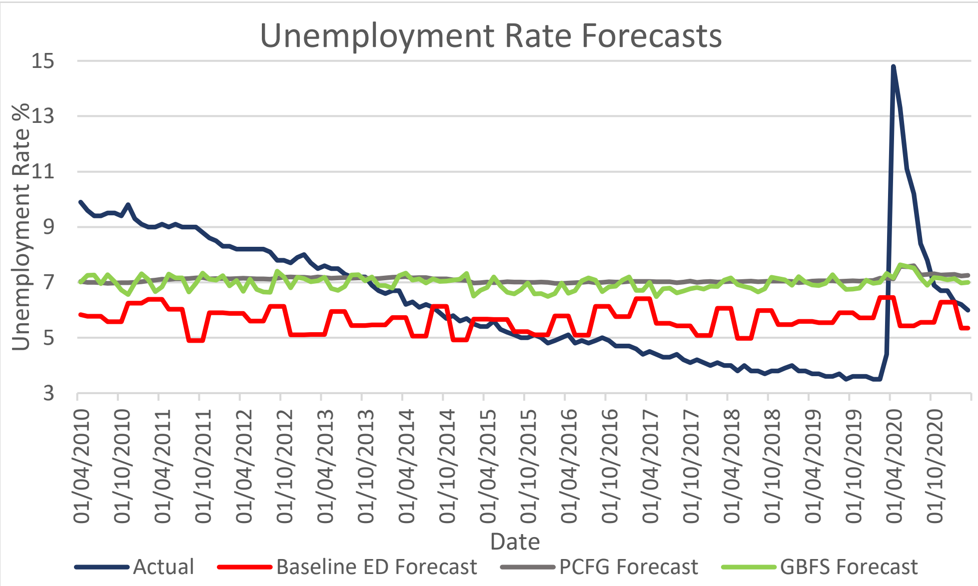

Figure 14 - Chart of Unemployment Rate Forecasts (Full Dataset) Against Actual Data

5.2 Reduced Dataset

CPI

Method	MAD	MSE	RMSE	MAPE
ARIMA	3.276294	15.13776	3.890728	0.013386
Equation Discovery	5.281494	31.96695	5.653932	0.02185
GBFS Equation Discovery	5.033653	29.64441	5.444668	0.020815
PCFG Equation Discovery	5.237873	31.2483	5.590018	0.021661

Table 4 - Results of CPI Forecasts on Reduced Dataset

GDP

Method	MAD	MSE	RMSE	MAPE
ARIMA	348.0719	214443.2	463.0802	0.019207
Equation Discovery	538.9758	449991.2	670.8138	0.030844
GBFS Equation Discovery	335.8288	215768.9	464.5094	0.019297
PCFG Equation Discovery	327.771	208857.2	457.009	0.018872

Table 5 - Results of GDP Forecasts on Reduced Dataset

Unemployment

Method	MAD	MSE	RMSE	MAPE
ARIMA	5.185499	33.66613	5.802252	1.072448
Equation Discovery	1.997259	5.345245	2.311979	0.387263
GBFS Equation Discovery	2.372017	6.8916	2.625186	0.473406
PCFG Equation Discovery	2.379256	6.916884	2.629997	0.474338

Table 6 - Results of Unemployment Rate Forecasts on Reduced Dataset

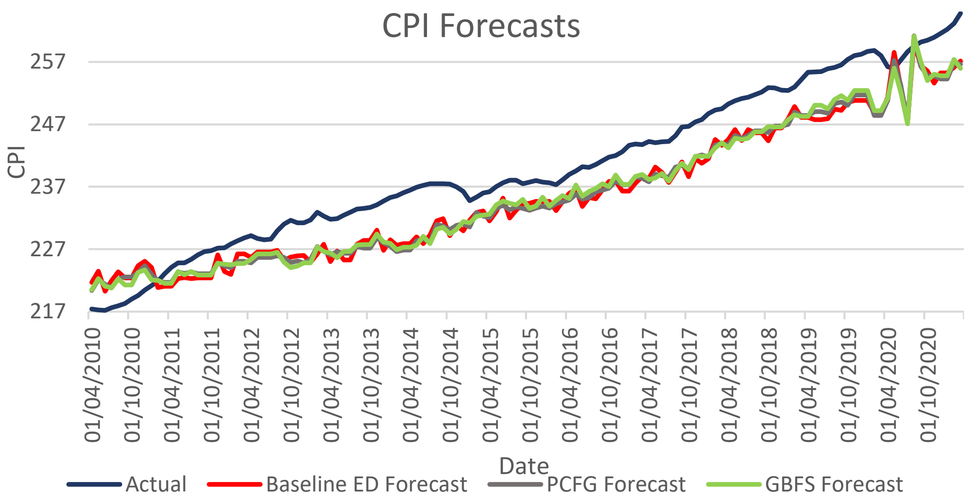

Figure 15 - Chart of CPI Forecasts (Reduced Dataset) Against Actual Data

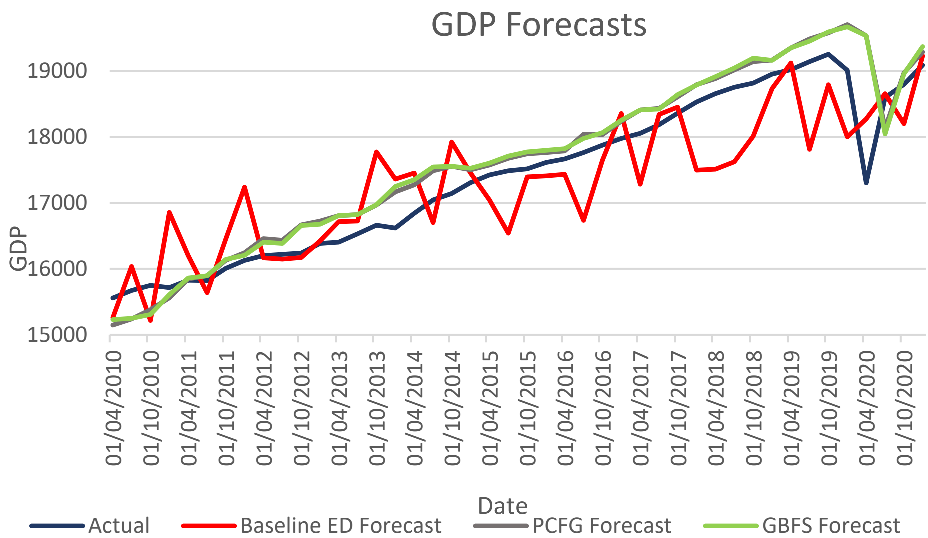

Figure 16 - Chart of GDP Forecasts (Reduced Dataset) Against Actual Data

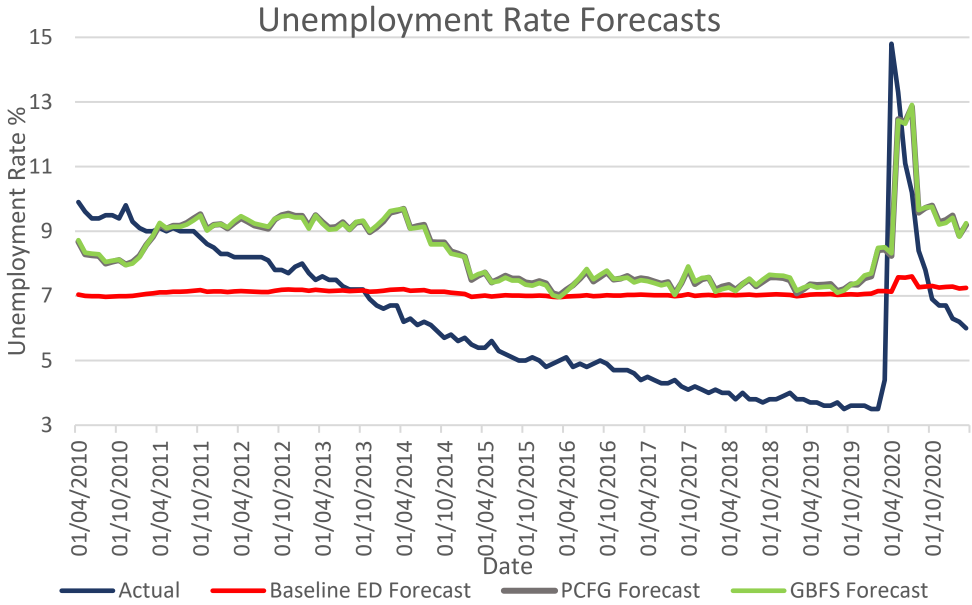

Figure 17 - Chart of Unemployment Rate Forecasts (Reduced Dataset) Against Actual Data

If the charts (above) are difficult to interpret, separate graphs for each forecast are located from Figure 24 in the Appendix, sections A.3 to A.5.

5.3 Baselines ARIMA Evaluation

On the full dataset, the ARIMA model for CPI was trained using parameters (1,1,3). GDP was trained using the parameters (1,1,2) and Unemployment was trained using the parameters (1,1,1). The results produced can be seen in Table 1 and full summaries of the models can be seen in Figures 18, 19 and 20 in the Appendix.

On the reduced dataset, the ARIMA model for CPI was trained using parameters (1,1,3). GDP was trained using the parameters (1,1,2) and Unemployment was trained using the parameters (1,1,1). The results produced can be seen in Table 4 and full summaries of the models can be seen in Figures 21, 22 and 23 in the Appendix.

5.4 Baselines Equation Discovery

When training on a full dataset, the previous tool, generating equations to a depth of 6, the tool achieved an RMSE of 11.6, 1112.15, 2.31 for CPI, GDP and unemployment rate respectively.

When training on a reduced dataset ranging from 01-04-1988 to 01-03-1999, the previous equation discovery tool, generating equations to a depth of 6, the tool achieved an RMSE of 5.65, 670.8, 2.31 for CPI, GDP and unemployment rate respectively.

The full set of equations, discovered by the tool, and used to forecast the parameters can be located in the appendix section A.2.1.

5.5 Equation Discovery Informed Search

5.5.1 Greedy best first search

The GBFS training on the full dataset performed well when forecasting the CPI and GDP values. All performance metrics show that the model was the best performer out of the equation discovery methods for CPI. The CPI forecast however did fail to outperform the baseline ARIMA forecast, by a non-negligible margin on all of the metrics. The equations for CPI and GDP appear to outperform the previous equation discovery tools equation as well as generating much fewer equations in doing so. The baseline equation discovery tool would have needed to generate equations to a depth of 7 (324 equations) which is unfeasible, even on a high spec desktop by current standards, due to the exponential complexity of the program. The GBFS method however was only required to generate 28 equations showing a great increase in performance on run time and predictive accuracy.

The GBFS discovered an equation for unemployment rate which did not outperform the previous equation discovery tool’s equation on MAD and MAPE, however it did on MSE and RMSE. This suggests that the equation discovered by the GBFS method may be slightly worse as finding a general trend, however it may perform better when picking up larger and more sudden changes in value. This is because the MSE and RMSE metrics are more heavily affected by larger errors which would most likely occur in the case of sudden changes in the data that deviate from the general trend. The model for unemployment rate appears to have outperformed the baseline ARIMA model showing a significant improvement in MAD and MAPE.

As well as observing the performance metrics, it is also worth considering the plots of the forecasts against the actual vales. You can see that the equations pick up general trends well however the CPI and GDP predictions are much more volatile than the actual data. This is perhaps the opposite in the unemployment rate forecasts as the equation seems to struggle to predict the downwards trend from 2010 up to 2019. Unlike it was suggested in the performance metrics, that the equation was able to pick up on sharp moves, the equation remains too stable throughout.

When training on a reduced dataset starting at 01-01-1988, the GBFS method produced equations that performed well when forecasting all three macroeconomic variables. The forecasts for CPI and GDP outperformed the baseline equation discovery tool forecast by a good margin, however the unemployment forecasts did not – as the previous equation discovery tool scored better on all performance metrics here. The CPI model also failed to outperform the ARIMA model by a non-negligible amount. The GDP model did show slight improvement on the baseline ARIMA model in terms of MAD however, the ARIMA model scored slightly better across MSE. RMSE and MAPE. Despite scoring worse than the baseline ARIMA model, the difference in results is negligible and the predictive accuracy of the GBFS and ARIMA model can be considered equivalent based off of the metrics. This means that the forecasts beat the previous equation discovery method baseline on most macroeconomic variables considered. The equations discovered for CPI and GDP using this method were found by generating only 24 equations, and the unemployment rate equation was discovered by generating 28 equations. This shows an example of where the reduced run time comes at the cost of predictive performance, as the more performant baseline equation discovered by the brute force approach was located in a part of the search space not explored by the greedy search. Although the models perform worse or just as good as the baseline ARIMA models, they may still pick up the sudden movements in the data which would have better real-world application in the case of a scenario such as the Covid-19 pandemic when the unemployment rate rapidly spiked. This can be seen in the plots of the forecasts against the actual data.

When looking at Figure 12, you can clearly see that the forecasts pick up sudden changes much better than the respective forecasts from models trained on the full dataset. This suggests that the models were overtrained to the data pre-dating 1988, which includes the post-World War two periods of multiple economic recessions and booms. It appears that removing this data has improved the predictive accuracy of the models seen in Table 4, which is reinforced by the appearance of the forecasts when plotted against the actual values. The improvement in all performance metrics on CPI and GDP forecasts is non-negligible.

5.5.2 A* Search

When implementing an A* search, which incorporates not only a heuristic value but also a travel cost ($g(n)$), there was difficulty with the magnitude of the values. As the travel cost was determined by the ‘complexity’ of the equation, in this case the number of terms, the value was often too low to impact the f(n) value. Although the travel cost could have been scaled up, so that it had impact when dealing with the values of GDP in the order of $1 * 10^4$, it would have caused the travel cost to dominate any heuristic when dealing with the lower CPI and unemployment values. Because of the differences in order of magnitude when implementing the travel cost, the A* search gave the same results as the greedy best first search in 5.5.1. A future piece of work may be experimenting with implementations of MDL and other ways of representing travel cost between two equations, in order to prevent equations becoming large and complex causing overfitting. As overfitting has been mitigated in the other experiments using a validation set and modifying the way in which equations are generated, the lack of a travel cost is compensated. Because the heuristic value is also non-admissible, the A* search method is non-optimal. A* may also have resulted in memory issues when running to a greater depth as the space complexity is exponential [29].

5.6 Equation Discovery with Probabilistic CFG

The PCFG used when generating the search space of equations for this experiment can be seen in Figure 1.

When training on the full dataset, the PCFG equation discovery tool finds much lower complexity equations than those discovered in the GBFS tool. This is due to the nature of how the equations are generated, in an attempt to find equation that are less overfit to the training data. The PCFG model for CPI fails to beat both baseline models, as well as failing to beat the equation discovery baseline for the unemployment rate model. Despite this, the PCFG model for GDP shows good predictive accuracy according to the metrics and significantly outperforms both baseline models here, only been narrowly beaten by the GBFS model.

When looking at Figure 12 and Figure 14, you can see that the models overreact to a small drop in the CPI data in mid-2020 yet fail to react well to the spike in unemployment rate around the same time. This is perhaps due to the underfitting of the model which occurs by oversimplifying the equations generated with the probabilistic constraints on the CFG.

The PCFG forecasts when using models trained on the reduced dataset perform far better than the same forecasts trained on the full dataset for the majority of the variables. The model beats both the baseline equation discovery and ARIMA models on the GDP forecasts, yet fail to perform better than the ARIMA model for CPI and the baseline equation discovery model for unemployment rate. Despite this, the models appear to perform better than at least one of the baselines on each of the macroeconomic variables. Interestingly, the forecast shows better predictive accuracy on the GDP forecast than the GBFS model according to all the metrics, despite the discovered equation containing far less terms.

When assessing the plots of the forecasts, you can see that not only do the models for all variables pick up the general trend, but they react very well to short term aggressive moves seen in mid-2020 amidst the pandemic. The forecasts clearly perform better than the respective PCFG forecasts when trained on a full dataset. This suggests that even though enforcing probabilities on the productions of the grammar to reduce overfitting, the models were still heavily affected by the periods of economic recession and boom following the second World War.

5.7 Locked Ordering and Lex-leader Method

Due to the large time complexity when fitting all the permutations of different macroeconomic variables, it is a part of the method which causes the largest delay. The run time implications were discussed in Terry’s work [2]. In order to reduce this, when adding terms to the best scoring equation to deepen the search space, whichever macroeconomic variables that were in place before expansion are locked. This means that despite the equations becoming quite complex, the number of permutations is greatly reduced. When running this method however, the improved run time comes at the cost of predictive accuracy. Although there are other methods of reducing the run time here, by ordering the variables and breaking the static symmetry using the Lex-leader method detailed in [40], it was decided that this is out of scope of the project given the time restrictions and the other methods being explored. Although the complexity of the current method causes problems when equations grow large, exploring to depths further than currently possible would very likely result in heavily overfitted equations. This means the motivation to explore any further into the search space of possible generated equations was reduced.

6 Conclusion

The project was successful in meeting the main objective of developing an equation discovery tool alternative to LAGRAMGE with modern Python libraries. This is because the tool, written in Python 3.8 allows for ease of extension, as well as being easier to port as Python 3 [6] is a language that is available multiple operating systems. The tool is also easier to extend as it is written with modern Python libraries with highly accessible and readily available documentation to accompany them.

The tool was used to forecast three US macroeconomic variables. These results were compared not only to the previous tool from Terry [2], but also to a baseline ARIMA model that was developed. The results proved to be equal to or an improvement on the baselines in most cases, fulfilling the criteria to forecast accurately enough to compete with modern day methods. Two methods of extension from the previous method [2] were successfully implemented, one which modified the declarative bias used to generate the search space (PCFG), and one which allowed a more efficient traversal of the search space (GBFS). The PCFG method successfully demonstrated the ability to restrict the search space generated by the tool, by modifying the declarative bias, this fulfilling the aim of versatility in the tool. The GBFS method, showed improved or equivalent predictive accuracy to the previous method, while evaluating significantly less equations in doing so. In Terry [2], it was stated that generating equations to a depth 6 took around 20 minutes on a powerful desktop computer. The equations discovered by the GBFS method would have required the previous method to generate equations to a depth of 7, which would take exponentially longer. The GBFS managed to discover the equations in less than 2 minutes on a high performant laptop, as the search used only needed to fit 28 equations instead of 324. This speed up did come at the cost of predictive accuracy in some of the variables, however the difference was marginal and the ability to search for equations at a greater depth balanced this out. This shows an improvement in efficiency against previous attempts, satisfying the criteria to find results more efficiently.

Unfortunately, cross-validation was not implemented as in order to implement a k-fold cross-validation, it would require the training and validation to be done once for each fold. Despite the improvement on run time from the previous equation discovery tool, the tool is still too slow to feasibly run to the depth required multiple times for each equation. There is also the implication that, in k-folds cross-validation, the training set may contain values at dates chronologically later than the validation set, meaning that the model would be using future data to predict past data.

6.1 Future Work

Although the tool was not tested on other economies, the tools produced would be capable of doing so. The tool could be model not only other economies but also other fields of work, previously done on the LAGRAMGE tool in Tsenova [5]. A key area of interest would be modelling the UK macroeconomic indicators in order to see if the tool could generate models which pick up on the Kondratiev wave patterns detailed in Lloyd-Jones [41].

Other variations of informed search could be implemented in the tool, and a method of achieving an admissible heuristic in A* could be produced to make the search optimal. This would be beneficial as the GBFS occasionally missed the best equation in the depth it was searching. Time complexity of the tool is still exponential in parts of the algorithm. This can be reduced by implementing the Lex-Leader method. Adding save states in order to resume computation later would allow the tool to be run to a higher depth, without the risk of progress being lost if computation were interrupted.

As the tool is written in Python 3 [6], it opens the door to improve the usability of the tool as a user. On the usability aspect, a command line interface or graphical user interface would give the experience of using the tool much more clarity.

The beginning of exploration into the usage of PCFGs as a method of declarative bias has been explored in this project. Despite the effectiveness already been evaluated in Brence [25], this project has shown that the results produced can match or improve on baseline methods of forecasting. The use of PCFGs can now be extended and further explored.

Appendix A

A.1 ARIMA Model Summaries

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	747
Model:	ARIMA(1, 1, 3)	**Log Likelihood**	-164.213
Method:	css-mle	**S.D. of innovations**	0.301
Date:	Sat, 12 Jun 2021	**AIC**	340.426
Time:	15:22:20	**BIC**	368.122
Sample:	1	**HQIC**	351.100

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2345	0.099	2.358	0.018	0.040	0.429
ar.L1.D.y	0.9943	0.005	188.791	0.000	0.984	1.005
ma.L1.D.y	-0.4239	0.039	-10.962	0.000	-0.500	-0.348
ma.L2.D.y	-0.4023	0.034	-11.981	0.000	-0.468	-0.337
ma.L3.D.y	-0.1132	0.037	-3.067	0.002	-0.186	-0.041

Figure 18 - ARIMA (1,1,3) Model Summary for CPI (Full Dataset)

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	248
Model:	ARIMA(1, 1, 2)	**Log Likelihood**	-1367.096
Method:	css-mle	**S.D. of innovations**	59.918
Date:	Sat, 12 Jun 2021	**AIC**	2744.192
Time:	15:22:52	**BIC**	2761.760
Sample:	1	**HQIC**	2751.264

	coef	std err	z	P>|z|	[0.025	0.975]
const	53.8169	8.082	6.659	0.000	37.976	69.658
ar.L1.D.y	0.6035	0.159	3.800	0.000	0.292	0.915
ma.L1.D.y	-0.2416	0.169	-1.425	0.154	-0.574	0.091
ma.L2.D.y	0.0887	0.093	0.950	0.342	-0.094	0.272

Figure 19 - ARIMA (1,1,2) Model Summary for GDP (Full Dataset)

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	747
Model:	ARIMA(1, 1, 1)	**Log Likelihood**	125.067
Method:	css-mle	**S.D. of innovations**	0.205
Date:	Sat, 12 Jun 2021	**AIC**	-242.134
Time:	15:22:03	**BIC**	-223.669
Sample:	1	**HQIC**	-235.018

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0095	0.016	0.575	0.566	-0.023	0.042
ar.L1.D.y	0.8656	0.032	27.414	0.000	0.804	0.927
ma.L1.D.y	-0.7031	0.040	-17.444	0.000	-0.782	-0.624

Figure 20 - ARIMA (1,1,1) Model Summary for Unemployment Rate (Full Dataset)

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	264
Model:	ARIMA(1, 1, 3)	**Log Likelihood**	-169.033
Method:	css-mle	**S.D. of innovations**	0.459
Date:	Sat, 12 Jun 2021	**AIC**	350.065
Time:	15:22:33	**BIC**	371.521
Sample:	1	**HQIC**	358.687

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3800	0.031	12.223	0.000	0.319	0.441
ar.L1.D.y	0.7648	0.248	3.082	0.002	0.278	1.251
ma.L1.D.y	-0.2308	0.261	-0.885	0.376	-0.742	0.281
ma.L2.D.y	-0.3588	0.118	-3.035	0.002	-0.590	-0.127
ma.L3.D.y	-0.1539	0.068	-2.274	0.023	-0.286	-0.021

Figure 21 - ARIMA (1,1,3) Model Summary for CPI (Reduced Dataset)

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	87
Model:	ARIMA(1, 1, 2)	**Log Likelihood**	-497.730
Method:	css-mle	**S.D. of innovations**	73.737
Date:	Sat, 12 Jun 2021	**AIC**	1005.461
Time:	15:23:51	**BIC**	1017.790
Sample:	1	**HQIC**	1010.425

	coef	std err	z	P>|z|	[0.025	0.975]
const	75.5532	16.078	4.699	0.000	44.042	107.065
ar.L1.D.y	0.5404	0.298	1.811	0.070	-0.045	1.125
ma.L1.D.y	-0.1633	0.313	-0.521	0.602	-0.777	0.451
ma.L2.D.y	0.1107	0.167	0.663	0.507	-0.216	0.438

Figure 22 - ARIMA (1,1,2) Model Summary for GDP (Reduced Dataset)

ARIMA Model Results

Dep. Variable:	D.y	**No. Observations:**	264
Model:	ARIMA(1, 1, 1)	**Log Likelihood**	144.717
Method:	css-mle	**S.D. of innovations**	0.140
Date:	Sat, 12 Jun 2021	**AIC**	-281.434
Time:	15:21:47	**BIC**	-267.130
Sample:	1	**HQIC**	-275.686

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0179	0.027	0.659	0.510	-0.035	0.071
ar.L1.D.y	0.9367	0.031	30.421	0.000	0.876	0.997
ma.L1.D.y	-0.7916	0.049	-16.225	0.000	-0.887	-0.696

Figure 23 - ARIMA (1,1,1) Model Summary for Unemployment Rate (Reduced Dataset)

A.2 Discovered Equations

A.2.1 Original Equation Discovery

Full Dataset

$$\boldsymbol{CPI} = -17.125 * sin(210761.99 * Unemp + 555702.511) * sin(-210761.759 * Unemp + -555694.084) + 0.0136294630 * Gdp$$

(A.0.1)

$$\boldsymbol{GDP} = 76.98 * Cpi + 164.048 * sin(-993.418 * Unemp + 6955.119) + -327.871 * sin(-1565.662 * Unemp + 3866.587) * sin(334.221 * Unemp + 4459.613)$$

(A.0.2)

$$\boldsymbol{Unemp} = -623005.744 + -0.808 * sin(-64.206 * Gdp + 302505.624) * sin(66.206 * Gdp + -302505.609) + 623011.43205$$

(A.0.3)

Reduced Dataset

$$\boldsymbol{CPI} = 2.717 * Unemp + 0.0126 * Gdp + -1.219 * sin(-11.634 * Unemp + 75.785)$$

(A.0.4)

$$\boldsymbol{GDP} = -203.787 * Unemp + 79.111 * Cpi + -106.042 * sin(-3793.462 * Unemp + -227.363) * sin(4390.370 * Unemp + -3700.914)$$

(A.0.5)

$$\boldsymbol{Unemp} = 0.243 * Cpi + 0.261 * sin(1034.701 * Cpi + 2027865.11) * sin(-1032.798 * Cpi + -2027848.33) + -0.00287216156 * Unemp$$

(A.0.6)

A.2.2 Greedy Best First Seach

Full Dataset

$$\boldsymbol{CPI} = -36.410 * Gdp + -15.805 * sin(-38354.995 * Unemp + 8200.181) * sin(12784.398 * Unemp + -2724.001) + 31.548 * sin(12785.382 * Unemp + -2667.829) * sin(12787.094 * Unemp + -2798.109) + 36.425 * Gdp$$

(A.0.7)

$$\boldsymbol{GDP} = 70.253 * Cpi + 1461.929 * sin(13356.763 * Unemp + 601.103) * sin(-13221.191 * Unemp + -1044.731) + 38.984 * Unemp * sin(0.692 * Cpi + 49.354)$$

(A.0.8)

$$\boldsymbol{Unemp} = 5.602 + 0.0302 * Cpi + -0.00338 * Gdp + 0.406 * sin(-10898.806 * Cpi + -1053996.99) * sin(10900.798 * Cpi + 1054000.3)$$

(A.0.9)

Reduced Dataset

$$\boldsymbol{CPI} = -267.682 * Gdp + 267.694 * Gdp + 2.621 * Unemp + 1.197 * sin(-409558.045 * Unemp + -44180.111) * sin(409554.487 * Unemp + 44220.366)$$

(A.0.10)

$$\boldsymbol{GDP} = -201.944 * Unemp + 79.074 * Cpi + 5.954 * Unemp * sin(145.036 * Unemp + -938.690)$$

(A.0.11)

$$\boldsymbol{Unemp} = -0.00284 * Gdp + 0.241 * Cpi + -0.000782 * Cpi * sin(1.07 * Cpi + -7.82)$$

(A.0.12)

A.2.3 Probabilistic Context-Free Grammar

Full Dataset

$$\boldsymbol{CPI} = 0.0157 * Gdp + -32.4056 + 1.491 * Unemp + -3.137 * sin(5.031 * Unemp + -22.656)$$

(A.0.13)

$$\boldsymbol{GDP} = 1525.110 * sin(-2097.402 * Unemp + 1757.378) * sin(2237.979 * Unemp + -2219.793) + 69.575 * Cpi$$

(A.0.14)

$$\boldsymbol{Unemp} = -0.000288 * Gdp + 274842.95 + -549680.458 + 274842.95 + 0.0278 * Cpi$$

(A.0.15)

Reduced Dataset

$$\mathbf{CPI} = -128878.513 * Unemp + 128881.199 * Unemp + 0.0125 * Gdp$$

(A.0.16)

$$\mathbf{GDP} = 41.63498904 * sin(714.3652412 * Unemp + -4322.179) + -201.169 * Unemp + -4828.383 * Cpi + -8356.944 * Cpi + 13264.351 * Cpi$$

(A.0.17)

$$\mathbf{Unemp} = -0.00284 * Gdp + -0.12 * sin(1.069 * Cpi + -7.654) + 0.241 * Cpi$$

(A.0.18)

A.3 Baseline Equation Discovery Forecasts vs Actuals

Full Dataset

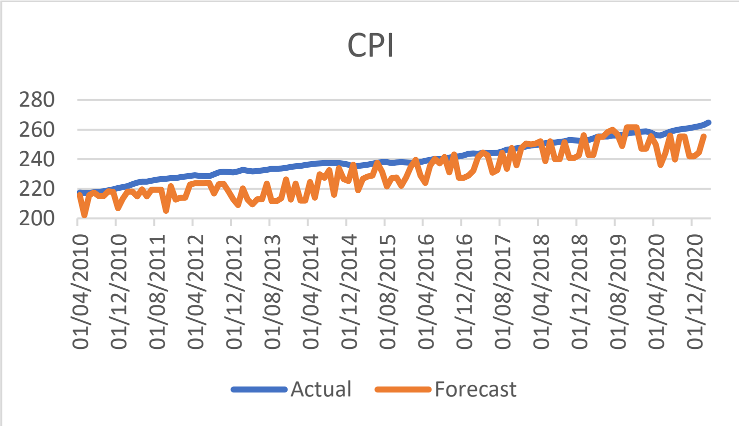

Figure 24 - Chart of Baseline Equation Discovery CPI Forecast (Full Dataset) Against Actual Data

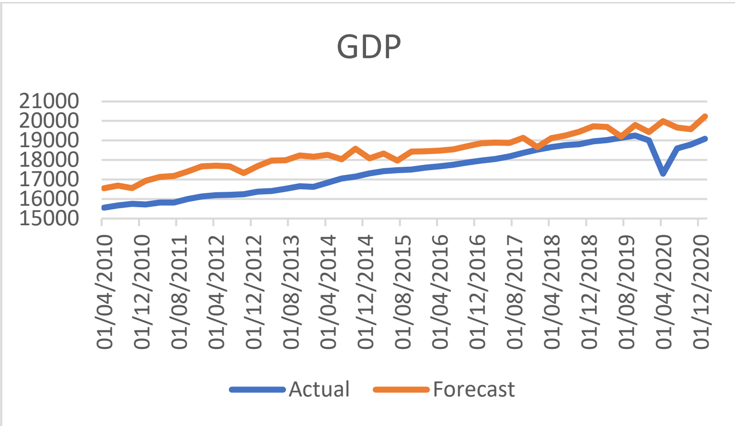

Figure 25 - Chart of Baseline Equation Discovery GDP Forecast (Full Dataset) Against Actual Data

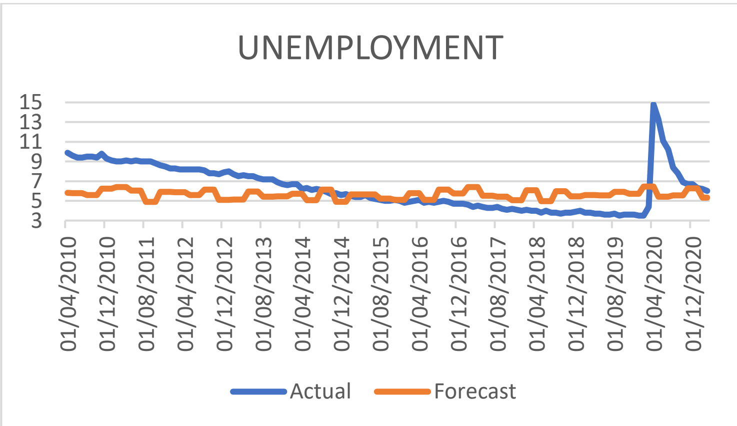

Figure 26 - Chart of Baseline Equation Discovery Unemployment Rate Forecast (Full Dataset) Against Actual Data

Reduced Dataset

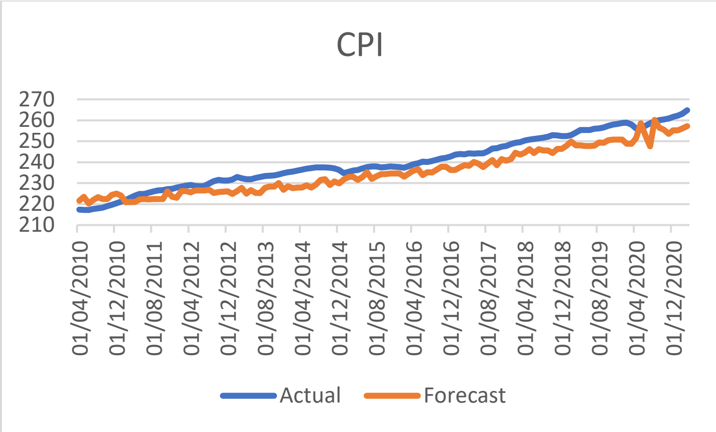

Figure A.20 - Chart of Baseline Equation Discovery CPI Forecast (Reduced Dataset) Against Actual Data

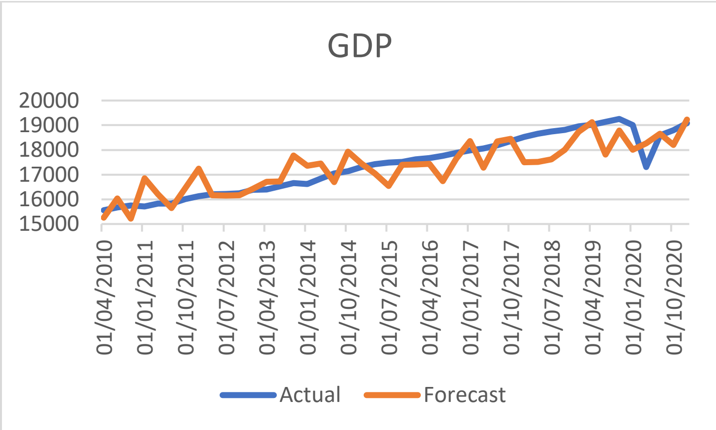

Figure 27 - Chart of Baseline Equation Discovery GDP Forecast (Reduced Dataset) Against Actual Data

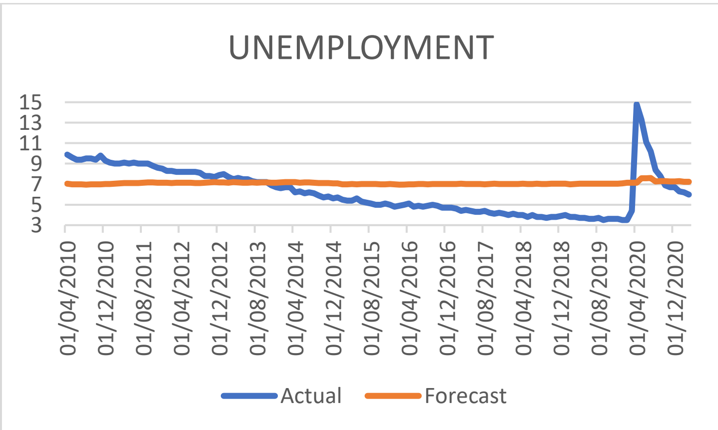

Figure 28 - Chart of Baseline Equation Discovery Unemployment Rate Forecast (Reduced Dataset) Against Actual Data

A.4 GBFS Equation Discovery Forecasts vs Actuals

Full Dataset

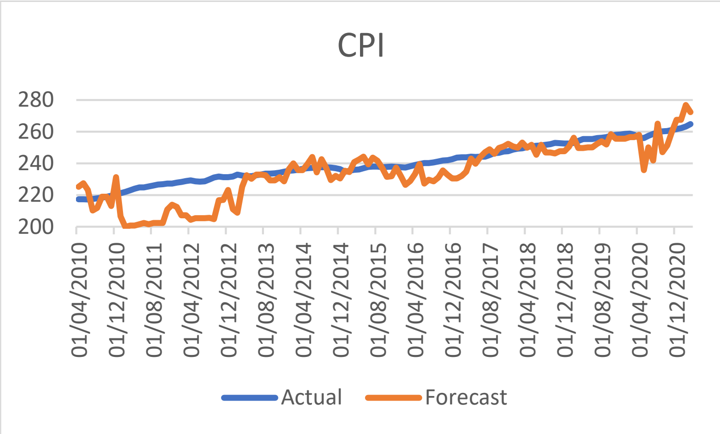

Figure 29 - Chart of GBFS Equation Discovery CPI Forecast (Full Dataset) Against Actual Data

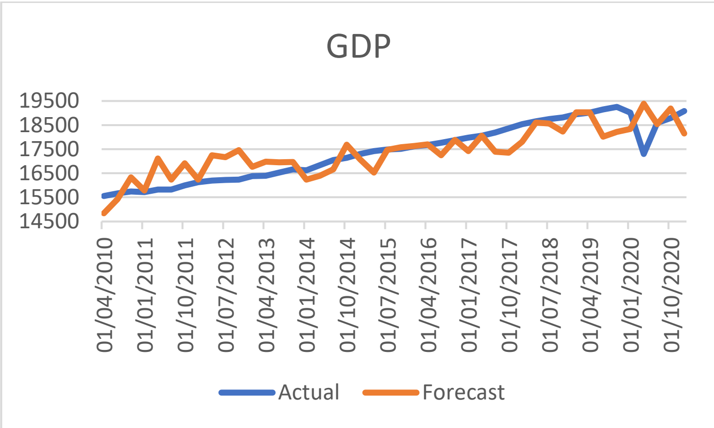

Figure 30 - Chart of GBFS Equation Discovery GDP Forecast (Full Dataset) Against Actual Data

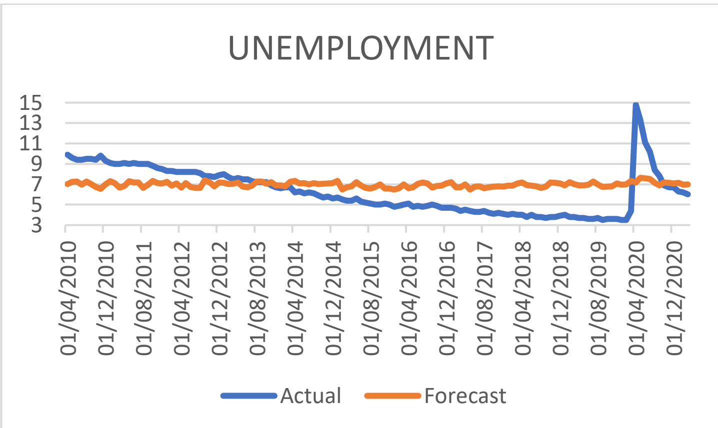

Figure 31 - Chart of GBFS Equation Discovery Unemployment Rate Forecast (Full Dataset) Against Actual Data

Reduced Dataset

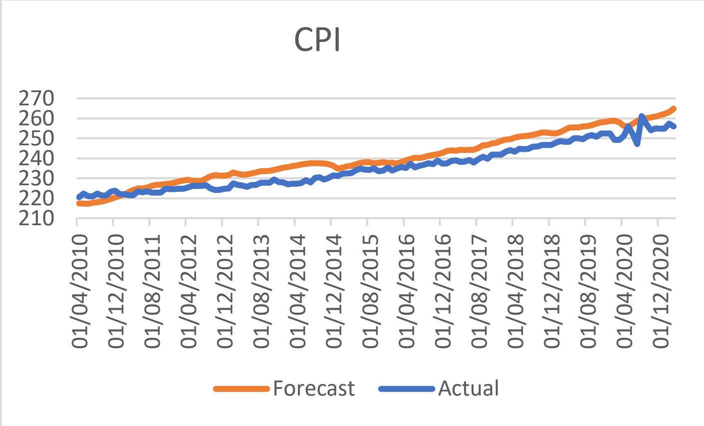

Figure 32 - Chart of GBFS Equation Discovery CPI Forecast (Reduced Dataset) Against Actual Data

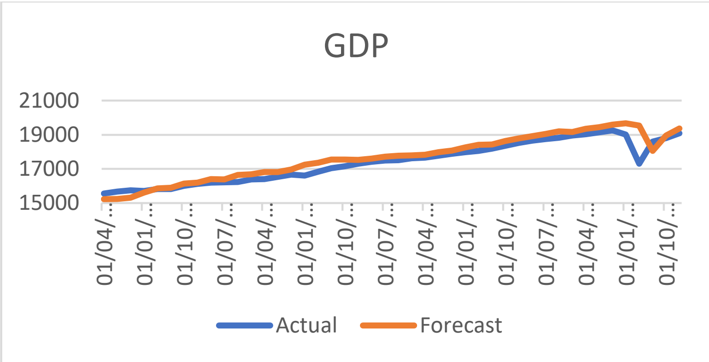

Figure 33 - Chart of GBFS Equation Discovery GDP Forecast (Reduced Dataset) Against Actual Data

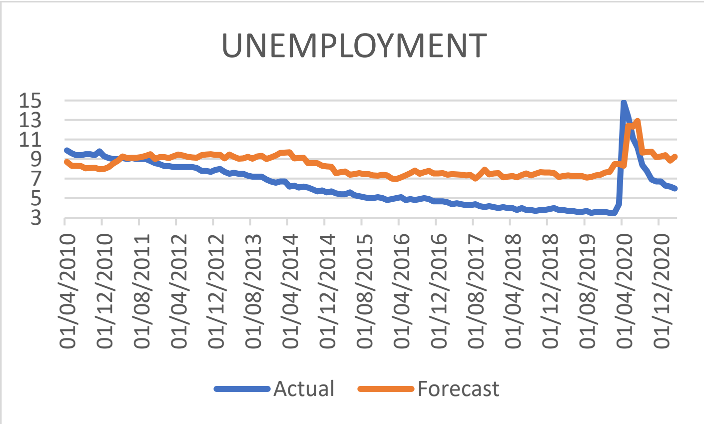

Figure 34 - Chart of GBFS Equation Discovery Unemployment Rate Forecast (Reduced Dataset) Against Actual Data

A.5 PCFG Equation Discovery Forecasts vs Actuals

Full Dataset

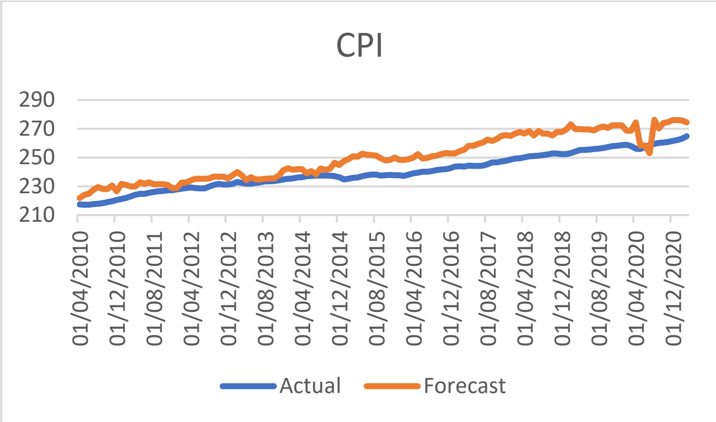

Figure 35 - Chart of PCFG Equation Discovery CPI Forecast (Full Dataset) Against Actual Data

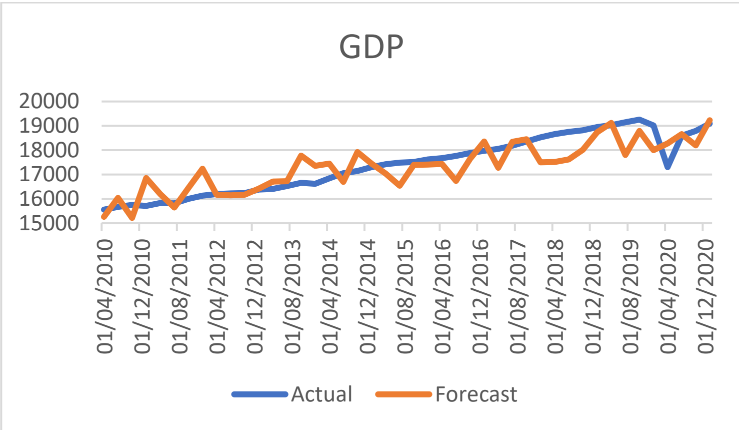

Figure 36 - Chart of PCFG Equation Discovery GDP Forecast (Full Dataset) Against Actual Data

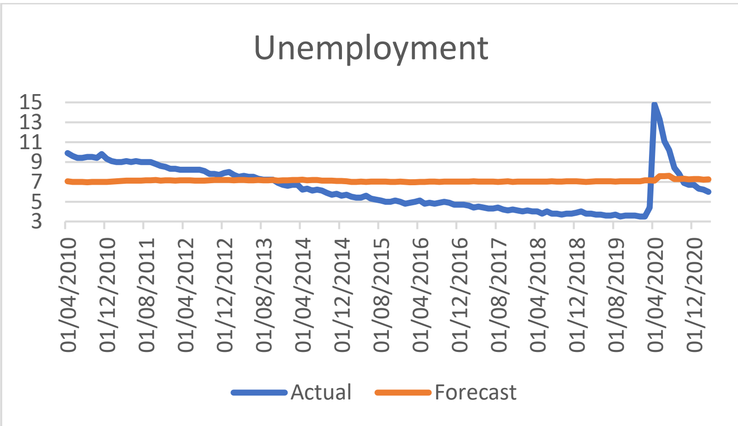

Figure 37 - Chart of PCFG Equation Discovery Unemployment Rate Forecast (Full Dataset) Against Actual Data

Reduced Dataset

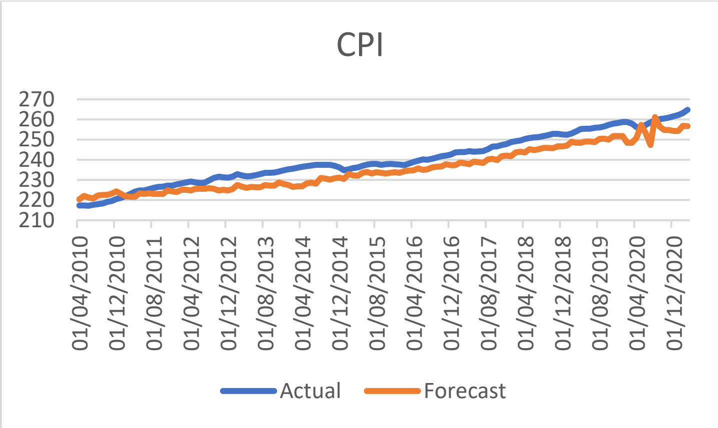

Figure 38 - Chart of PCFG Equation Discovery CPI Forecast (Reduced Dataset) Against Actual Data

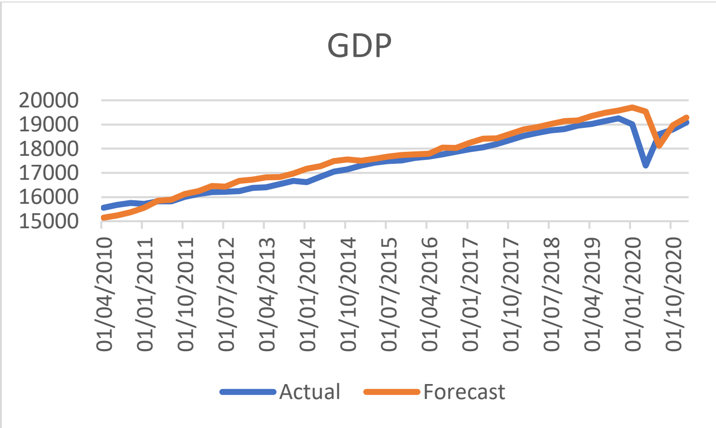

Figure 39 - Chart of PCFG Equation Discovery GDP Forecast (Reduced Dataset) Against Actual Data

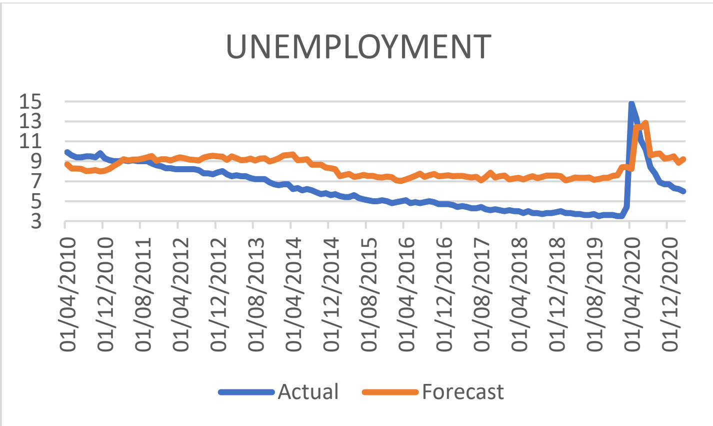

Figure 40 - Chart of PCFG Equation Discovery Unemployment Rate Forecast (Reduced Dataset) Against Actual Data

References

[1] "Equation discovery systems - Lagrange and Lagramge", *Kt.ijs.si*, 1997. [Online]. Available: http://kt.ijs.si/ljupco_todorovski/ed/lagrange.html. [Accessed: 13- Mar- 2021].

[2] J. Terry – "Constructing an Equation Discovery Tool for Time Series Forecasting", Undergraduate computer science project, 14th May 2020.

[3] J. Keynes, "*The general theory of employment, interest, and money"*, 1936.

[4] L. Todorovski, S. Dzeroski, "Declarative bias in equation discovery". In *Proceedings of Fourteenth International Conference on Machine Learning*, 1997, pages 376-384, San Mateo, CA.

[5] L. Todorovski, S. Džeroski, P. Langley, C. Potter, "Using equation discovery to revise an Earth ecosystem model of the carbon net production, Ecological Modelling", Volume 170, Issues 2–3, 2003, Pages 141-154, ISSN 0304-3800, DOI: 10.1016/S0304-3800(03)00222-9.

[6] "Welcome to Python.org", *Python.org*. [Online]. Available: https://www.python.org/ . [Accessed: 08- Jan- 2021].

[7] "Macroeconomics - Wikipedia", *En.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/Macroeconomics. [Accessed: 11- Dec- 2020].

[8] S. K. McNees, "Forecasting Accuracy of Alternative Techniques: A Comparison of U.S. Macroeconomic Forecasts", Journal of Business & Economic Statistics, 1986, vol.4, Pages 5–15, DOI: 10.2307/1391379.

[9] G. Gawera, "Neural Networks for Macro-Economic Forecasts", Undergraduate computer science project, University of York, May 2020.

[10] R. Fildes, H. Stekler, "The state of macroeconomic forecasting", Journal of Macroeconomics, Volume 24, Issue 4, 2002, Pages 435-468, ISSN 0164-0704. DOI: 10.1016/S0164-0704(02)00055-1.

[11] F. Panetta, “The Stability of the Relation Between the Stock Market and Macroeconomic Forces”, Economic Notes, vol. 31, Pages 417–450, 2002, DOI: 10.1111/1468-0300.00093

[12] A. Humpe & P. Macmillan “Can macroeconomic variables explain long-term stock market movements? A comparison of the US and Japan”, Applied Financial Economics, 2009, Pages 111-119, DOI: 10.1080/09603100701748956

[13] Bank of England Knowledge Bank – “What is GDP?” [Online]. Available: https://www.bankofengland.co.uk/knowledgebank/what-is-gdp [Accessed: 11- Feb- 2021].

[14] P. Bondarenko, “Gross domestic product,” Britannica, 28 February 2017. [Online]. Available: https://www.britannica.com/topic/gross-domestic-product. [Accessed: 11- Feb- 2021].

[15] Office of National Statistics – “A guide to labour market statistics”, 16 June 2020. [Online]. Available: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employment. [Accessed: 11- Feb- 2021].

[16] "Consumer Price Index (CPI) Definition", *Investopedia*. [Online]. Available: https://www.investopedia.com/terms/c/consumerpriceindex.asp. [Accessed: 13- Feb- 2021].

[17] R. H. Webb, W. Lupoletti, “Defining and Improving the Accuracy of Macroeconomic Forecasts: Contributions from a VAR Model”, October 1984, FRB Richmond Working Paper No. 84-6, DOI: 10.1086/296328.

[18] Riaz Khan, “ARIMA model for forecasting – Example in R”, December 31, 2017, [Online], Available: https://rpubs.com/riazakhan94/arima_with_example. [Accessed: 01- Mar- 2021].

[19] C. R. Nelson, “A Benchmark for the Accuracy of Econometric Forecasts of GNP”, Business Economics Volume 19, April 1984, Pages 52-58.

[20] M. Masum, "Identifying AR and MA terms using ACF and PACF Plots in Time Series Forecasting", *Towards data science,* 2020. [Online]. Available: https://towardsdatascience.com/identifying-ar-and-ma-terms-using-acf-and-pacf-plots-in-time-series-forecasting-ccb9fd073db8. [Accessed: 03- Mar- 2021].

[21] J. bhattacharjee, “*Common metrics for Time Series Analysis”.* [online] *Medium*, 2019, Available at: https://joydeep31415.medium.com/common-metrics-for-time-series-analysis-f3ca4b29fe42 [Accessed 2 June 2021].

[22] P. Langley, S. Herbert, G. L. Bradshaw, J. M. Zythow, “Scientific Discovery: Computational Explorations of the Creative Process”, 1987, MIT Press, Cambridge, MA, DOI: 10.1177/027046768800800417.

[23] D. Kazakov and T. Tsenova, “Equation Discovery for Macroeconomic Modelling”, English, in ICAART 2009: proceedings of the international conference on agents and artificial intelligence, A. Fred, ed., insticc-inst syst technologies information control & communication, 2009, Pages 318–323, DOI: 10.5220/0001802403180323

[24] Z. Georgiev and D. L. Kazakov, “Learning Ordinary Differential Equations for Macroeconomic Modelling”, English, in IEEE SSCI 2015, IEEE, 2015, Pages 905–909, DOI: 10.1109/SSCI.2015.133.

[25] J. Brence, L. Todorovski, S. Džeroski, “Probabilistic grammars for equation discovery”, Knowledge-Based Systems, Volume 224, 2021, DOI: 10.1016/j.knosys.2021.107077

[26] S. Edelkamp, S. Schrödl, “*Heuristic search: theory and applications”*. Elsevier, 2011.

[27] S. Russell, P. Norvig, “*Artificial intelligence: A Modern Approach”*, 3rd ed. 2010.

[28] R. Dechter, J. Pearl, “Generalized best-first search strategies and the optimality of A*”, Journal of the ACM (JACM) 32, Pages 505-536, DOI: 10.1145/3828.3830

[29] "A* search algorithm - Wikipedia", *En.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/A*_search_algorithm. [Accessed: 12- Apr- 2021].

[30] "Macroeconomic and sectoral statistics - ECB Statistical Data Warehouse", *Sdw.ecb.europa.eu*. [Online]. Available: https://sdw.ecb.europa.eu/browse.do?node=9691101. [Accessed: 09- Dec- 2020].

[31] The Bank of England, “A millennium of macroeconomic data”, [Online]. Available: https://www.bankofengland.co.uk/statistics/research-datasets [Accessed: 03- Dec- 2020].

[32] "Federal Reserve Economic Data | FRED | St. Louis Fed", *Fred.stlouisfed.org*. [Online]. Available: https://fred.stlouisfed.org/. [Accessed: 03- Dec- 2020].

[33] R. Nau, “Notes on non-seasonal ARIMA models”, [Online] Available: http://people.duke.edu/~rnau/Notes_on_nonseasonal_ARIMA_models--Robert_Nau.pdf [Accessed: 03- April- 2021].

[34] "Unit Root Tests", University of Washington, Available: https://faculty.washington.edu/ezivot/econ584/notes/unitroot.pdf

[35] R. Davidson, J. G. MacKinnon, “Econometric theory and methods”, Volume 5. New York: Oxford University Press, 2004.

[36] J. Brownlee, "How to Check if Time Series Data is Stationary with Python", *Machine Learning Mastery*, 2016. [Online]. Available: https://machinelearningmastery.com/time-series-data-stationary-python/. [Accessed: 15- Feb- 2021].

[37] "SciPy.org — SciPy.org", *Scipy.org*. [Online]. Available: https://www.scipy.org/. [Accessed: 10- Apr- 2021].

[38] T. Hastie, R. Tibshirani, J. Friedman, “The elements of statistical learning: data mining, inference, and prediction”, Springer Science & Business Media, 2009.

[39] "List of recessions in the United States - Wikipedia", *En.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_recessions_in_the_United_States. [Accessed: 13- Mar- 2021].

[40] F. Rossi, “*Handbook of constraint programming”*. Amsterdam: Elsevier, 2006.

[41] R. Lloyd-Jones, “The First Kondratieff: The Long Wave and the British Industrial Revolution.” *The Journal of Interdisciplinary History*, vol. 20, no.4, 1990, Pages 581–605. *JSTOR*, www.jstor.org/stable/204000. Accessed 12 June 2021.