Norwegian School of Economics
Bergen, Autumn 2023

Empirical Asset Pricing via Neural Networks and Macroeconomic Data

Juan Berasategui Gallego

Supervisor: Walter Pohl

Master thesis, Economics and Business Administration

Major: Economic Analysis

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis marks the end of my master's at the Norwegian School of Economics. I am truly grateful to both the school and Norway for the opportunity to partake in this outstanding academic program. During these past years, I have had the privilege of meeting and sharing the classroom with numerous brilliant individuals, many of whom I am fortunate enough to count as friends today. Furthermore, I would like to express my heartfelt thanks to these friends, my family, and Frida for their continuous support, understanding, and counsel. Additionally, I would like to express my gratitude to my supervisor, Walter Pohl, for his guidance, invaluable feedback, and suggestions, without which this thesis would not have been possible.

Norwegian School of Economics

Bergen, December 2023

Abstract

This study showcases the benefits of expanding the dimensions of the variable input vector with macroeconomic predictors when predicting monthly out-of-sample stock-level risk premiums. Using 610 predictor variables, we achieve a prediction performance of 3.19% R^2_{oos} for our best model and for stocks with a large market value of equity, over a fourfold increase in performance compared to previous research on this area. Furthermore, by using Shapley values, we show the pricing importance of each group of variables, challenging the view of neural networks as *black boxes*. The resulting Shapley values indicate that the neural networks weight higher variables such as labor market, interest and exchange rates, and prices, during recession, and stock characteristics during expansion periods.

Contents

List of Figures

List of Tables

1 Introduction

The contributions of this paper are twofold. Firstly, we are looking to develop a neural network model capable of achieving state-of-the-art performance for the fundamental problem in empirical asset pricing research, measuring asset risk premiums. We do so by expanding the number of predictor variables up to a total of 610. Then, we look into these neural networks, often described as *"black boxes"*, and observe which variables are relevant to the aforementioned problem. Lastly, we make a distinction among expansion, and recession economic periods, to understand the dynamics of pricing during different phases of the economic cycle.

The first part of our study is based on the *Empirical Asset Pricing via Machine Learning* paper by Gu et al., 2020. Regression-based methods have been a standard to finance academia, but the aforementioned research paper shows that the use of non-linear predicting methods, such as neural networks, can double the leading prediction performance on regression strategies when forecasting risk premiums. The main advantage of neural networks is their ability to use a large amount of predictors, and being able to find patterns and connections between them, almost independently, through optimization algorithms.

Our theory, on which we base this first part of the paper, is that the capital markets act as an aggregator of information. Collectively, news, rumors, speeches, and potentially other millions of data points are considered and weighted to produce risk premiums for each individual asset. Taking this into account, the idea of predicting returns with a handful of factors seems relatively naive. However, until recently, the popularity and research on machine learning models was rather scarce, and their applications to finance and how to infer causality from its results was an obscure and alchemy-like process, leaving regression-based strategies with a few predictor variables as the only choice for academics.

An explosion of research in deep learning has given place to the use of these techniques in the finance field. Gu et al., 2020, use close to a hundred stock characteristics exclusively as predictor variables to achieve their results using neural networks. Based on our theory, we believe that with an even larger number of relevant predictors, one may achieve better prediction performance. Therefore, we use six times as many variables as Gu et Al. Furthermore, we believe that in order to paint a more holistic picture of the risk premiums

as an aggregate of information, we need to take into account and include macroeconomic variables, since they affect the return distribution of each asset individually. In addition, we believe that the aforementioned is especially true in periods of economic recession.

Our results for the first part of the research seem to substantially support our theory, by producing a monthly out-of-sample stock level prediction of 3.19% R^2 for stocks with a large market value of equity and 1.69% for stocks with a small market value. Compared to 0.70% and 0.47% R^2, respectively, achieved in Gu et al., 2020. The 4-times-as-high prediction performance may be an indicator that the inclusion of more variables can be relevant to a higher forecasting score.

As economists, we are especially interested in causality. In some areas of machine learning, techniques such as principal component analysis, have been developed to understand how the models work by pinpointing which variables are important to the model's output. Many such methods, are often much more complex, if not completely unavailable, to use with deep learning. The challenges in inferring causality from neural networks have been a catalyst of creativity, opening the door to methods from other disciplines to be used in order to understand the intricacies of deep learning predictions.

One of these methods consists of using Shapley values, a technique loaned from game theory research, in order to calculate how each variable influences the model outputs. Then, when scaling these values, one may easily get the importance of a specific variable, as a percentage of the total output. We demonstrate that these values give coherent and consistent results across models. For each model, and for each economic period we showcase the top 20 variables with the most importance to the model's output, and then group all 610 variables into 9 groups: outcome and income, labor market, consumption and orders, orders and inventories, money and credit, interest and exchange rates, prices, stock market, and stock characteristics; we show that the importance of each one of these groups varies depending on the economic cycle. For expansions, the labor market is the group with the largest importance, while in recessions interest and exchange rates stand at the top. In addition, a striking result consists of the importance of individual stock characteristics towards forecasting performance, which results in remarkably lower in recession periods. This challenges the traditional focus on factor model pricing.

During the Hypothesis Development chapter, we discuss the background both in economic

and in technical terms, touching on topics relevant to this research ranging from the complex macroeconomic situation where we stand now, to the recent development of neural networks. Moreover, we discuss and formally define our hypotheses for this research. Then, in the Data chapter, we depict the retrieving, processing, splitting, and standardization of the data. In the Methodology chapter we describe in depth each of the techniques used in this paper, with the goal of making them understandable for any outsider without a background in asset pricing or machine learning. We continue by showcasing our results in the Analysis chapter and describing them. Finally, we ponder over the results and their significance, both for academia and market participants.

2 Hypothesis Development

In this second chapter, we explore the economic and technological background relevant to this paper. Moreover, we analyze pertinent literature and conclude by developing our hypotheses.

2.1 Background

2.1.1 Economic Background

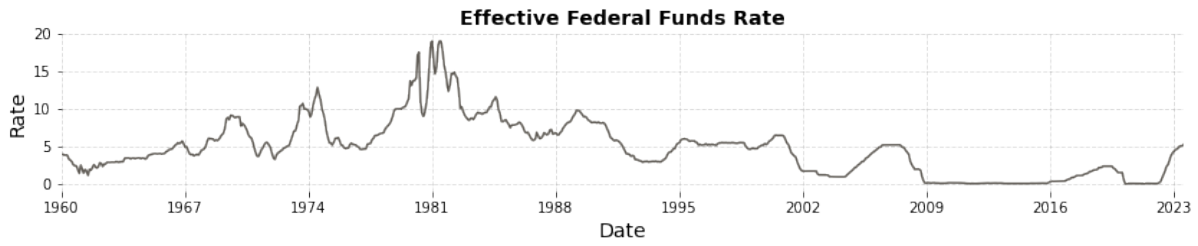
Following an extraordinarily long period with a very low federal funds rate, it appears that this phase has come to an end. Investors and institutions have faith in what has been coined as a *soft landing*. The term, natural to the aviation field, which refers to a pilot being able to land an aircraft in a controlled manner and without damaging it or the payload, is an analogy to the chair of the Federal Reserve, Jerome Powell, being able to cool down the economy without hurting it in the process. The pilot figure is clear, however, what is slightly less obvious is what exactly is the Federal Reserve landing, or what do we mean by *the economy*.

Despite it being a broad term, the Federal Reserve System (Fed) eyes the economy through five different lenses: gross domestic product (GDP), unemployment, inflation, lending, and asset prices (Blinder, 2023). All of these sides of the economy are interlinked in some way, and depending on the occasion or goal, the Fed will monitor them more or less closely. GDP refers to the measure of all final goods and services produced in a country at a specific time, and usually, a growing GDP will be a good sign, that is, unless inflation is a concern (Callen, 2008). Inflation is often measured by the consumer price index (CPI), which tracks the percentage change in a basket of goods and services consumed by households. Following the standard economic knowledge behind supply and demand, it is then straightforward how an increase in consumption, as pictured by a growing GDP, may be detrimental when prices are high (ongoing inflation). The primary instrument at the disposal of the Federal Reserve Chair in combating the potentially hazardous economic occurrence of inflation is the effective federal funds rate (EFFR). This rate denotes the interest at which depository institutions, such as commercial banks and credit unions,

lend reserve balances to other depository institutions overnight on an unsecured basis. As aforementioned, the EFFR has been outstandingly low up until the reopening after the pandemic. One of the effects of this monetary policy is its positive correlation with low returns on safe financial assets, while at the same time favoring borrowers boosting the prices of riskier asset classes ("Effective Federal Funds Rate", 2023). Considering this, we can see another connection between lending and asset prices, two of the pillars that the Federal Reserve is in charge of looking after.

We have seen that low effective rates serve as polishing for the gears of the economic machine, reducing friction by boosting lending, asset prices, and consumption. However, not all that glitters is gold and most countries are now experiencing an overheated machine making way for inflation in all parts of the economy. The Federal Reserve is planning to make a soft landing and plans on doing so by raising the EFFR. High-interest rates have the complete opposite effect on lending, making savings and safe assets more attractive, while lowering consumption (Cœuré, 2013). Looking at Figure 2.1, it can be seen that the EFFR is the highest it has been in the past 20 years. Faced with the above climate, investors and institutions are left with an asset pricing puzzle.

Figure 2.1: Effective Federal Fund Rate (EFFR) across time

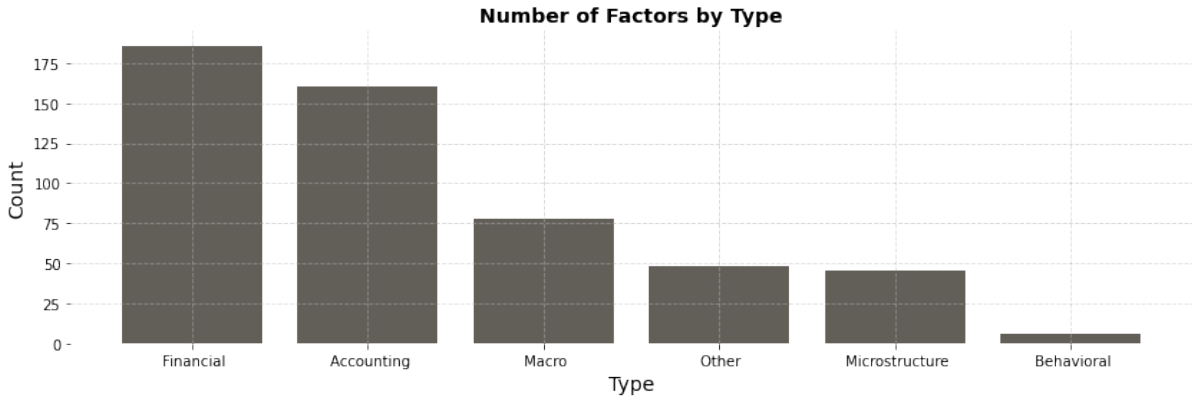


Asset pricing is a central topic in finance and consists of the pursuit of understanding and being capable of measuring the source of aggregate risk that drives asset prices, as described by Cochrane, 2009. Multiple models have emerged in the literature trying to explain how assets are priced in the financial market. The quintessential model is the Capital Asset Pricing Model (CAPM) by Sharpe, 1964 which calculates the expected return of an asset by adding the risk-free rate to the risk premium multiplied by the beta of the asset.

$$E[r_i] = r_f + \beta_i(r_m - r_f) \tag{2.1}$$

After this model countless factors models have emerged, shaping a phenomenon coined by academics as *the factor zoo*. This so-called zoo, counts as of 2020 with over 500 factors, each building a new risk factor to the equation based on empirical analysis or theory, with different types ranging from common financial factors to individual microstructure factors, all with the same goal of understanding the drivers of an asset's returns (Harvey & Liu, 2019). One of the issues with these factors is that the majority of them are based on individual firm characteristics or common financial factors, however, macroeconomic factors, which as seen in the previous paragraphs have become more and more important, represent only about 14% of the factors census. One explanation for this may be that academics have been overly focused on the idiosyncratic part of the risk premium, and not so much on the systematic risk compensation, which we could argue is more closely related to the macroeconomic environment.

Figure 2.2: Asset Pricing Factors by Type (Harvey & Liu, 2019)



The goal of the factor models is to predict equity risk premiums, and as a prediction task the most used techniques until lately consist of linear methods, which work fine with a small set of predictors, but begin encountering difficulties when one adds a larger set of predictors. In recent years, there has been a significant and rapid advancement in prediction technology, particularly in the realm of nonlinear predictive models. This progress has ushered in promising opportunities and possibilities for the asset pricing field

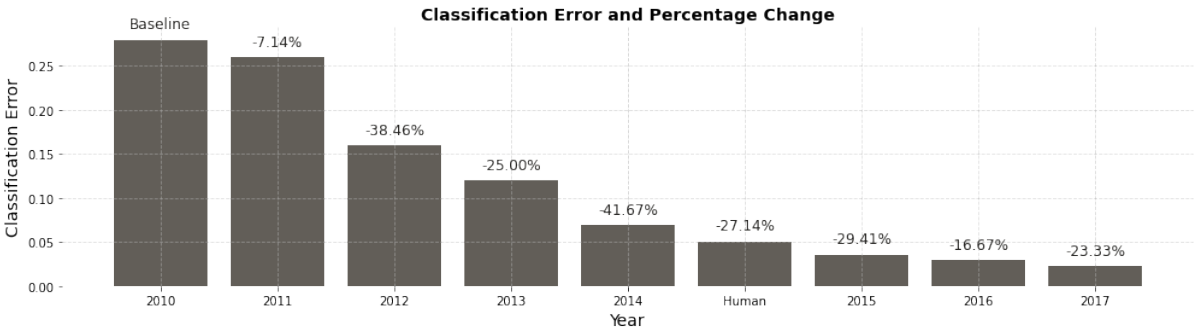
by allowing the use of as many factors as wanted in a single regression problem.

2.1.2 Technological Background

The advent of ChatGPT has been a transformative event for the machine learning field. Even if the technology back this generative chatbot dates from 2017, when the groundbreaking *Attention Is All You Need* (Vaswani et al., 2017) paper describing transformers architecture was published, it is after OpenAI's product was launched that the general public rushed into the field. The shock has been so large that governments across the world have begun planning for tight regulations for the field and have started hearings discussing oversight with industry leaders. Furthermore, some of these leaders have alerted society of the potential extinction risk from artificial intelligence through a public statement, "Pause Giant AI Experiments: An Open Letter", 2023, calling to halt the development of AI models, especially, open-source models. Whether the aforementioned is truly a legitimate concern, or rather an effort to exert selective control or authority over access to large artificial intelligence models remains unclear.

However, not all is about generative artificial intelligence, machine learning has been used extensively by all fields of academia for multiple years. A great example that appropriately showcases the development of the field is the ImageNet Contest. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was a contest for object detection and image classification, which each year brought together researchers to compete against each other with the goal of achieving the best predictor models (Russakovsky et al., 2015). As seen in Figure 2.3, the change in classification error for the top teams between 2011 and 2012 was quite drastic. The difference in performance was based on the introduction of artificial neural networks (ANN) to the competition. Soon after 2012, all participating teams were using convolutional neural networks (a specific kind of ANN that works particularly great for machine vision tasks) in the competition. After 2014, the algorithms achieved performance that surpassed the human classification benchmark, and in 2017, it was the last time that the contest was held since the ImageNet dataset became obsolete for the models.

Figure 2.3: ImageNet Contest Lowest Classification Error per Year (Fei-Fei & Deng, 2017)



In Chapter 4, we dive deeper into the technical explanation of what really is an artificial neural network. However, to summarize it briefly, an artificial neural network is a system that emulates that of biological neurons, both in its form and in the learning process. Instead of somas, dendrites, and nuclei, an ANN contains, weights, and nodes with activation functions (Ghedira & Bernier, 2004). As seen in the ImageNet example, deep learning models have outpaced most other prediction models, and have become the preferred choice for most forecasting tasks. These systems have shared characteristics with the biological neural networks that differentiate them from the linear machine learning models, such as massive parallelism, adaptability, fault tolerance, or distributed computation (Sharma et al., 2012). The field of deep learning is not without detractors, however, many deem these models as "black boxes", meaning that the reasoning behind its predictions is opaque and unclear, and its interpretability is lacking. Another common critique is the processing time and compute power correlation, in the sense that if one plans to work with ANNs, there are significant barriers to entry on training and inference time, if computer power is low, or exorbitant prices otherwise.

Looking back at the first critique, one field that has been especially skeptical of these methods is economics. The field focuses on causality, meaning that we, economists, try to answer the following question: does X *cause* Y? In order to answer such a question, economists often use linear regressions, where the output is easily interpretable and can be tested for significance. While, with neural networks, it is extremely complicated, or even sometimes impossible to understand which predictors are important for the generated output and to what extent. Given that all the calculations happen in a so-called *hidden*

layer, making causal inference results in a Herculean task.

2.1.3 On Empirical Asset Pricing Via Machine Learning

In part, this paper is based on the *Empirical Asset Pricing Via Machine Learning* article by Gu et al., 2020, where they look at forecasting excess returns by using multiple machine learning methods, both linear and nonlinear. The authors in this article show the promising performance of non-linear predicting methods on the empirical asset pricing task. Gu et al., 2020 research focuses on showcasing that neural networks outperform any other kind of time-series forecasting model by achieving the best monthly out-of-sample prediction performance. Furthermore, they use a rich group of predictors containing 94 characteristics per stock and 8 common macroeconomic variables. Table 2.1 shows that neural networks achieve a better-predicting performance for the firms in the top 1000 ordered by descending market value of equity (log).

Table 2.1: Results from Gu et al., 2020 (as percentage of out-of-sample R^2)

	NN1	NN2	NN3	NN4	NN5
Top 1000	0.49	0.62	0.70	0.67	0.64
Bottom 100	0.38	0.46	0.45	0.47	0.42

These results seem to support our idea that modeling a richer environment in terms of the count of variables, is beneficial to prediction performance.

2.2 Theoretical Development

The fundamental theory behind this research is that asset prices essentially constitute an aggregate representation of parameters, potentially adding to thousands or even millions. These parameters may manifest in multiple forms, ranging from financial ratios, such as earnings per share (EPS), to geopolitical news, social media posts, rumors, among many other factors. The task of modeling all these parameters, discovering their interactions, and determining their respective weights would essentially mean solving the entire financial market—a rather implausible feat. Nevertheless, we can test this theory by employing a

simplified approach in our model. Gu et al., 2020 show that by applying modern machine learning to the asset pricing field, through empirical risk premium forecasting, they are able to achieve outstanding results. Moreover, compared to traditional asset pricing research, they use a significantly higher amount of parameters.

We argue that the incorporation of a greater number of predictive parameters into neural networks will result in better-performing models for asset pricing. In line with our objective to test this hypothesis, we will largely adhere to the methodology outlined in Gu et al., 2020, with the main modification being the expansion of variables utilized to forecast the monthly excess return for individual stocks. This approach establishes a ceteris paribus condition, allowing us to observe the impact of our theory on the prediction performance. While maintaining the core methodology, some minor adjustments may be necessary, as Gu et al. designed and processed their data with a different focus, including the testing of linear forecasting methods. However, we will refrain from exploring such methods, as their findings indicated their limited efficacy when applied to large input vectors. The minor variations will be elaborated upon in the forthcoming data and methodology section.

The theoretical parameters can take multiple forms, and adding any significant amount of them, we believe, may increase the out-of-sample R^2. However, due to the impracticality of obtaining millions of parameters, the key question arises: which set of pertinent parameters should be incorporated to broaden the input? The economic background section of this chapter underscores a notable shift in the macroeconomic landscape, as the Federal Reserve, the architect of the economy, envisions a departure from a low-interest-rate paradigm toward a more intricate macroeconomic scenario. Therefore, we consider it timely and relevant to expand the input vector using macroeconomic parameters up to a total of 610 predictor variables, that is, close to six times the amount of inputs employed in Gu et al., 2020.

Hence, our initial hypothesis is formulated as follows:

H1: The inclusion of a larger count of macroeconomic variables in addition to the stock characteristics, produces a better prediction performance.

We assume that an expansion of the input vector dimensions is a sine qua non of better

prediction results. However, we are also interested in understanding to what extent does each one of these new variables affect the outputs from the model. In a way, one may correctly describe our goal as looking inside the *black boxes*. We may do so by calculating the marginal contribution of each feature across all possible scenarios. This approach helps in expanding our understanding of asset pricing, potentially revealing unexpected parameters that have not been thoroughly researched and challenging the significance of others thought to be more pivotal. As of the resulting importance of our predictors, one may be able to learn from them and when further expanding our theory make a more educated choice on which new variables to include.

One important consideration for future research is the potential variability in the prediction power of variables based on the timing of the prediction. Meaning that certain variables perceived as highly significant may experience a notable reduction in prediction power depending on whether the prediction is made during periods of recession or expansion, and vice versa.

The aforementioned leads us to formulate the following hypothesis:

H2: Some variables may have a larger prediction power than others, these may vary in periods of recession and expansion.

We hope that the outcomes of this hypothesis will not only contribute to academic understanding but also offer practical guidance. By discerning the differential impact of variables in different economic phases, investors and policymakers can make more informed decisions. This strategic insight may help in monitoring key economic parameters and formulating positive-sum strategies, fostering better outcomes in both financial markets and macroeconomic policy development.

3 Data

The following chapter depicts the data for which this research is based, and the methods used to obtain them. Furthermore, we give insights into the sampling procedure and train test split.

The data used in this paper and described below is open-sourced and freely available at this Hugging Face repository.

3.1 Financial Data

The financial data consists of stock predictive characteristics based on extensive stock returns literature, and excess returns. Since the hypothesis consists of benchmarking on the existing research on machine learning applied to empirical asset pricing, we use the exact same predictive characteristics used in Gu et al., 2020. A total of 98 characteristics are used, including industry, market capitalization, and cash position, among many others (a total overview of the predicting characteristics can be found in the Appendix A). The majority of the characteristics are made available to the public after some delay. To prevent forward-looking bias, the dataset is built under the assumption that monthly attributes are delayed by a maximum of one month, quarterly attributes by at least four months, and annual attributes by at least six months. Therefore, when predicting returns for month t_{+1}, we utilize the most recent monthly attributes available at the end of month t, the most recent quarterly data up to the end of month t_4, and the most recent annual data up to the end of month t_6. The returns are extracted from the CRSP database, and the 3-month Treasury-bill rate is used as a proxy for the risk-free rate.

3.2 Macroeconomic Data

The main pillar of our paper consists of understanding the effect of macroeconomic indicators on individual stock excess returns. In order to achieve the aforementioned, we retrieve the 128 monthly updated variables described in the FRED-MD article by McCracken and Ng, 2016 which can be grouped into the following groups: output and income, labor market, consumption and orders, orders and inventories, money and credit,

interest rate and exchange rates, prices, and stock market. For each variable in the macroeconomic data, is either shown in nominal terms, log transformation, change, first or second difference, or percentage change(an overview of all the indicators and the transformations can be found in Appendix A).

Furthermore, we build three new variables for each one of the 128 base predictors, one for the nominal change with a month lag, another for the nominal change with a quarter lag, or 4 months, and lastly, a variable describing the yearly change, or 12 months. We do this to add more information to the macroeconomic predictors since most of them grow consistently from 1960 until 2021, but we believe that the magnitude of the change is relevant to our goal of predicting excess returns.

$$\Delta_{i,1} = X_{i,t} - X_{i,t-1} \tag{3.1}$$

$$\Delta_{i,4} = X_{i,t} - X_{i,t-4} \tag{3.2}$$

$$\Delta_{i,12} = X_{i,t} - X_{i,t-12} \tag{3.3}$$

Moreover, adding the nominal change in each variable adds background to the model without increasing its architecture's complexity, since our feedforward neural network will be able to access information from the past, although limited, which otherwise would have only been possible by using bi-directional ANNs such as recurrent neural networks (RNN).

Consequently, the total macroeconomic set is composed of 512 macroeconomic variables. Below we showcase an example of a variable, in this case, the Real Personal Income (RPI), by its nominal value across all dates in our dataset, and the monthly, quarterly, and yearly change. Figure 3.2 shows that by including the 1,4, and 12-month changes to each variable we get an approximately Gaussian distribution positively skewed while maintaining outliers. Achieving this distribution improves the learning results and significantly fastens calculations (Sola & Sevilla, 1997). .

Figure 3.1: Example of a macroeconomic variable with its monthly, quarterly, and yearly change

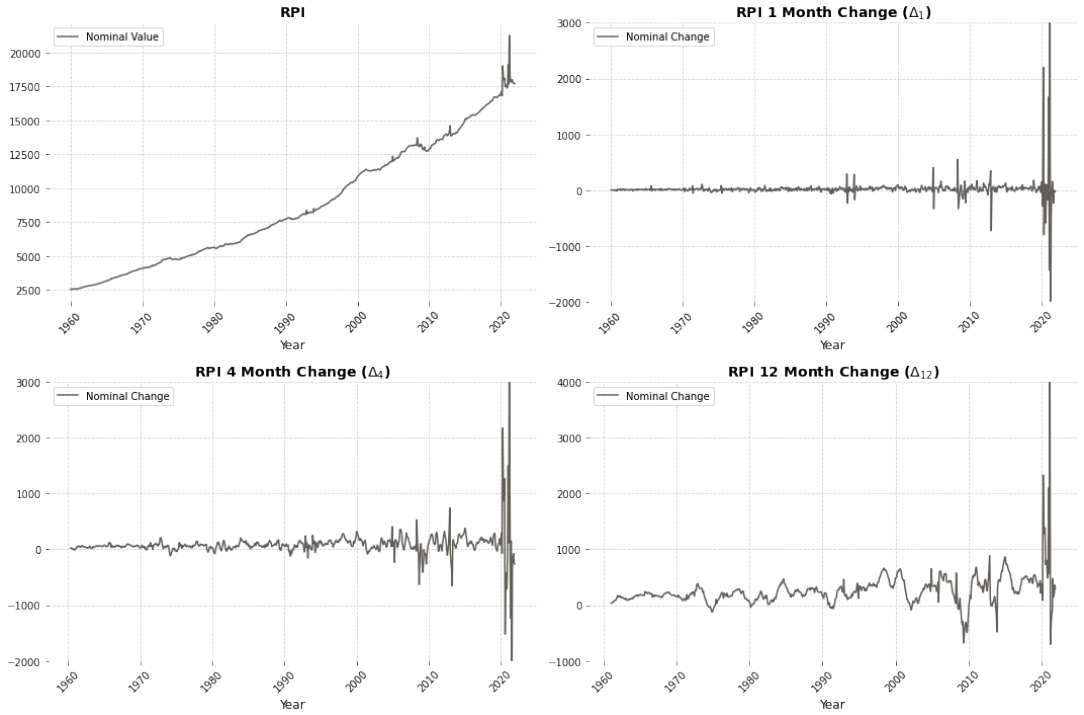
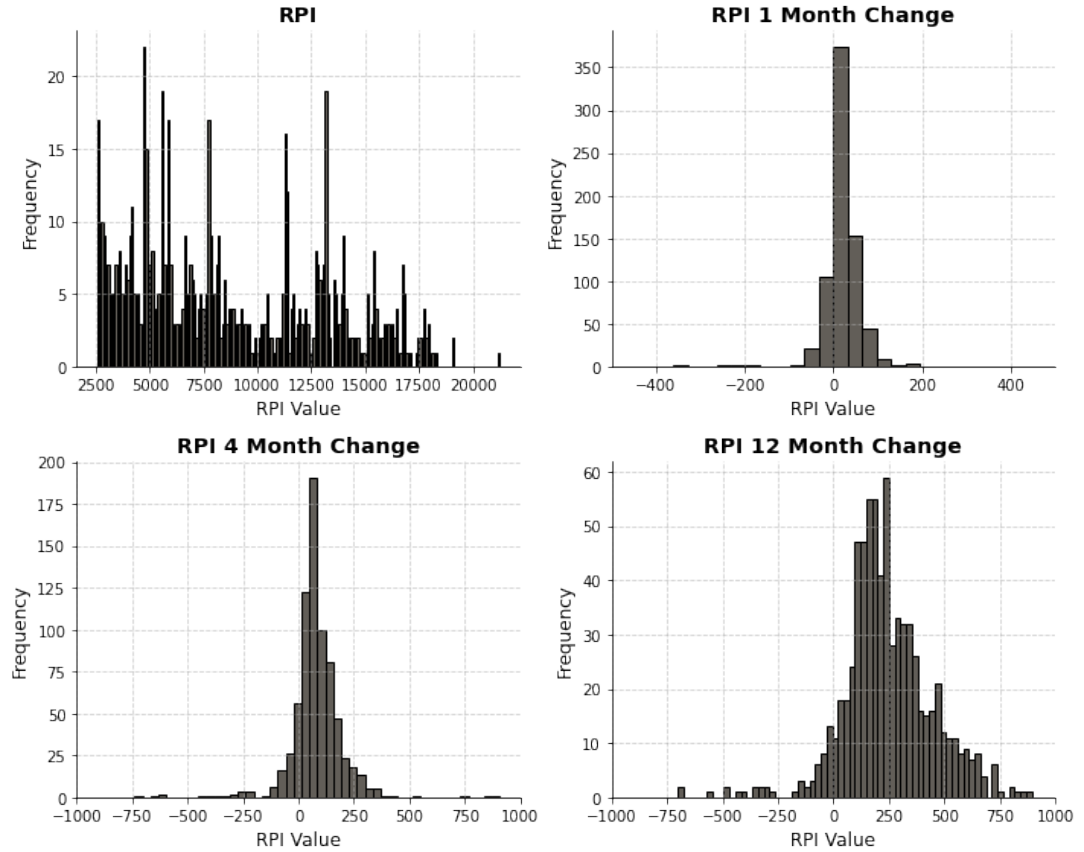


Figure 3.2: Distribution of a macroeconomic variable with its monthly, quarterly, and yearly change

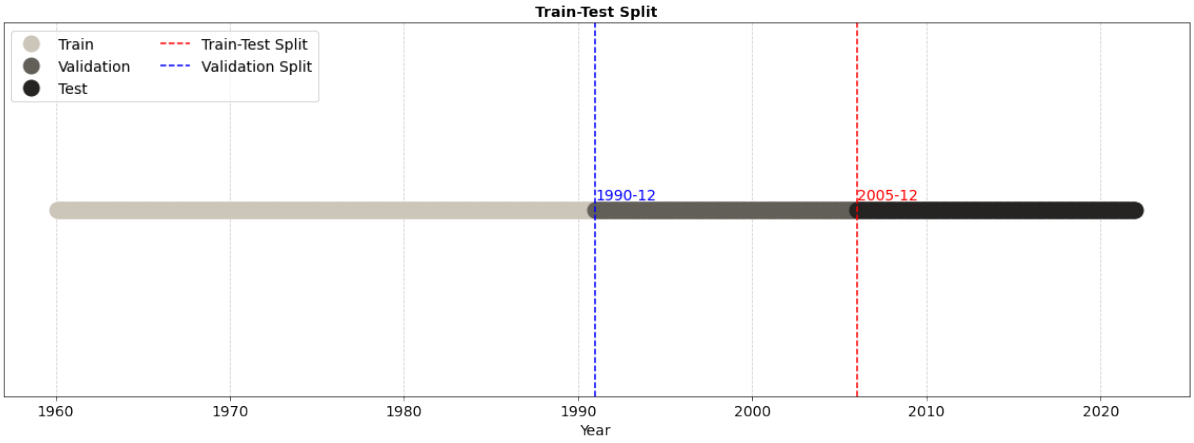


3.3 Data Split

Once the financial and macroeconomic data are merged together, the resulting dataset consists of 4,078,827 observations and 610 columns. Afterwards, we build two data sets, one for large market value stocks (Top 1000) and the other one for low market value stocks (Bottom 1000). The procedure to build these datasets consists of sorting the stocks by descending market capitalization on a monthly basis and taking the 1000 first stocks, for the first dataset, and the bottom 1000 for the next ones. Each of the subgroups consists of 720021 observations.

.

Figure 3.3: Chronological Data Split



In order to avoid look-ahead bias, we run a chronological train-test split for each dataset, which consists of making two datasets, one with which we will train the model, and adjust the model (the train set), and one with which we will "hide" from our model and will then be used for evaluating the predicting capabilities of our model (the set). We choose a split of 75/25 then we divide the train set into train and validation, so at the end, 50% of the data will be used in the train set, and 25% for both the validation and test set. This split is different from the one used by Gu et al., 2020, and was chosen deliberately in order to include the 2007-2008 great recession in the test sample while maintaining a meaningful size for all sets. Table 3.1 closely examines each one of the recessions appearing in our sample, its duration, and its belonging in the data split.

Table 3.1: Recessions included in our datasets. (Hall, 2003)

Peak Month (Start)	Trough Month (End)	Duration (Months)	Set
April 1960	February 1961	10	Train
December 1969	November 1970	11	Train
November 1973	March 1975	16	Train
January 1980	July 1980	6	Train
July 1981	November 1982	16	Train
July 1990	March 1991	8	Train
March 2001	November 2001	8	Validation
December 2007	June 2009	18	Test
February 2020	April 2020	2	Test

3.4 Standardization

Data scaling is crucial to machine learning problems, and makes it easier for the models to generalize on new data. It is even more important when the data at hand are time series from a large range of years with highly fluctuating values. Furthermore, neural networks tend to ignore small values if they come together with other values, therefore, making the process of scaling essential to our dataset. For this task, we choose standardization as our method for scaling the data which works by subtracting the mean of the variable column μ to each value x and dividing by the standard deviation of the column σ. To implement it, we use the function *StandardScaler()* from the *Scikit-learn* package. By standardizing the data, each variable will have a mean of 0 and a standard deviation of 1.

$$z = \frac{x - \mu}{\sigma} \tag{3.4}$$

This stage of data standardization represents one of the previously mentioned deviations from the methodology outlined in the *Empirical Asset Pricing via Machine Learning* paper. Gu et Al. describe their scaling method as ranking their variables period-by-period and mapping this rank to a [-1,1] interval. Our method of standardization, however, has the benefit of maintaining consistency of scale, which is particularly important when working with algorithms sensitive to the scale of variables, such as gradient descent Fei et al., 2021. Consequently, we believe that when working with neural networks only, as opposed to Gu et al., 2020, one may rather use z-score standardization. Moreover,

our chosen method of standardization is particularly well-fitted to work with variables measured in different units. Therefore, when working towards testing our theory of asset prices as aggregates of millions of different parameters, one may acknowledge the benefits of this method.

4 Methodology

In this chapter, we explore the theory back the methods used in this paper. First, we look at the economic methodology, in the sense of what is it that we are trying to answer, and what it means from an economic perspective. Then, we cover the methodology behind neural networks and their multiple parameters. Lastly, we look into how the theory is applied to our problem.

4.1 Economic Methodology

4.1.1 Risk Premiums

The goal of this research piece is to expand the asset pricing knowledge of the asset pricing field. This goal is inherently of an economic nature. Our plan is to achieve this by finding a way to predict risk premiums. In finance, we have the central notion that risk goes hand in hand with return, and investors must receive compensation when taking risks. Risk premiums can also be understood as conditional expected returns in excess of the risk-free rate. That is, in order to find the risk premium of an asset, we subtract the risk-free rate from its expected return. The risk-free rate represents an investment without risk, United States Treasury Bills are commonly used as a proxy for the risk-free rate, and depending on the prediction period (in our case monthly), one may use bills with longer or shorter bill maturities.

$$RP_i = E(R_i) - R_f \tag{4.1}$$

The risk premium is a conditional price on risk, it is conditional on millions, or potentially, an infinite amount of factors $\mathcal{F}$ (or information). Some of these factors include, but are not limited to, news, expectations, or social media posts. Each market participant may have its own expectations or understanding of these factors, and most may not even take into account a fraction of them (Damodaran, 1999). As previously described, the market works as an aggregator of expectations to form the risk premium.

$$RP = E(R_{i,t+1}|\mathcal{F}_t) \tag{4.2}$$

The field of economics tries to uncover causality, and therefore, asset pricing economic models, try to understand which economic mechanisms or equilibria, cause the $\mathcal{F}$ parameter in our equation. Consequently, when looking back at our hypothesis, we believe that the inclusion of more macroeconomic variables will make a better approximation of the market's information aggregate.

4.1.2 Recessions

Recessions or contractions, form part of the business cycles, they start at the peaks and end at the trough and can be described as a negative fluctuation in the aggregate economic activity of a nation. Usually, the measure of two consecutive quarters with negative GDP is used as to identify a recession period. However, since the time horizon for this thesis lies on a monthly basis, the definition of recession coined by the National Bureau of Economic Research (NBER) seems more relevant: *a recession involves a significant decline in economic activity that is spread across the economy and lasts more than a few months* (Hall, 2003).

As to what can be interpreted by economic activity, the NBER cites alll personal income less transfers, non-farm payroll employment, employment as measured by the household survey, real personal consumption expenditures, wholesale-retail sales adjusted for price changes, and industrial production, all of which are included in our macroeconomic predictor variables.

4.2 Technological Methodology

4.2.1 Neural Networks

As its name hints, a neural network (NN) is a mathematical representation of the structure and communications of biological neurons (Pohl, 2023). The popularity of neural networks has exploded in the last decade due to their outstanding performance in prediction problems. NNs are said to be non-linear, meaning that the network is capable of accurately

forecasting results that deviate from a linear or straightforward trajectory. Moreover, these methods have been able to prove the practicality of the universal approximation theorem, in part thanks to the recent advances in computing power (Higgins, 2021). This means that neural networks, as universal function approximates, are capable of predicting any non-linear existing function, including the risk premium function (4.2). Neural networks are composed of three different types of layers which are: the input layer, hidden layers, and output layer. Each one of these layers is formed by nodes or artificial neurons. A neural network can be classified as deep or shallow depending on the number of hidden layers in its architecture.

Every layer in a neural network is formed by nodes, and then every node is connected to all other nodes of the following layer. A node works by taking a vector of inputs $\mathbf{X}$, these inputs are received directly from the data (in the case of the nodes in the input layers) or are the output from all other connected nodes. Inside each node, we have a vector of weights $\mathbf{W}$, an activation function f, and a bias b (Pohl, 2023). Then, the node produces an output y, by producing a sum of the product of each input with its corresponding weight, and after that, the bias is added to the sum. Then, the final number is passed through the activation number. Equation (4.4) shows this process. Finally, the output is passed into the nodes of the following layer, and the process repeats, unless it is the final node on the output layer, which will then be the final prediction.

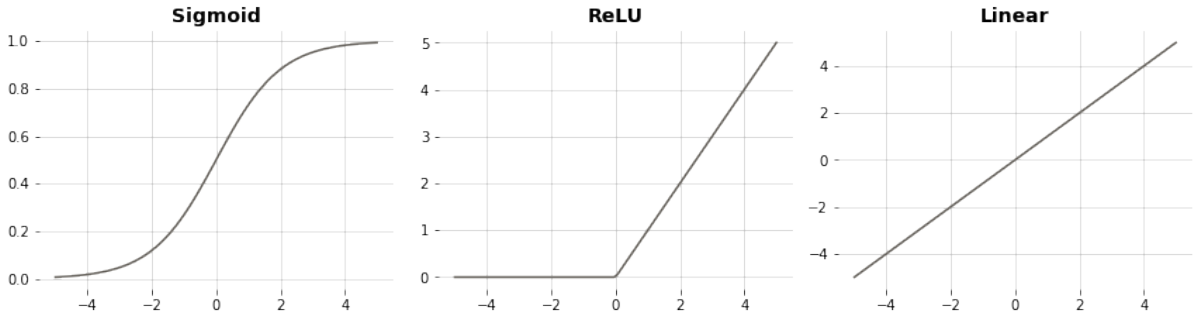
$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{4.3}$$

$$y = f\left(b + \mathbf{X}^T \cdot \mathbf{W}\right) \tag{4.4}$$

The weights are parameters determined during the training process of the neural network and represent the importance that the neural network assigns to each one of the inputs. Consequently, the weights control how much influence each node has on the following

ones. The bias parameter plays an important role in the node, it is a constant added to the product of weights and inputs before being passed through the activation function, and it represents the node's propensity to activate by shifting the activation function towards the left or right (negatively or positively, respectively). The activation function consists of a mathematical function that can take multiple forms depending on the layer and the type of prediction. whenever the node is not in the output layer, the activation will usually be non-saturated,(e.g. rectified linear unit (ReLU)) which gives the network the property of non-linearity and facilitates convergence (Xu et al., 2015). However, if it is the last node (output layer), the activation function will depend on the prediction, for example, if the target variable is a class, a sigmoid activation will be used, or in our case, predicting floats, a linear activation, or no activation will be used.

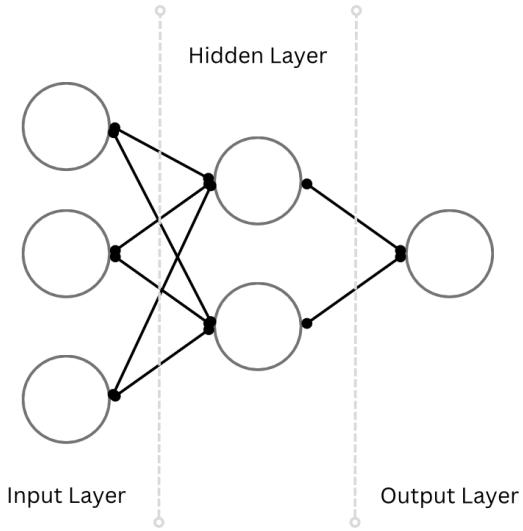
Figure 4.1: Examples of Activation Functions (f)



The number of nodes in the input layer is dictated by the shape of the predictive variable, equalling a node, or neuron, per variable, which will subsequently process the data and pass it through to the hidden layers. Another important characteristic of the input layer of an artificial neural network is connectivity, meaning that each node is fully connected to each of the subsequent nodes in the hidden layer, Figure 4.2 shows this. Regarding the activation function, which we will cover further in this chapter, it is relevant to specify that in most cases, input layers do not have activation functions. Then, depending on the number of hidden layers present in a model, there exist deep neural networks, for models with more than one hidden layer, and shallow neural networks otherwise. The deeper the network the better the capacity of extracting hierarchical features and relationships from the data, although by adding more layers, one also increases the risk of overfitting (Robles Herrera et al., 2022). Also, depending on the number of nodes per hidden layer,

we can classify hidden layers into narrow or broad. The more neurons on a layer, the higher the capacity of a network to learn details from the data, at the expense of a higher computational complexity. Furthermore, by adjusting the weights, the hidden layers serve as feature extractors and transformers, where most of the learning happens in the network. Lastly, the output layer is where the information converges and a prediction is given. The activation, here is dependent on what the prediction problem consists of.

Figure 4.2: Topology of a Neural Network



When considering all of the nodes in every layer of the neural network, a neural network can be represented as one function of a vector of inputs $\mathbf{X}$ with a single output y.

$$y = F(\mathbf{X}) \tag{4.5}$$

where

$$F = \{f_1, f_2, \ldots, f_n\} \tag{4.6}$$

Once the topology of the neural network and its width are specified, one follows up by selecting the data relevant to the goal. Ideally, one splits the data set into three different sets, train, validation, and test. The train set is the one on which the network learns, it is also, usually, the largest set. Then another set, the validation set, is selected and used for evaluating the trained models through each iteration. Lastly, the test set, should be held

out from the process, until the very end when the model will not be further trained. This test set is what is used to report model performance (Bai et al., 2021). Relevant to this topic, the term *overfitting* was mentioned and is relevant to this split. The term itself is used when a model is not able to generalize what it learns in the training set to data that it has not yet seen, or the test set (Ying, 2019). Effectively, when training a model, one would see that it overfits when the performance scores look great in the training set but not so much in the out-of-sample data.

Regarding performance scores, The concept that showcases how the model learns is called the *loss function*. The loss function L measures how bad a model is, or how far off is each prediction $F(\mathbf{X})$ from the real value y that is being predicted (Pohl, 2023). There are multiple loss metrics to measure the learning ability of a model, some tailored to the prediction problem and some more general. Some examples of loss functions and associated use cases are, mean square error (MSE) and regression problems, or binary-crossentropy for binary classification problems. The concept that showcases how the model learns is called the *loss function*.

For an individual output, we have:

$$Loss_y = L(y, F(\mathbf{X})) \tag{4.7}$$

for a vector of target values $\mathbf{Y}$

$$Loss_{\mathbf{Y}} = L(\mathbf{Y}, F(\mathbf{X})) \tag{4.8}$$

Since the loss is a measure of prediction error for a model, it creates an optimization problem to minimize it. The most commonly used algorithm is gradient descent, which works by iteratively adjusting the weights $\mathbf{W}$ and biases $\mathbf{b}$ in a way that minimizes the function's gradient ($-\Delta f$), constituted of a vector of partial derivatives, representing the direction and magnitude, in which one may update the parameters in order to decrease the loss value.

$$-\nabla f_i = \left(-\frac{\partial f_i}{\partial x_1}, -\frac{\partial f_i}{\partial x_2}, \ldots, -\frac{\partial f_i}{\partial x_n}\right) \tag{4.9}$$

The process of gradient descent starts by selecting the learning rate hyperparameter λ, which determines how big of a change to the weights and bias we make for each minimizing step we take. The choice of this parameter has a large effect on training time, or convergence time, and training stability (Liu et al., 2019). Therefore, the simplified algorithm for gradient descent consist of firstly, choosing a random value for all parameters θ, composed of weights and biases for each neuron. Then, choosing the learning rate, and specifying the amount of iterations for calculating the gradients, or an objective, such as the loss not decreasing.

$$\theta = \{\boldsymbol{W_1}, b_1; \boldsymbol{W_2}, b_2; \ldots; \boldsymbol{W_n}, b_n\}$$

Algorithm 1 Gradient Descent Optimization

1: Initialize model parameters θ randomly or with initial values
2: Set learning rate λ
3: **while** not converged or not reached maximum iterations **do**
4: Compute predictions $F(\mathbf{X})$
5: Calculate loss $Loss_{\mathbf{Y}} = L(\mathbf{Y}, F(\mathbf{X}))$
6: Compute gradient of the loss: $\nabla_\theta Loss_{\mathbf{Y}}$
7: Update parameters: $\theta \leftarrow \theta - \lambda \nabla_\theta Loss_{\mathbf{Y}}$
8: **end while**

Gradient descent is a fundamental optimization technique to deep learning porblems. However, it has some limitations, such as sensitivity to the choice of the learning rate and slow convergence. To address these issues, more advanced optimization algorithms, such as Adaptive Moment Estimation (Adam), have been developed. Adam works by dynamically adjusting the learning rates for each parameter during training, effectively accelerating convergence and increasing overall stability(Kingma & Ba, 2014).

Adam works by randomly initializing the model parameters θ. Then, it maintains two moving averages, or moments, for each parameter, the first estimates m_t, which captures the mean of the gradients, and the second estimates v_t, which represents the uncentered variance of the gradients. For each iteration, Adam will calculate the gradient of the loss $\nabla_\theta Loss(\theta)_{\mathbf{Y}}$, and update the moments. To prevent initialization bias, Adam uses two hyperparameters β_1 and β_2. Using these bias-corrections techniques as scaling tools, then

the optimization algorithm uses the bias-corrector parameters to scale the moments and get bias-free moments estimates, $\hat{m}_t$ and $\hat{v}_t$, that will ultimately be used for updating the parameters (Kingma & Ba, 2014).

Algorithm 2 Adam Optimization Algorithm

Initialize model parameters θ randomly or with initial values
Initialize first moment estimates $m_t = 0$ for all parameters
Initialize second moment estimates $v_t = 0$ for all parameters
Initialize iteration $t = 0$
while not converged or not reached maximum iterations **do**
    Compute gradient of the loss: $g_t = \nabla_\theta Loss(\theta)_{\mathbf{Y}}$
    Update first moment estimate: $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    Update second moment estimate: $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    Bias-correct first and second moment estimates:
        $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$
        $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
    Update parameters: $\theta \leftarrow \theta - \frac{\alpha}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$
    Increment iteration t
end while

When running any of the algorithms above, the iterations are strictly dependent on two hyperparameters, batch size and epoch. The prior refers to the amount of data the model uses to calculate the gradient and update parameters θ, while the latter consists of the number of times that the whole dataset is used in the training process. As hyperparameters, the choice for the epochs and batch sizes is specific to the task and the computing resources available at hand. The usual tradeoff between training time and convergence is relevant to this choice (Brownlee, 2018).

$$iterations = \left(\frac{N}{batch\ size}\right) \cdot epochs \tag{4.10}$$

Where:

$$N : \text{Size of the training dataset}$$

While training the model, one has to monitor the loss in the validation test (the data that the model does not use for gradient descent) for each epoch, in order to avoid overfitting,

which usually shows the training loss decreasing consistently while the validation loss increases. Moreover, one may also include the metric on which the model will then be evaluated. This metric is dependent on the model and task (e.g. R^2 for regression problems, or accuracy for classification).

For this research, considering that predicting risk premiums is a regression problem, we use the R^2 metric to evaluate our models. R^2 as a measure of goodness of fit, it calculates the proportion of data in the outcome variable which is explained by the predictive variables $\mathbf{X}$ (Miles, 2005). The coefficient of determination, or R^2, is then calculated by subtracting to 1 the result of the sum of squares of residuals over the total sum of squares. Where $\mathbf{Y}_i$ is the vector with the predicted output from the model, $\hat{\mathbf{Y}}_i$ is a vector with the actual values, and $\bar{y}$ the mean of the actual values.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2}{\sum_{i=1}^{n}(\hat{\mathbf{Y}}_i - \bar{y})^2} \tag{4.11}$$

4.2.2 Regularization

In instances where overfitting arises during the training of a neural network, various methods are available to address this issue and enhance generalization. These approaches, collectively referred to as regularization methods, constitute a substantial area of ongoing research in the field of deep learning. In this section, we will discuss a few methods relevant to our research.

Firstly, there is L1 regularization, also known as the least absolute shrinkage and selection operator (LASSO). The primary objective of this method is to penalize large values of model weights, denoted as $\mathbf{W}i$, by incorporating the absolute values of these weights into the loss function. The parameter γ_{l1} value, acting as a scalar, governs the trade-off between fitting the data (minimizing the original loss) and maintaining sparse or zero model weights. A higher γ intensifies the regularization effect, driving weights towards zero (Nusrat & Jang, 2018). Notably, when opting for a substantial γ_{l1} value, L1 regularization facilitates feature selection by setting numerous weights to zero. This may become a problem if the l1 parameter is too high, which can then lead to a model that is overly simplified. This is because a high regularization parameter intensifies the effect of driving

the model weights towards zero. As a result, more coefficients are set to zero, leading to a sparser model. If this happens, one may encounter an underfitting model, meaning that the model is not capable of learning at all from the training data Mueller and Groble, 2023.

$$Loss_{\mathbf{Y}} = L(\mathbf{Y}, F(\mathbf{X})) + \gamma_{l1} \sum_{i=1}^{n} |\mathbf{W}_i| \tag{4.12}$$

Furthermore, a similar regularization method is Ridge, or L2, regularization. The goal of this method is similar to L1, in that both penalize large weight values by adding them to the loss function. However, instead of encouraging sparsity or a value equal to zero for the weights, l2 works by encouraging the weight to be small but not necessarily zero, resulting in more complex models, in which all variables influence the output. Furthermore, Cortes et al., 2012 research shows that Ridge regularization seems to work best for complex regression problems with a rich environment of variables such as sentiment analysis since it will be more likely to maintain the density, described by R. C. Moore and DeNero, 2011 as the proportion of feature weights in a model that are nonzero, than LASSO regularization.

$$Loss_{\mathbf{Y}} = L(\mathbf{Y}, F(\mathbf{X})) + \lambda_{l2} \sum_{i=1}^{n} \mathbf{W}^2 \tag{4.13}$$

Finally, the last regularizer relevant to this research consists of batch normalization. This technique is quite different from the aforementioned. L1 and L2 regularization focus on modifying the loss function to control the magnitude of model weights, but batch normalization operates at the layer level and has a different purpose (Nusrat & Jang, 2018). This method provides faster and more stable training in the neural network. Batch normalization works by stabilizing the distributions of the layer inputs, by adding layers that control the mean and variance of the distributions (Santurkar et al., 2018). Then, after normalization, this method uses two learnable parameters, gamma γ, and beta β. These parameters allow the model to learn the optimal scale and shift for the normalized values, giving the network the freedom to undo the normalization if necessary.

For x_i in $\mathbf{X}$:

$$\hat{x}_i = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \tag{4.14}$$

$$\text{BatchNorm}(x_i) = \gamma_i \hat{x}_i + \beta_i \tag{4.15}$$

4.2.3 Ensemble Neural Networks

With a fundamental understanding of the theory behind neural networks, and some of the many methods available to achieve great prediction performance, we have all we need to jump into the last part. Here, we discuss ensemble neural networks, and more specifically, averaging ensemble networks.

Ensemble neural networks refer to the technique of grouping different neural networks trained on the same task to achieve more stable predictions (Zhang et al., 2020). Various ensembling techniques exist, some of these techniques are stacking, voting, or weighted ensembles. The main idea behind ensemble networks is that depending on the seed used, a neural network will produce some results, as the seed is what dictates the initial parameters θ. Given this, some neural networks will perform better than others, and some may overfit while others do not, and all this on the same data and with the same topology. An ensemble of multiple networks solves this by training multiple models and combining its predictions, reducing the variance of the predictions and many times achieving better, and more consistent results(Li et al., 2021).

An ensemble network method relevant to our research is the averaging ensemble. Despite it being relatively simple, this ensemble method provides the aforementioned benefit of reduced prediction variance. It does so, by summing the predictions $\mathbf{Y}$ from each one of the models and then dividing the vector with the predictions with the total amount of models in the ensemble n. Averaging ensemble networks have proven their effectiveness having been used in multiple research fields and deep learning challenges such as the Detection and Classification of Acoustic Scenes and Events (DCASE) by Huang et al., 2019

$$\hat{Y}_{\text{ensemble}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{Y}_i \tag{4.16}$$

Beyond averaging ensembles, other ensemble strategies have been employed in the context of neural networks, such as voting, weighted, and stacked ensembles. Voting ensembles involve selecting the prediction from the network with the highest confidence score. Weighted ensembles assign different weights to each network's prediction based on their performance. Stacked ensembles incorporate the outputs of multiple networks as input features for a subsequent layer, further enriching the prediction process.

4.2.4 Shapley Values

Neural networks are commonly characterized as "*black boxes*" and this description holds truth in many cases when looking at neural networks alone. In linear regression, determining the significance of an independent variable can be relatively straightforward by calculating its P-value. However, interpreting neural networks is a much more complex task given the presence of hidden layers, which introduce a higher level of complexity, making it challenging to understand the relationships among variables and their importance to the outputs. Consequently, traditional methods like P-values may not be applicable in the same straightforward manner for neural networks. This complexity becomes an obstacle to our understanding of how individual variables influence the network's predictions.

As economists, we are deeply interested in causality. Therefore, the need to find a method to shed light on the black boxes. One such method is the use of Shapley values. The concept of Shapley values originates from the field of game theory. We can imagine a scenario where a group of players forms a coalition to achieve a common goal. However, not all players contribute equally to this goal. The question then arises: how should the rewards for achieving the goal be distributed among the players to ensure that the cooperation continues? This is where Shapley values come into play. They provide a way to fairly distribute the rewards based on each player's contribution (Fadel, 2022). Drawing parallels to the field of neural networks, we can think of the variables $\mathbf{X}$ as the players forming a coalition to achieve the outcome y. Just like in the game theory scenario, not all variables contribute equally to the outcome. The Shapley values, in this context, help us understand the contribution of each variable towards the outcome. Consequently, they help us quantify the importance of each variable in the prediction of the outcome.

Another way one may describe Shapley values of a variable, or reward for a player, in

game theory terminology, consists of taking the average of the variable/player marginal contributions over all possible orders of coalition formation. We may define the marginal contribution of a player to a coalition as the value of the coalition with the player minus the value of the coalition without the player. Another benefit of using Shapley values is that it enables contrasting explanations, meaning that it provides the flexibility to contrast a prediction with a more specific subset of data or even with an individual data point (Molnar, 2023). This allows for a more targeted examination of the factors influencing a specific prediction, which may be particularly useful when looking into the different predictions in recessions and expansions.

Looking into how to calculate the Shapley value ϕ for variable x_i in vector $\mathbf{X}$, first one calculates the marginal contribution of a player or feature, that is, how much does a model F with a set of variables $\mathbf{S}$, including variable x_i, differs against a model with a set of variables $\mathbf{S}$, without the variable for which the Shapley value is being calculated. Then, one may calculate the weight importance by calculating the factorial of the set $|\mathbf{S}|$ and multiplying it by the factorial of the number of remaining variables in $\mathbf{X}$ after excluding those in set, or coalition, $\mathbf{S}$, and subtracting 1 for the variable x_i and dividing over the factorial of the total amount of variables $|\mathbf{X}|!$. Lastly, we take the average summing over all possible coalitions $\mathbf{S}$ that can be formed by taking subsets of the vector of variables $\mathbf{X}$ excluding x_i and then dividing over the total number of possible permutations(or coalitions) N in which the variables $\mathbf{X}$ can cooperate (Fadel, 2022).

$$\phi_{x_i}(\mathbf{X}) = \frac{1}{N} \sum_{\mathbf{S} \subseteq \mathbf{X} \setminus \{x_i\}} \frac{|\mathbf{S}|!(|\mathbf{X}| - |\mathbf{S}| - 1)!}{|\mathbf{X}|!} [F(\mathbf{S} \cup \{x_i\}) - F(\mathbf{S})] \tag{4.17}$$

4.3 Research Methodology

This part of the chapter covers all the steps taken to achieve the results showcased in the next chapter. It is meant to be a guide to replicate the results and understand all the choices made. First, we introduce the software used in the project and then describe how we achieved the results for each of the hypotheses.

4.3.1 Software

This research can be characterized as a deep learning project applied to an economic problem. As with any other deep learning project, where neural networks and large datasets are central, the computation needed is quite large. To analyze and process the data we use Python on Google Colab, In this step, a CPU with 51 Gigabytes of RAM is used. On the same runtime, we build the neural networks and the calculation for the Shapley Values. The ANN interface used in this paper is Keras, a high-level library, which in turn acts as an interface for Google's TensorFlow library (Pohl, 2023).

4.3.2 Answering Hypothesis 1

Regarding the first hypothesis in our research, *the inclusion of a larger count of macroeconomic variables in addition to the stock characteristics, produces a better prediction performance*, we are looking to benchmark our results with the ones by Gu et al., 2020 showcased in Table 2.1. To do so fairly, we plan to use a methodology as similar as possible to the one described in their paper, with only some small changes, some of them described in the previous sections.

Firstly, we have five different neural networks, each of them with a different amount of hidden layers, ranging from 1 to 5. One difference between our research method and Gu et al., 2020 is that we choose wider layers, with the first layer after the input layers always having 256 neurons. The reason behind this choice is that our dataset is richer in terms of predictors, resulting in more interactions and relationships among the variables that the models will have to take into account. Consequently, a greater number of neurons per layer is necessary. Each consecutive layer contains the amount of neurons from the previous layer divided by two. Therefore, our 4-layered neural network, for example, will consist of 256 neurons on its first hidden layer, 128 on its second, 64 on the third, and 32 on the last layer, altogether totaling 480 neurons.

The input layer has as many neurons as variables $\mathbf{X}$ in the dataset. We use a rectified linear unit (ReLU) activation function for each one of the hidden layers. After each dense hidden layer, we add a batch normalization layer, with as many inputs as neurons in the prior hidden layer. Finally, for the output layer, we incorporate a single-neuron layer

with L1 and L2 kernel regularization. Given that we are tackling a regression problem, there is no activation function applied in this layer. The rate across all models for Lasso regularization (L1) is 0.001 and 0.1 for Ridge regularization (L2). We keep L1 to a minimum since we are looking to include as many variables as possible in predictions, and a higher L1 would potentially set some weights to 0 effectively omitting some variables. Since L1 is small we increase L2 to achieve the benefits of regularization.

Table 4.1: Depth and Size of Each Model

	NN1	NN2	NN3	NN4	NN5
Hidden Layers	1	2	3	4	5
Parameters	157697	190977	199425	201601	202177

Then, for the loss function of the model we choose mean squared errors (MSE), which is inversely correlated with R^2, that is, when we train the model, we will be minimizing the MSE and consequently, maximizing the coefficient of determination. We use the Adam optimization algorithm, together with a learning rate of 0.01.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2 \tag{4.18}$$

Furthermore, when choosing the iterations, we use a relatively large batch size parameter of the length of the train dataset over 50, and 200 epochs. Therefore, looking at Equation (4.10), we have a total of 10000 iterations. The amount of iterations, together with the learning rate, may seem slightly high, but this is adjusted for by two callbacks. A Keras callback consists of an object that can perform actions at different stages of the training process (Keras, 2023). They are a powerful form of customization. The first callback used is learning rate shrinkage, which works by reducing the learning rate as the gradient gets closer to 0, that is, whenever the optimization is nearing convergence. Then, regarding the number of iterations, we use the early-stopping callback which functions by stopping the training whenever the monitored metric stops improving. The metric chosen to monitor is the validation loss, and then we choose patience, or the number of epochs without

improving in the monitored metric equal to 5. Therefore, the total iterations may vary from model to model, but it will rarely run through the 10000 maximum.

Regarding the ensemble networks, we use the same ensembling methods as in the Gu et al., 2020 paper, by initializing the same model on different seeds, and then summing the prediction vectors and dividing it by the amount of models in the ensemble, as showcased in Equation (4.16). Since the number of models used in the paper is not specified, we decided to use 30 models, per ensemble. When reporting the R^2 results for each model, we chose the monthly out-of-sample stock-level prediction performance (R^2) equal to the 30-model ensemble value.

4.3.3 Answering Hypothesis 2

When looking to test our second hypothesis whether some variables may have a larger prediction power than others and if these may vary in periods of recession and expansion, it will involve looking inside the "*black boxes*". We will use the predictions from the models above and divide those predictions into two groups, one with predictions during recessions and another one with predictions during expansion periods. For each group, we will select a sufficiently large representative sample of the data population by simple random sampling, and calculate the absolute mean Shapley values for each input x_i. Absolute, because a large negative Shapley value is as important as a large positive one, and mean because different dates and stocks will have different values.

To understand the importance of each variable as a percentage of the prediction output, we will assign a scaled value to each variable based on the mean absolute Shapley value, with all scaled values adding up to 1. A benefit of Shapley values is that they allow for

Moreover, we will look into the most important variables for model output for each group, that is, the variables with the highest absolute mean Shapley value, and then look at how the groups of variables rank against each other, to infer what variables are important in recessions and which ones are important in expansions. Then, we will add the scaled values for the predictor groups for the purpose of seeing how the asset pricing models weigh the variables during recession and expansion periods. Lastly, we look at the distributions of the mean absolute Shapley values to grasp how relevant on average is each variable, and to what extent some of our variables are redundant.

5 Analysis

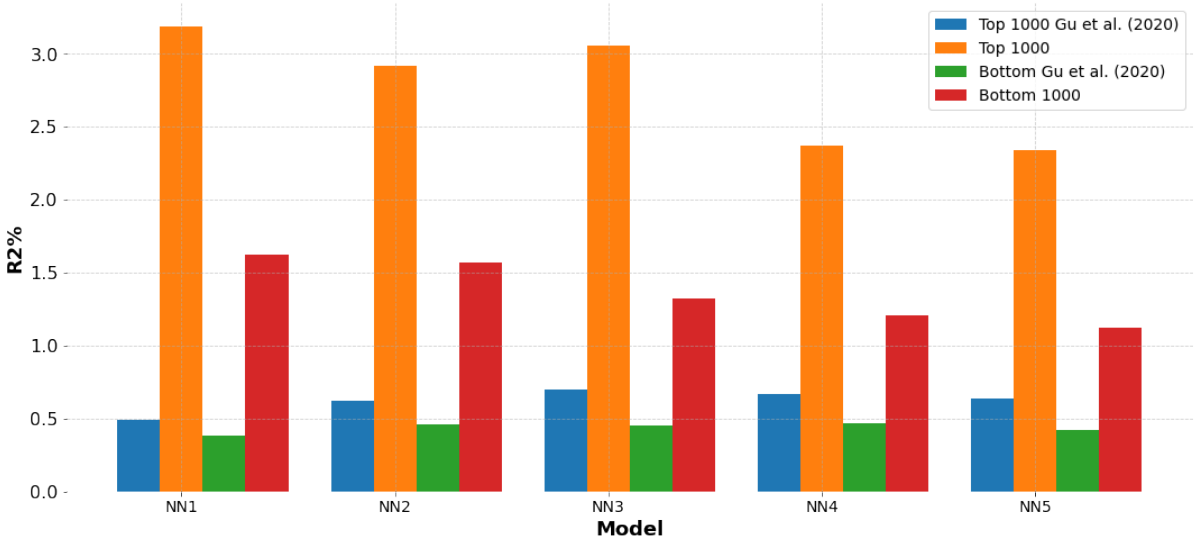
In this chapter, we look at the results of the conducted research. Our goal is to showcase them in depth from a descriptive matter. Then, in the next chapter, we will discuss them, and draw conclusions.

5.1 Results For Hypothesis 1

Looking back to our first hypothesis, we are looking to see if the inclusion of macroeconomic variables, in addition to the stock characteristic variables used in Gu et al., 2020, have an effect on predictive performance measured in monthly out-of-sample stock-level R^2. If so, we would expect the R^2 from comparable models to be larger than the one reported in the Gu et al. paper. In the figure below, we show the summarized results that seek to shed light on our hypothesis.

Table 5.1: Monthly out-of-sample stock-level prediction performance (percentage R^2_{oos})

	NN1	NN2	NN3	NN4	NN5
Top 1000 Gu et al	0.49	0.62	0.70	0.67	0.64
Bottom 1000 Gu et al	0.38	0.46	0.45	0.47	0.42
Top 1000	3.19	2.92	3.06	2.37	2.34
Bottom 1000	1.62	1.57	1.32	1.21	1.12



There are a couple of insights from the results above. Firstly, the most visible is the difference in prediction performance in the models from the Gu et al. study compared to the ones developed here. When looking at firms with large market equity (Top 1000) and using the same target variables, we achieve an R^2_{oos} of 3.19% for our best model, while the benchmark reports 0.70% as their best result, that is, we report an increase of 355% in monthly out-of-sample stock-level prediction performance. Along the same lines as the large market value firms, the monthly out-of-sample stock-level prediction performance result of the best for the low market equity firms achieved in our study (1.62%) represents a 245% increase in performance compared to the benchmark (0.47%).

Another interesting insight, and common to the Gu et al., 2020 paper, is the significant difference in prediction performance between large (Top 1000) and small (Bottom 1000). Consequently, the *erratic* behavior of small stocks, based on low liquidity, as described in the benchmark paper, seems to be supported by our results. However, the inclusion of macroeconomic variables attains a better prediction of the risk premium behavior, highlighting its importance.

Moreover, when observing the neural network topology, our models show that a shallower neural network tends to produce better results, both when considering large and small stocks, while Gu et al., 2020 results indicate that narrow and deeper (up to 4 hidden layers) seems to give them better results. In this case, it is slightly more complicated to reach an interpretation further than what is previously described in the methodology chapter as the benefit of wide hidden layers when working with predictor-rich datasets.

Lastly, in the following figures, we show the percentage R^2_{oos} prediction performance achieved for each individual model, and for each ensemble model per neural network topology. The results display an interesting property common to all topologies, consisting of decreased variance of prediction performance when using ensemble networks. This property shows a nearly stable prediction in all the neural networks, for ensembles composed of over 20 individual models. While some of these individual models may achieve better predictive performances in unique cases, when looking at using the model without the ability to test, as we have done now, individual models are not reliable and one should clearly implement ensembles.

Figure 5.1: Individual and Ensemble R^2_{oos} prediction performance: NN1 Model

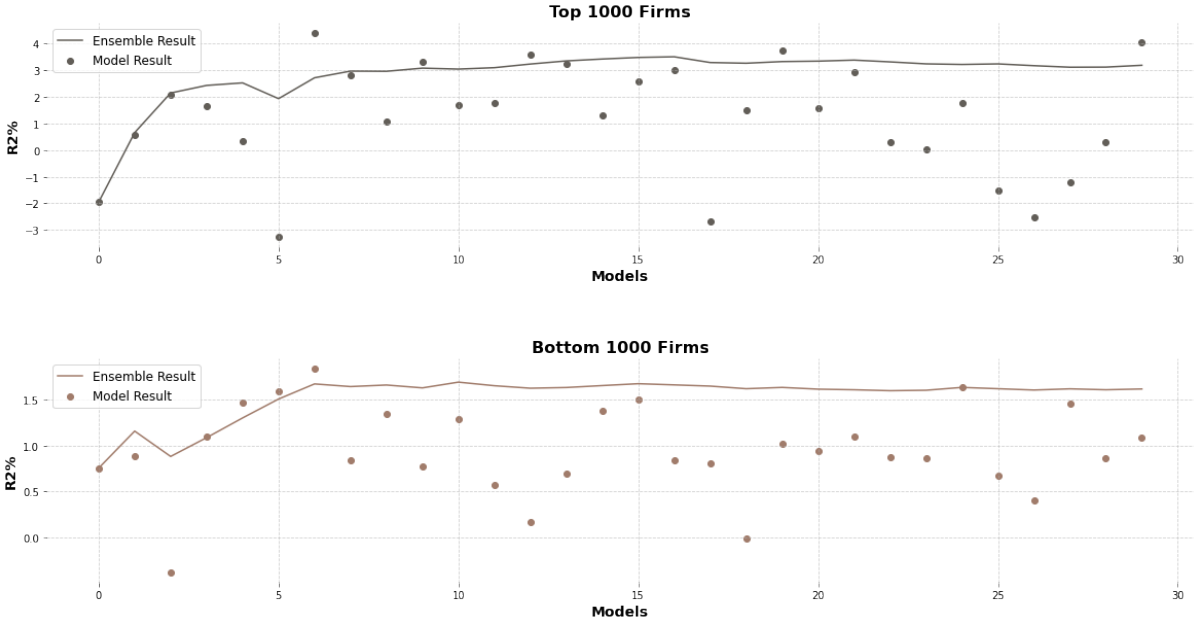


Figure 5.2: Individual and Ensemble R^2_{oos} prediction performance: NN2 Model

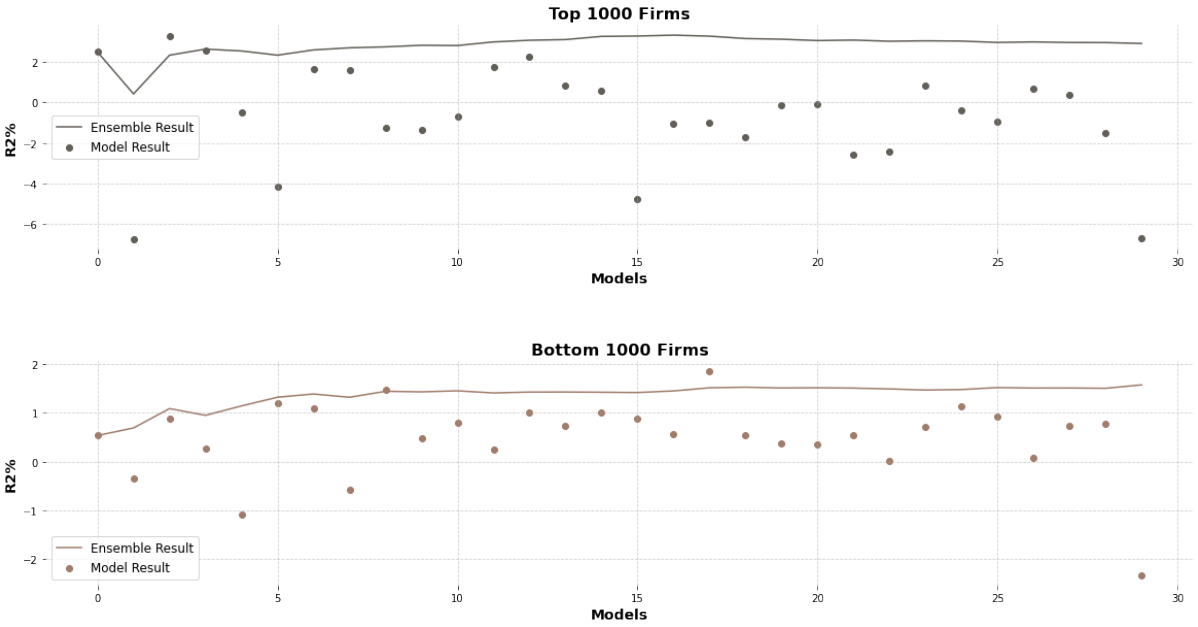


Figure 5.3: Individual and Ensemble R^2_{oos} prediction performance: NN3 Model

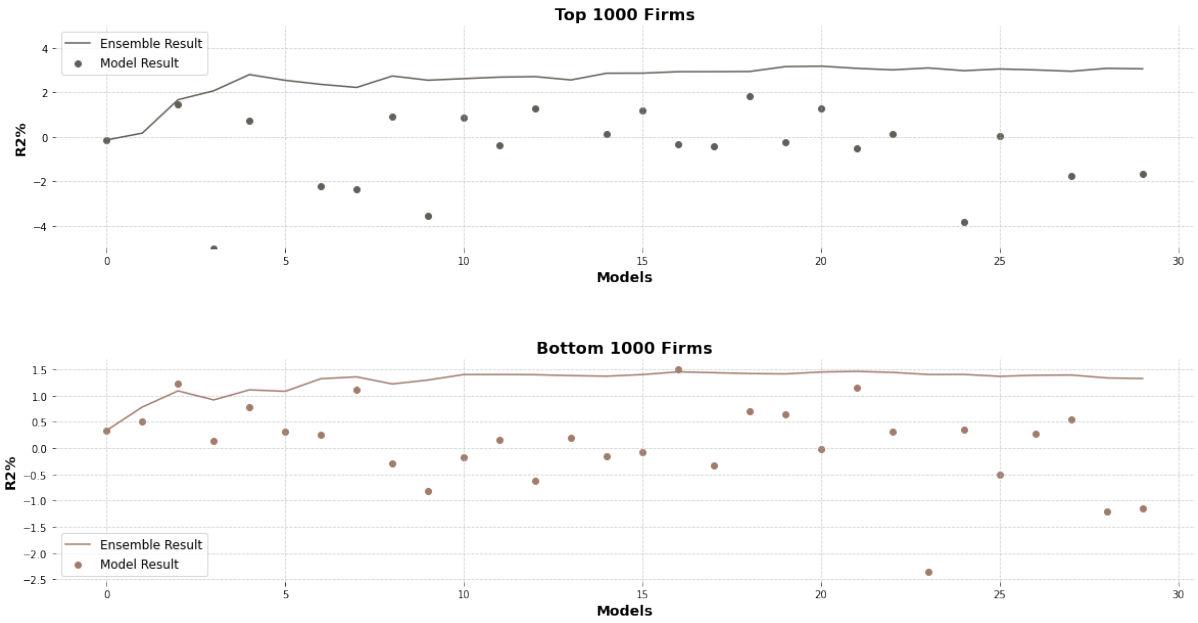


Figure 5.4: Individual and Ensemble R^2_{oos} prediction performance: NN4 Model

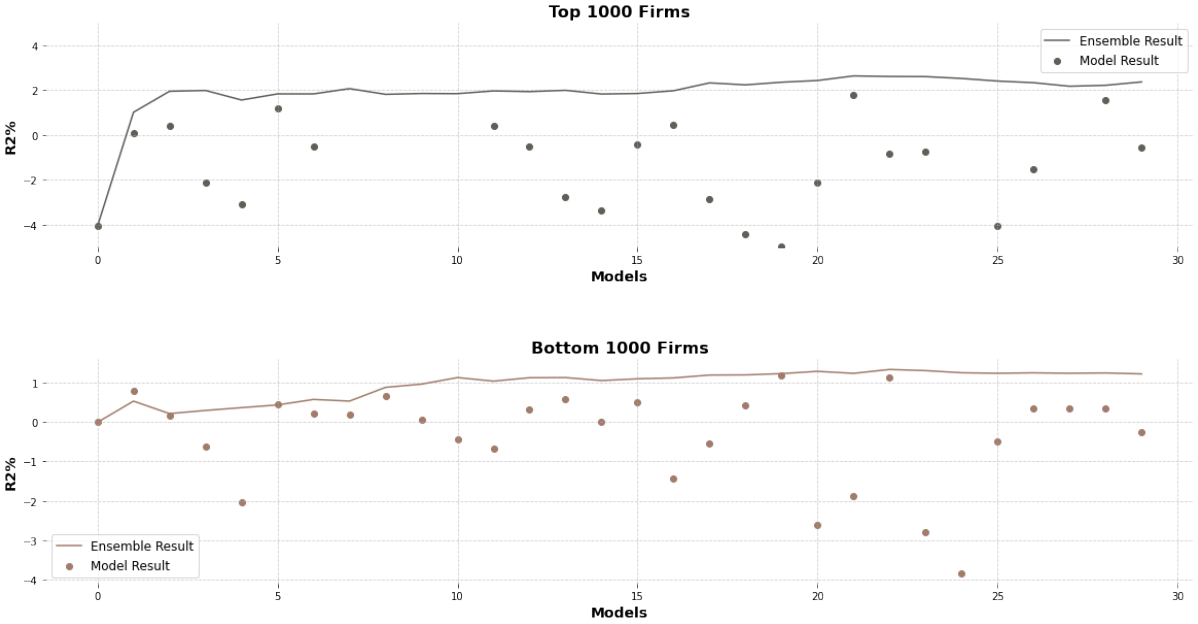
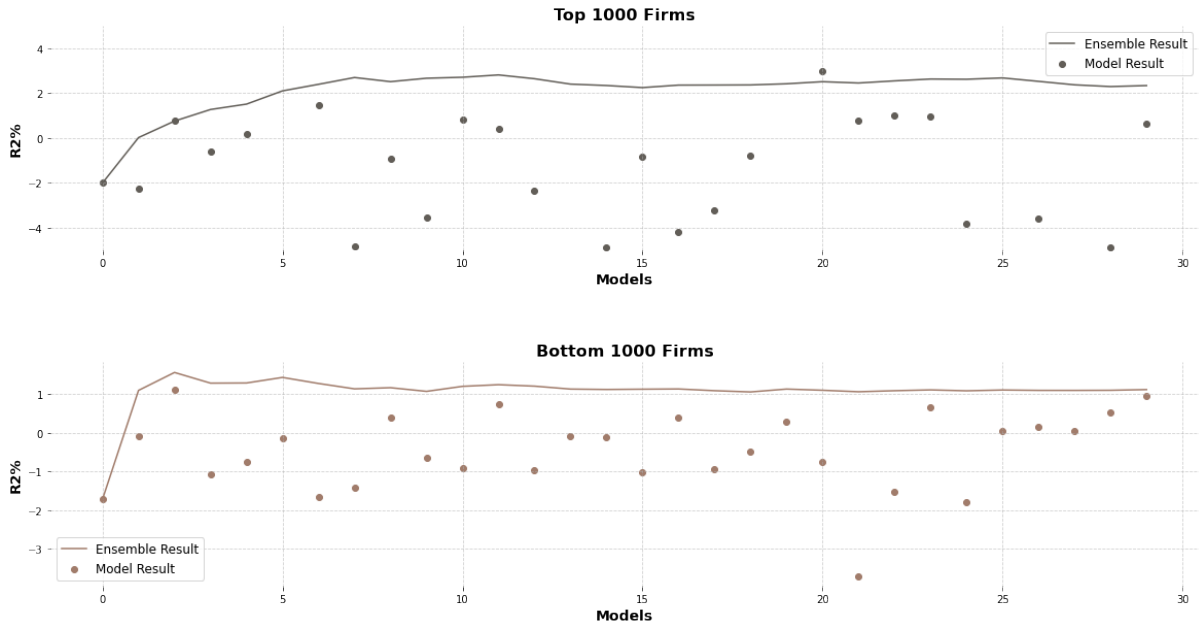


Figure 5.5: Individual and Ensemble R^2_{oos} prediction performance: NN5 Model



5.2 Results For Hypothesis 2

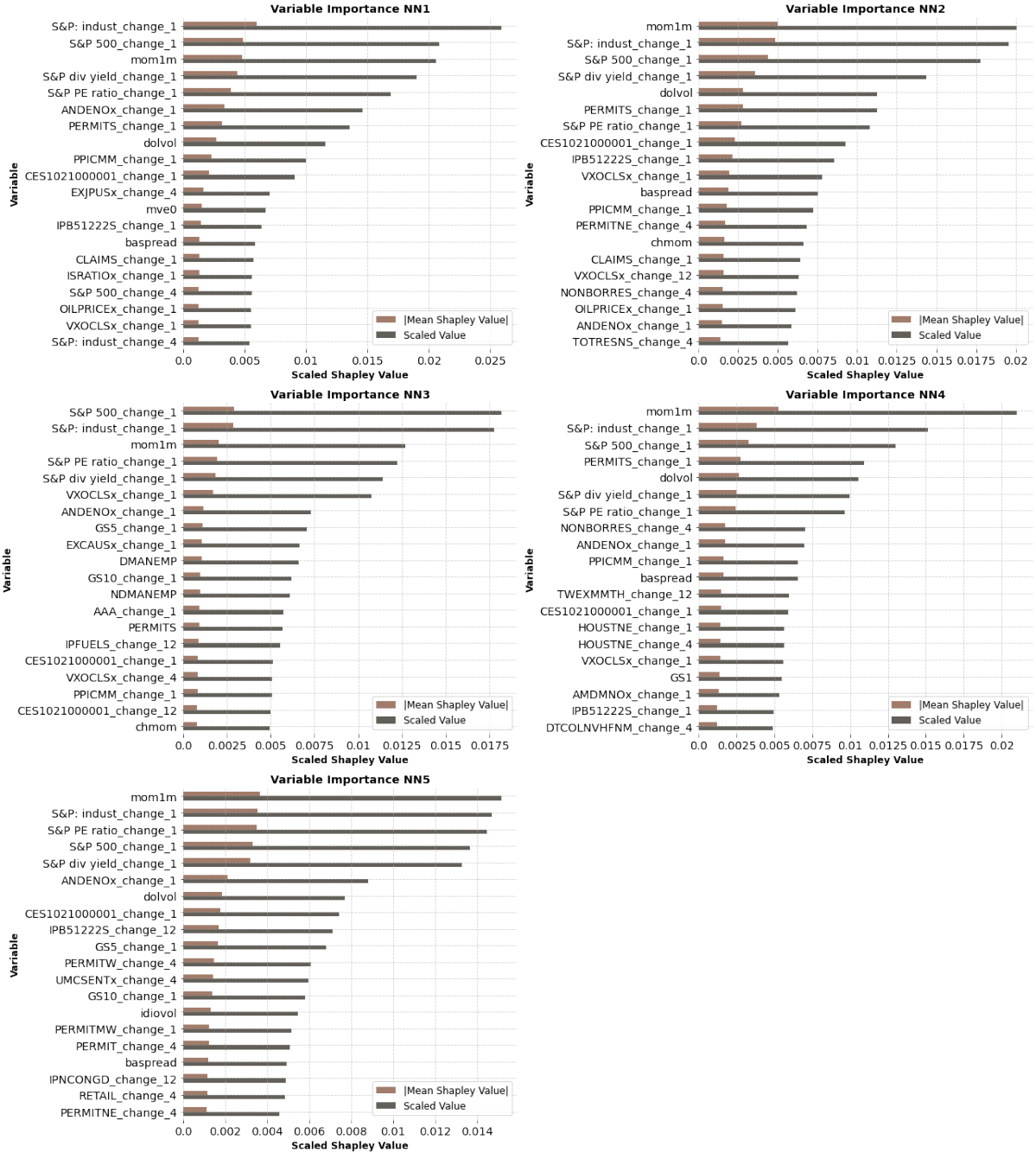
When looking to answer the second hypothesis, we will be looking at a couple of concepts from two different perspectives, recession periods, and expansion periods. For each one of these perspectives and for each one of the model architectures, we analyze which of the variables are the most important, by selecting the top 20 variables, based on the absolute mean Shapley value. Then, we look into the importance of the different predictor groups (output and income, labor market, consumption and orders, orders and inventories, money and credit, interest rate and exchange rate, prices, stock market, and stock characteristics), and will finish by evaluating the absolute Shapley value distributions across predicting variables in order to understand whether all variables are equally valuable to the model.

In the previous hypothesis, we look at a divide between large and small firms, based on the market value of equity. Here, however, we are interested in the distinction between expansion and recession periods and will look at the results from those two different perspectives. Distinguishing between these two economic periods allows us to expand on our main theory. We now know that adding more variables results in better forecasting performance, and following this, we hope to uncover the distribution of importance among those variables. Moreover, the results will expand our understanding of the relevance of

each part of the economy, as represented by the aforementioned groups, in expansion and recession periods.

In the Figure below we can observe the 20 most important variables for each model for the expansion periods, the mean absolute Shapley value is depicted in brown while the scaled values, adding altogether to 1, are colored in grey. Then we specify the 20 most common variables across the 5 groups.

Figure 5.6: Variable Importance In Expansion Periods (measured in scaled Shapley value)



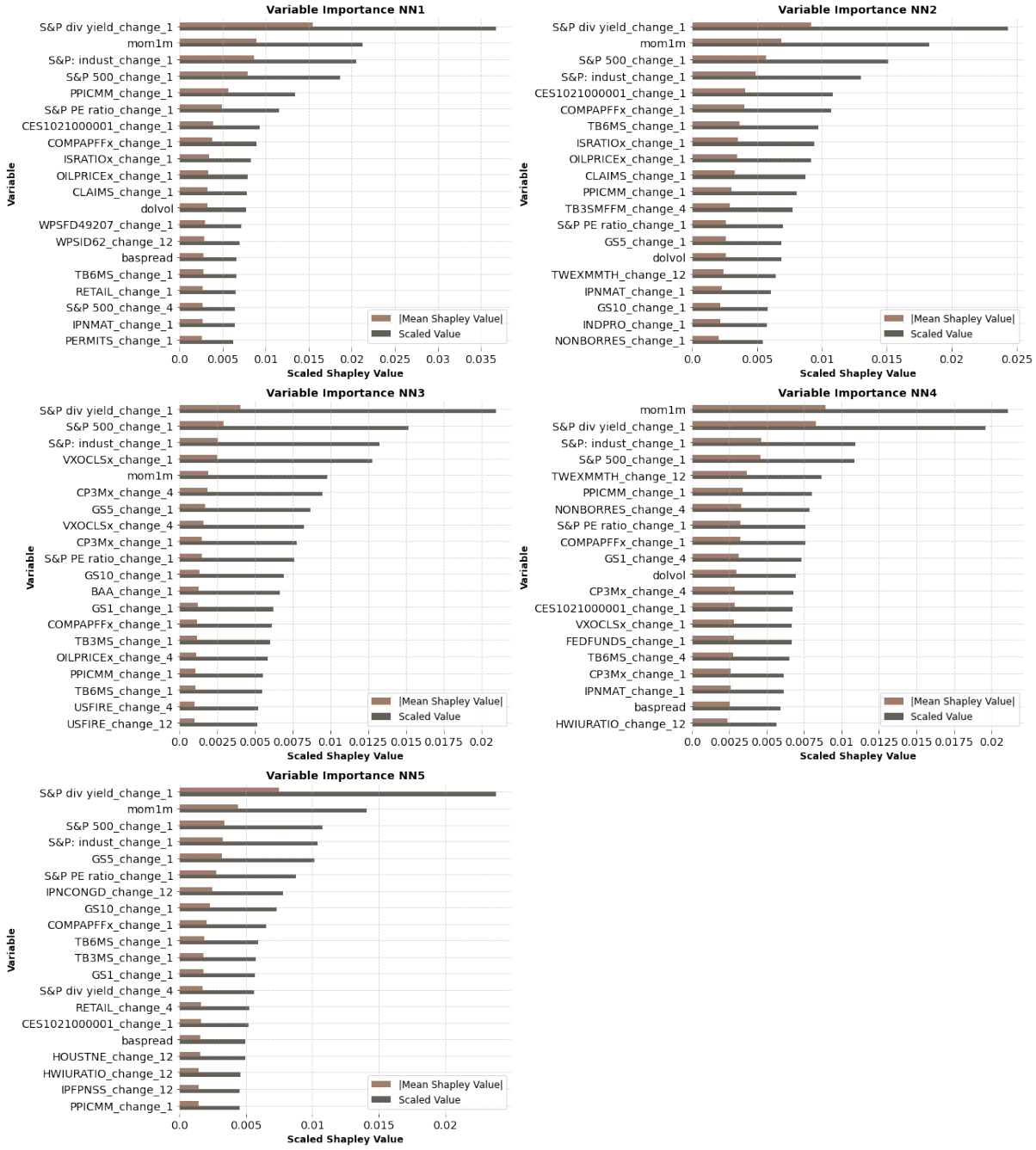
Looking at the figure above we see that most of the variables are common to all the different networks. In addition, going from shallow to deep topology, the concentration of importance is at its highest for the shallowest network (NN1) with the most important variable making up 2.6% of the output prediction, and at its lowest for the deepest network (NN5) with the most important variable representing 1.5% of the output prediction.

The 20 most common variables in the above graphs are: 1-month change in S&P's Common Stock Price Index: Industrials (S&P: indust_change_1), 1-month change in S&P's Common Stock Price Index: Composite (S&P 500_change_1), short-term reversal (mom1m), 1-month change in S&P's Composite Common Stock: Dividend Yield (S&P div yield_change_1), 1-month change in S&P's Composite Common Stock: Price-Earnings Ratio (S&P PE ratio_change_1), 1-month change in New Orders for Nondefense Capital Goods (ANDENOx_change_1), 1-month change in All Employees: Mining and Logging: Mining (CES1021000001_change_1), Dollar Volume (dolvol), 1-month change in PPI: Metals and metal products (PPICMM_change_1), Bid Ask Spread (baspread), 1-month change in Volatility Index (VXOCLSx_change_1), 1-month change in New Private Housing Permits: South (PERMITS_change_1), 1-month change in Industrial Production: Residential Utilities (IPB51222S_change_1), 1-month change in Initial Claims (CLAIMS_change_1), 1-month change in Crude Oil, spliced WTI and Cushing (OILPRICEx_change_1), 4-month change in New Private Housing Permits: North East (PERMITNE_change_4), Momentum Change (chmom), 4-month change in Reserves Of Depository Institutions (NONBORRES_change_4), 1-month change in 5-Year Treasury Rate (GS5_change_1), and 1-month change in 10-Year Treasury Rate (GS10_change_1). The rest of the full description of the other variables showcased above but not among the 20 most common variables can be consulted in **Appendix**.

We see that among the 20 most common important variables for models predicting risk premia in expansion periods, only 4 are variables from the Gu et al., 2020 study.

In the Figure below we can observe the 20 most important variables for each model for the recession periods, the mean absolute Shapley value is depicted in brown while the scaled values, adding altogether to 1, are colored in grey. Then we specify the 20 most common variables across the 5 groups and discuss the differences with the variables on the expansion period.

Figure 5.7: Variable Importance In Recession Periods (measured in scaled Shapley value)



The first thing one may observe is that the variable importance concentration trend, as we saw in the expansion periods, continues to be present in recession periods. This may be interpreted as being a neural network property (the deeper a topology the lowest the concentration of importance (as in contribution to output value) in a variable. In the case of recession, however, we do see a more exaggerated concentration, with the highest

importance value of 3.7% from the shallowest network to the highest importance value of 2.4% in the deepest model.

When looking at the 20 most common variables in the graphs above, we find an overlap of 14 variables with those variables that are most important in expansion, those being: S&Pdivyield_change_1, mom1m, S&P:indust_change_1, S&P500_change_1, PPICMM_change_1, S&PPEratio_change_1, CES1021000001_change_1, dolvol, baspread, GS5_change_1, GS10_change_1, OILPRICEx_change_1, CLAIMS_change_1, VXOCLSx_change_1.

Then the remaining 6 highly-important variables present in the top 20 in for recession prediction, but not for expansion prediction are: 1-month change in 3-Month Commercial Paper Minus Effective Federal Funds Rate Spread(COMPAPFFx_change_1),1-month change in 6-Month Treasury Bill (TB6MS_change_1), 1-month change in Industrial Production: Nondurable Materials (IPNMAT_change_1), 1-month change in Total Business: Inventories to Sales Ratio(ISRATIOx_change_1), 12-month change in Trade Weighted U.S. Dollar Index: Major Currencies (TWEXMMTH_change_12), and 4-month change in 3-Month AA Financial Commercial Paper Rate (CP3Mx_change_4).

One interesting result is that despite the fact that the stock characteristic variables from Gu et al., 2020 represent just 10% of the top 20 most important variables for any of the periods, those 4 variables that we find among the most relevant are also found to be the most relevant on the Gu et al., 2020 paper, from which we can then corroborate the results.

In the following page, we showcase the cumulative scaled importance value for each one of the groups of variables and for each different neural network model.

Table 5.2: Variable Group Importance as Sum of Scaled Values(measured in % percentage)

	Expansion					Recession				
	NN1	NN2	NN3	NN4	NN5	NN1	NN2	NN3	NN4	NN5
Outcome & Income	7.97	8.54	9.27	9.30	10.11	8.76	10.91	10.49	10.64	12.90
Labor Market	15.14	**17.44**	**18.16**	**15.87**	**16.20**	16.13	17.88	18.29	16.71	17.98
Consumption & Orders	9.22	7.96	7.91	9.33	10.01	6.81	6.35	5.47	6.89	6.83
Orders & Inventories	7.13	6.88	6.43	7.16	8.34	7.05	6.83	5.82	6.47	6.57
Money & Credit	7.60	8.59	7.41	8.75	7.70	6.02	6.30	5.76	7.51	5.85
Interest & Exchange Rates	12.53	15.11	14.30	14.24	13.20	**17.08**	**21.06**	**22.34**	**20.44**	**20.01**
Prices	12.50	10.81	12.39	12.89	11.54	15.02	12.11	13.26	12.76	12.39
Stock Market	12.04	9.45	9.71	7.47	8.94	12.64	8.67	10.83	8.06	8.47
Stock Characteristics	**15.86**	15.22	13.97	14.99	13.95	10.48	9.88	7.73	10.53	9.00

We can observe some clear differences between periods of expansion and periods of recession. In the prior, we see that the labor market contributes the most to the output, up to over 18% in neural networks with 3 hidden layers. The variables, used in the Gu et al., 2020 study, seem quite relevant to the study since they represent between 16 to 14% of importance, which is close to the percentage they represent over the total number of predictors. Then, the least important group of variables for predicting risk premiums are orders and inventories, and money and credit, both of them ranging a contribution from 7% to 8%.

In recession periods, the labor market group of variables remains a significantly important group to outcome prediction contribution, as a matter of fact, the contribution is even higher than on expansion cycles. However, the most important group of variables in downturn periods is interest and exchange rates, and by a large margin. Variables in this group represent 17% of the output at its lowest percentage in the most shallow neural networks while representing over 22% of the outcome at its highest for networks composed of 3 hidden layers. This time, consumption and orders, together with orders and inventories, and money and credit, are the groups that contribute the least to the output predictions. An interesting phenomenon visible in recession periods is the concentration of importance, meaning that the most important groups represent a much larger percentage, while the least important groups have an even lower importance.

This table uncovers some clear dynamics in the pricing of assets depending on the economic period. One important takeaway is the importance of stock characteristics. As we have discussed, the field of asset pricing has focused on researching individual stock characteristics, under the form of factor models. We now see that all things equal, the relevance of such research, in terms of forecasting power, is considerably lower in periods of recessions. Furthermore, we described the Federal Reserve as the pilot of the economy, implying that it has the power to steer it. This claim is supported by our results, there is an ample difference in importance in the Interest & Exchange Rates, that is, the main tool of the Federal Reserve. Intuitively, the relevance of the Prices groups is higher in recession periods than in expansions.

Table 5.3: Descriptive Statistics From The Scaled Absolute Mean Shapley Values

	Expansion					Recession				
	NN1	NN2	NN3	NN4	NN5	NN1	NN2	NN3	NN4	NN5
Count	610	610	610	610	610	610	610	610	610	610
Mean	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016
Standard Deviation	0.0023	0.0020	0.0017	0.0017	0.0016	0.0025	0.0020	0.0018	0.0018	0.0017
Minimum	0.000031	0.000026	0.000036	0.000017	0.000027	0.000025	0.000017	0.000069	0.000034	0.000040
25%	0.00062	0.00068	0.00082	0.00080	0.00084	0.00050	0.000561	0.000752	0.00069	0.00074
Median	0.0011	0.0011	0.0013	0.0012	0.0013	0.0010	0.0010	0.0011	0.0011	0.0012
75%	0.0020	0.0019	0.0019	0.0020	0.0019	0.0019	0.0020	0.0019	0.0020	0.0020
Maximum	0.026	0.020	0.018	0.021	0.015	0.037	0.024	0.021	0.021	0.024

Furthermore, we explore the descriptive statistics from the scaled absolute mean Shapley values. Firstly, as the values are scaled, the mean is equal across economic cycles and models. Then, however, the standard deviation is higher for recessions than expansion periods and higher for more shallow models. The standard deviation in this context can understood as the spread of the values, meaning that when weighting the variables, the models do so less harmonically in shallower networks and recession periods. This can be further proved by looking at the maximum values for the models. One hidden layer network, which is the one with the highest prediction power, has the highest maximum values as priorly seen in the importance tables, from which we may infer that the reason they perform as well as they do, is because they are able to discriminate among important and less important variables, and weight them accordingly. Moreover, when looking at the interquartile range, or the spread between the 25% and 75% quartiles, the shallow networks have the largest spread, and following the trend, recessions do as well. This can be interpreted as another proof revealing the greater variability in importance among variables in the predictor set.

Lastly, we observe an interesting phenomenon, we see that the percentage of importance represented by macroeconomic groups in expansions is equal to 85.21% on average. At the same time, the same measure in recession periods is 90.48% representing a 6.2% increase. Therefore, according to these results, one may say that macroeconomic variables are even more important in periods of economic recession.

6 Conclusion

The theory on which we based our first hypothesis, *the inclusion of a larger count of macroeconomic variables in addition to the stock characteristics, produces a better prediction performance*, consisted of the representation of the market and its participants, as an information aggregator. Meaning that the collectivity is capable of weighting and inferring risk premiums from thousands, or millions of parameters. Until recently, the models used for asset pricing have focused on finding a few relevant factors to predict risk premiums. Our results prove that there is a benefit in expanding the amount of predictors. We discussed that the prior focus on factor model pricing may be a result of the techniques available, as shown in Gu et al., 2020, it is not possible to run regressions with high dimensional vectors of predictor variables.

The results for the first part of our study seem to support our hypothesis. The fact that the technical part of the study was designed to be as similar as possible to the benchmark from Gu et al., 2020, was done to discard the possibility of the predictions being superior as a result of the technologies used. The inclusion of our variables resulted in a 355% increase in monthly out-of-sample stock-level prediction performance, achieving an R^2_{oos} of 3.19% for our best model predicting large market value equity stocks. When predicting stocks with a small market value of equity, the performance increase against the benchmark was nearly as high, at R^2_{oos} of 1.62%. These results support the pattern described by Gu et al., showing that stocks with small market value of equity are harder to predict. They attribute it to erratic behavior based on low liquidity, based on our theory we may describe it as fewer participants aggregating the information, translating into less smooth predictions. Although not necessarily related to our theory, we find an additional insight relevant to the technical part of the paper, highlighting the importance of ensemble neural networks. Showing that the higher the amount of networks used in an averaging ensemble, the lower the variance of the resulting predictions.

The second hypothesis, *Some variables may have a greater prediction power than others, these may vary in periods of recession and expansion*, consisted of expanding our theory, and shedding light into our forecasting models. We used Shapley values to understand whether our hypothesis was correct. The results, seem to support the hypothesis since we

see that not only do the variables themselves have different degrees of importance, but we see that the degrees also change depending on the economic cycle.

We see that labor market and stock characteristics seem to carry the most importance, as they result in having the highest impact on model output during expansion periods. We then observe that the most important groups in recession periods are not the same as in expansions, and what is even more interesting is that the main three groups in recessions, interests and exchange rates, labor market, and prices, are extremely correlated to what the Federal Reserve monitors during recessions, labor market and prices, and the tools it uses to alter them, interest and exchange rates. Moreover, the results in this second part show that stock characteristics' importance is considerably smaller in recession periods, displaying a pricing dynamic depending on the economic cycle.

These results are fascinating, they support the idea of market aggregators following closely the variables relevant to Federal Reserve decisions. They also show the benefits of using neural networks for pricing, proving that these methods are capable of weighing information logically and reasonably according to what we would expect. Furthermore, the insights we describe, challenge the view of neural networks as *"black boxes"*, hopefully giving place to more generalized adoption in their use in the field of economics.

6.1 Going Forward

In this research, the expansion of the predictor input vector was crucial to achieve higher prediction performance. Based on the main theory behind our paper, this performance could be even higher with an even richer input vector. Up until now, we have taken into account macroeconomics and particular stock characteristics. However, many more parameters relevant to the market aggregate exist. Some of these parameters for which data is relatively accessible, include but are not limited to, news, social media discussions, or speeches. Other parameters, may yet to be discovered. We believe the inclusion of such variables in predictive models will not only increase prediction performance but may also expand our understanding of asset pricing and economics in general.

Moreover, classifying neural networks as completely inscrutable models may no longer be correct. We showed that the use of Shapley values is capable of shedding light on how important each variable is for a model. Furthermore, with the increase of

regulatory scrutiny over generative artificial intelligence, new techniques, such as the ones described in the *Decomposing Language Models Into Understandable Components* article by Anthropic, 2023, are being developed at an outstanding pace to further our understanding of mechanistic interpretability of neural network models. As a result of this, economists should reconsider their stance on deep learning methodologies, which in many cases, has proven to be superior to conventional methods, while allowing us to infer causality from its results.

References

Anthropic. (2023). Decomposing language models into understandable components.

Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J., Kakade, S., Wang, H., & Xiong, C. (2021). How important is the train-validation split in meta-learning? *Proceedings of the 38th International Conference on Machine Learning*, 543–553.

Berasategui Gallego, J. (2023). Master thesis data. *Hugging Face*. https://huggingface.co/datasets/juanberasategui/Master_Thesis_Data/tree/main

Blinder, A. S. (2023). Landings, soft and hard: The federal reserve, 1965–2022. *Journal of Economic Perspectives*, *37*(1), 101–120.

Brownlee, J. (2018). What is the difference between a batch and an epoch in a neural network. *Machine Learning Mastery*, *20*.

Callen, T. (2008). What is gross domestic product. *Finance & Development*, *45*(4), 48–49.

Cochrane, J. (2009). *Asset pricing: Revised edition*. Princeton university press.

Cœuré, B. (2013). The economic consequences of low interest rates. *International Center for Monetary and Banking Studies*.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). L2 regularization for learning kernels.

Damodaran, A. (1999). Estimating equity risk premiums.

Effective federal funds rate. (2023). *Markets and Policy Implementation*.

Fadel, S. (2022). Explainable machine learning, game theory, and shapley values: A technical review.

Fei, N., Gao, Y., Lu, Z., & Xiang, T. (2021). Z-score normalization, hubness, and few-shot learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 142–151.

Fei-Fei, L., & Deng, J. Imagenet. where have we been? where are we going? In: 2017.

Ghedira, H., & Bernier, M. (2004). The effect of some internal neural network parameters on sar texture classification performance. *6*, 3845–3848 vol.6.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Hall, R. (2003). Business cycle dating. *National Bureau of Economic Research*.

Harvey, C. R., & Liu, Y. (2019). A census of the factor zoo. *Available at SSRN 3341728*.

Higgins, I. (2021). Generalizing universal function approximators. *Nature Machine Intelligence*.

Huang, J., Lu, H., Lopez Meyer, P., Cordourier, H., & Del Hoyo Ontiveros, J. (2019). Acoustic scene classification using deep learning-based ensemble averaging.

Keras. (2023). Callbacks.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, W., Paffenroth, R. C., & Berthiaume, D. (2021). Neural network ensembles: Theory, training, and the importance of explicit diversity.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Masters, T. (1993). *Practical neural network recipes in c++*. Academic Press Professional, Inc.

McCracken, M. W., & Ng, S. (2016). Fred-md: A monthly database for macroeconomic research. *Journal of Business and Economic Statistics*, *34*(4), 574–589.

Miles, J. (2005). R-squared, adjusted r-squared. *Encyclopedia of statistics in behavioral science*.

Molnar, C. (2023). Interpreting machine learning models with shap. *Lulu. com*.

Moore, G. H. (1967). What is a recession? *The American Statistician*, *21*(4), 16–19.

Moore, R. C., & DeNero, J. (2011). L1 and l2 regularization for multiclass hinge loss models. *Symposium on Machine Learning in Speech and Natural Language Processing*. http://www.ttic.edu/sigml/symposium2011/papers/Moore+DeNero_Regularization.pdf

Mueller, A., & Groble, J. (2023). Scaling the regularization parameter for svcs. *Scikit-Learn*.

Nusrat, I., & Jang, S.-B. (2018). A comparison of regularization techniques in deep neural networks. *Symmetry*, *10*(11), 648.

Pause giant ai experiments: An open letter. (2023). *Future of Life Institute*.

Pohl, W. (2023). What is deep learning? *Lectures on Deep Learning with Application to Finance*.

Robles Herrera, S., Ceberio, M., & Kreinovich, V. (2022). When is deep learning better and when is shallow learning better: Qualitative analysis. *International Journal of Parallel, Emergent and Distributed Systems*, *37*(5), 589–595.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252.

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc.

Sharma, V., Rai, S., & Dev, A. (2012). A comprehensive study of artificial neural networks. *International Journal of Advanced research in computer science and software engineering*, *2*(10).

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, *19*(3), 425–442.

Sola, J., & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, *44*(3), 1464–1468.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need.

Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of physics: Conference series*, *1168*, 022022.

Zhang, S., Liu, M., & Yan, J. (2020). The diversified ensemble neural network. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 16001–16011). Curran Associates, Inc.

Appendices

A Variable Description

A.1 Macroeconomic Data

The column tcode denotes the following data transformation for a series x : (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\log(x_t)$; (5) $\Delta \log(x_t)$; (6) $\Delta^2 \log(x_t)$. (7) $\Delta(x_t/x_{t-1} - 1.0)$. The FRED column gives mnemonics in FRED followed by a short description. The comparable series in Global Insight is given in the column GSI (McCracken & Ng, 2016).

Table A.1: Group 1: Outcome & Income

	id	tcode	fred	description	gsi	gsi:description
1	1	5	RPI	Real Personal Income	$M_1$4386177	PI
2	2	5	W875RX1	Real personal income ex transfer receipts	$M_1$45256755	PI less transfers
3	6	5	INDPRO	IP Index	$M_1$16460980	IP: total
4	7	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies	$M_1$16460981	IP: products
5	8	5	IPFINAL	IP: Final Products (Market Group)	$M_1$16461268	IP: final prod
6	9	5	IPCONGD	IP: Consumer Goods	$M_1$16460982	IP: cons gds
7	10	5	IPDCONGD	IP: Durable Consumer Goods	$M_1$16460983	IP: cons dble
8	11	5	IPNCONGD	IP: Nondurable Consumer Goods	$M_1$16460988	IP: cons nondble
9	12	5	IPBUSEQ	IP: Business Equipment	$M_1$16460995	IP: bus eqpt
10	13	5	IPMAT	IP: Materials	$M_1$16461002	IP: matls
11	14	5	IPDMAT	IP: Durable Materials	$M_1$16461004	IP: dble matls
12	15	5	IPNMAT	IP: Nondurable Materials	$M_1$16461008	IP: nondble matls
13	16	5	IPMANSICS	IP: Manufacturing (SIC)	$M_1$16461013	IP: mfg
14	17	5	IPB51222s	IP: Residential Utilities	$M_1$16461276	IP: res util
15	18	5	IPFUELS	IP: Fuels	$M_1$16461275	IP: fuels
16	19	2	CUMFNS	Capacity Utilization: Manufacturing	$M_1$16461602	Cap util

Table A.2: Group 2: Labor Market

	id	tcode	fred	description	gsi	gsi:description
1	21	2	HWI	Help-Wanted Index for United States		Help wanted indx
2	22	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed	$M_1$10156531	Help wanted/unemp
3	23	5	CLF16OV	Civilian Labor Force	$M_1$10156467	Emp CPS total
4	24	5	CE16OV	Civilian Employment	$M_1$10156498	Emp CPS nonag
5	25	2	UNRATE	Civilian Unemployment Rate	$M_1$10156541	U : all
6	26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)	$M_1$10156528	U : mean duration
7	27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	$M_1$10156527	U < 5wks
8	28	5	UEMP5TO14	Civilians Unemployed for 5 - 14 Weeks	$M_1$10156523	U 5-14 wks
9	29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over	$M_1$10156524	U 15+ wks
10	30	5	UEMP15T26	Civilians Unemployed for 15 − 26 Weeks	$M_1$10156525	U 15 − 26 wks
11	31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	$M_1$10156526	U 27+ wks
12	32	5	CLAIMSx	Initial Claims	$M_1$5186204	UI claims
13	33	5	PAYEMS	All Employees: Total nonfarm	$M_1$23109146	Emp: total
14	34	5	USGOOD	All Employees: Goods-Producing Industries	$M_1$23109172	Emp: gds prod
15	35	5	CES1021000001	All Employees: Mining and Logging: Mining	$M_1$23109244	Emp: mining
16	36	5	USCONS	All Employees: Construction	$M_1$23109331	Emp: const
17	37	5	MANEMP	All Employees: Manufacturing	$M_1$23109542	Emp: mfg
18	38	5	DMANEMP	All Employees: Durable goods	$M_1$23109573	Emp: dble gds
19	39	5	NDMANEMP	All Employees: Nondurable goods	$M_1$23110741	Emp: nondbles
20	40	5	SRVPRD	All Employees: Service-Providing Industries	$M_1$23109193	Emp: services
21	41	5	USTPU	All Employees: Trade, Transportation & Utilities	$M_1$23111543	Emp: TTU
22	42	5	USWTRADE	All Employees: Wholesale Trade	$M_1$23111563	Emp: wholesale
23	43	5	USTRADE	All Employees: Retail Trade	$M_1$23111867	Emp: retail
24	44	5	USFIRE	All Employees: Financial Activities	$M_1$23112777	Emp: FIRE
25	45	5	USGOVT	All Employees: Government	$M_1$23114411	Emp: Govt
26	46	1	CES0600000007	Avg Weekly Hours : Goods-Producing	$M_1$40687274	Avg hrs
27	47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	$M_1$23109554	Overtime: mfg
28	48	1	AWHMAN	Avg Weekly Hours : Manufacturing	$M_1$4386098	Avg hrs: mfg
29	127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing	$M_1$23109182	AHE: goods
30	128	6	CES2000000008	Avg Hourly Earnings : Construction	$M_1$23109341	AHE: const
31	129	6	CES3000000008	Avg Hourly Earnings : Manufacturing	$M_1$23109552	AHE: mfg

Table A.3: Group 3: Consumption & Orders

	id	tcode	fred	description	gsi	gsi:description
1	50	4	HOUST	Housing Starts: Total New Privately Owned	$M_1$10155536	Starts: nonfarm
2	51	4	HOUSTNE	Housing Starts, Northeast	$M_1$10155538	Starts: NE
3	52	4	HOUSTMW	Housing Starts, Midwest	$M_1$10155537	Starts: MW
4	53	4	HOUSTS	Housing Starts, South	$M_1$10155543	Starts: South
5	54	4	HOUSTW	Housing Starts, West	$M_1$10155544	Starts: West
6	55	4	PERMIT	New Private Housing Permits (SAAR)	$M_1$10155532	BP: total
7	56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)	$M_1$10155531	BP: NE
8	57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)	$M_1$10155530	BP: MW
9	58	4	PERMITS	New Private Housing Permits, South (SAAR)	$M_1$10155533	BP: South
10	59	4	PERMITW	New Private Housing Permits, West (SAAR)	$M_1$10155534	BP: West

Table A.4: Group 4: Orders and Inventories

	id	tcode	fred	description	gsi	gsi:description
1	3	5	DPCERA3M086SBEA	Real personal consumption expenditures	$M_1$23008274	Real Consumption
2	5	5	RETAILx	Retail and Food Services Sales	$M_1$30439509	Retail sales
3	64	5	ACOGNO	New Orders for Consumer Goods	$M_1$4385863	Orders: cons gds
4	65	5	AMDMNOx	New Orders for Durable Goods	$M_1$4386110	Orders: dble gds
5	66	5	ANDENOx	New Orders for Nondefense Capital Goods	$M_1$78554409	Orders: cap gds
6	67	5	AMDMUOx	Unfilled Orders for Durable Goods	$M_1$4385946	Unf orders: dble
7	68	5	BUSINVx	Total Business Inventories	$M_1$5192014	M&T invent
8	69	2	ISRATIOx	Total Business: Inventories to Sales Ratio	$M_1$5191529	M&T invent/sales
9	130	2	UMCSENTx	Consumer Sentiment Index	hhsntn	Consumer expect

Table A.5: Group 5: Money & Credit

	id	tcode	fred	description	gsi	gsi:description
1	70	6	M1SL	M1 Money Stock	$M_1$10154984	M1
2	71	6	M2SL	M2 Money Stock	$M_1$10154985	M2
3	72	5	M2REAL	Real M2 Money Stock	$M_1$10154985	M2 (real)
4	73	6	AMBSL	St. Louis Adjusted Monetary Base	$M_1$10154995	MB
5	74	6	TOTRESNS	Total Reserves of Depository Institutions	$M_1$10155011	Reserves tot
6	75	7	NONBORRES	Reserves Of Depository Institutions	$M_1$10155009	Reserves nonbor
7	76	6	BUSLOANS	Commercial and Industrial Loans	BUSLOANS	C&I loan plus
8	77	6	REALLN	Real Estate Loans at All Commercial Banks	BUSLOANS	DC&I loans
9	78	6	NONREVSL	Total Nonrevolving Credit	$M_1$10154564	Cons credit
10	79	2	CONSPI	Nonrevolving consumer credit to Personal Income	$M_1$10154569	Inst cred/PI
11	131	6	MZMSL	MZM Money Stock	N.A.	N.A.
12	132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	N.A.	N.A.
13	133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding	N.A.	N.A.
14	134	6	INVEST	Securities in Bank Credit at All Commercial Banks	N.A.	N.A.

Table A.6: Group 6: Interest Rate & Exchange Rates

	id	tcode	fred	description	gsi	gsi:description
1	84	2	FEDFUNDS	Effective Federal Funds Rate	$M_1$10155157	Fed Funds
2	85∗	2	CP3Mx	3-Month AA Financial Commercial Paper Rate	CPF3M	Comm paper
3	86	2	TB3MS	3-Month Treasury Bill:	$M_1$10155165	3 mo T-bill
4	87	2	TB6MS	6-Month Treasury Bill:	$M_1$10155166	6 mo T-bill
5	88	2	GS1	1-Year Treasury Rate	$M_1$10155168	1yrT-bond
6	89	2	GS5	5-Year Treasury Rate	M-110155174	5 yr T-bond
7	90	2	GS10	10-Year Treasury Rate	$M_1$10155169	10yr T-bond
8	91	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield		Aaa bond
9	92	2	BAA	Moody's Seasoned Baa Corporate Bond Yield		Baa bond
10	93	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS		CP-FF spread
11	94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS		3 mo-FF spread
12	95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS		6 mo-FF spread
13	96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS		1yr − FF spread
14	97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS		5 yr-FF spread
15	98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS		10yr − FF spread
16	99	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS		Aaa-FF spread
17	100	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS		Baa-FF spread
18	101	5	TWEXMMTH	Trade Weighted U.S. Dollar Index: Major Currencies		Ex rate: avg
19	102	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	$M_1$10154768	Ex rate: Switz
20	103	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate	$M_1$10154755	Ex rate: Japan
21	104	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate	M-110154772	Ex rate: UK
22	105	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate	M-110154744	EX rate: Canada

Table A.7: Group 7: Prices

	id	tcode	fred	description	gsi	gsi:description
1	106	6	WPSFD49207	PPI: Finished Goods	M110157517	PPI: fin gds
2	107	6	WPSFD49502	PPI: Finished Consumer Goods	M110157508	PPI: cons gds
3	108	6	WPSID61	PPI: Intermediate Materials	$M_1$10157527	PPI: int matls
4	109	6	WPSID62	PPI: Crude Materials	$M_1$10157500	PPI: crude matls
5	110	6	OILPRICEx	Crude Oil, spliced WTI and Cushing	$M_1$10157273	Spot market price
6	111	6	PPICMM	PPI: Metals and metal products:	$M_1$10157335	PPI: nonferrous
7	112	1	NAPMPRI	ISM Manufacturing: Prices Index	$M_1$10157204	NAPM com price
8	113	6	CPIAUCSL	CPI : All Items	$M_1$10157323	CPI-U: all
9	114	6	CPIAPPSL	CPI : Apparel	$M_1$10157299	CPI-U: apparel
10	115	6	CPITRNSL	CPI : Transportation	$M_1$10157302	CPI-U: transp
11	116	6	CPIMEDSL	CPI : Medical Care	$M_1$10157304	CPI-U: medical
12	117	6	CUSR0000SAC	CPI : Commodities	$M_1$10157314	CPI-U: comm.
13	118	6	CUSR0000SAD	CPI : Durables	$M_1$10157315	CPI-U: dbles
14	119	6	CUSR0000SAS	CPI : Services	$M_1$10157325	CPI-U: services
15	120	6	CPIULFSL	CPI : All Items Less Food	$M_1$10157328	CPI-U: ex food
16	121	6	CUSR0000SA0L2	CPI : All items less shelter	$M_1$10157329	CPI-U: ex shelter
17	122	6	CUSR0000SA0L5	CPI : All items less medical care	$M_1$10157330	CPI-U: ex med
18	123	6	PCEPI	Personal Cons. Expend: Chain Index	gmdc	PCE defl
19	124	6	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	gmdcd	PCE defl: dlbes
20	125	6	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	gmdcn	PCE defl: nondble
21	126	6	DSERRG3M086SBEA	Personal Cons. Exp: Services	gmdcs	PCE defl: service

Table A.8: Group 8: Stock Market

	id	tcode	fred	description	gsi	gsi:description
1	80	5	S&P 500	S&P's Common Stock Price Index: Composite	$M_1$10155044	S&P 500
2	8	5	S&P: indust	S&P's Common Stock Price Index: Industrials	$M_1$10155047	S&P: indust
3	82	2	S&P div yield	S&P's Composite Common Stock: Dividend Yield		S&P div yield
4	83	5	S&P PE ratio	S&P's Composite Common Stock: Price-Earnings Ratio		S&P PE ratio
5	135	1	VXOCLSx	VXO		

A.2 Financial Data

Table A.9: Stock Characteristics (Gu et al., 2020)

No.	Acronym	Firm characteristic	Paper's author(s)	Year, Journal	Data Source	Frequency
1	absacc	Absolute accruals	Bandyopadhyay, Huang & Wirjanto	2010, WP	Compustat	Annual
2	acc	Working capital accruals	Sloan	1996, TAR	Compustat	Annual
3	aeavol	Abnormal earnings announcement volume	Lerman, Livnat & Mendenhall	2007, WP	Compustat+CRSP	Quarterly
4	age	# years since first Compustat coverage	Jiang, Lee & Zhang	2005, RAS	Compustat	Annual
5	agr	Asset growth	Cooper, Gulen & Schill	2008, JF	Compustat	Annual
6	baspread	Bid-ask spread	Amihud & Mendelson	1989, JF	CRSP	Monthly
7	beta	Beta	Fama & MacBeth	1973, JPE	CRSP	Monthly
8	betasq	Beta squared	Fama & MacBeth	1973, JPE	CRSP	Monthly
9	bm	Book-to-market	Rosenberg, Reid & Lanstein	1985, JPM	Compustat+CRSP	Annual
10	bm$_i a$	Industry-adjusted book to market	Asness, Porter & Stevens	2000, WP	Compustat+CRSP	Annual
11	cash	Cash holdings	Palazzo	2012, JFE	Compustat	Quarterly
12	cashdebt	Cash flow to debt	Ou & Penman	1989, JAE	Compustat	Annual
13	cashpr	Cash productivity	Chandrashekar & Rao	2009, WP	Compustat	Annual
14	cfp	Cash flow to price ratio	Desai, Rajgopal & Venkatachalam	2004, TAR	Compustat	Annual
15	cfpia	Industry-adjusted cash flow to price ratio	Asness, Porter & Stevens	2000, WP	Compustat	Annual
16	chatoia	Industry-adjusted change in asset turnover	Soliman	2008, TAR	Compustat	Annual
17	chesho	Change in shares outstanding	Pontiff & Woodgate	2008, JF	Compustat	Annual
18	chempia	Industry-adjusted change in employees	Asness, Porter & Stevens	1994, WP	Compustat	Annual
19	chinv	Change in inventory	Thomas & Zhang	2002, RAS	Compustat	Annual
20	chmom	Change in 6-month momentum	Gettleman & Marks	2006, WP	CRSP	Monthly
21	chpmia	Industry-adjusted change in profit margin	Soliman	2008, TAR	Compustat	Annual
22	chtx	Change in tax expense	Thomas & Zhang	2011, JAR	Compustat	Quarterly
23	cinvest	Corporate investment	Titman, Wei & Xie	2004, JFQA	Compustat	Quarterly
24	convind	Convertible debt indicator	Valta	2016, JFQA	Compustat	Annual
25	currat	Current ratio	Ou & Penman	1989, JAE	Compustat	Annual
26	depr	Depreciation / PP&E	Holthausen & Larcker	1992, JAE	Compustat	Annual
27	divi	Dividend initiation	Michaely, Thaler & Womack	1995, JF	Compustat	Annual
28	divo	Dividend omission	Michaely, Thaler & Womack	1995, JF	Compustat	Annual
29	dolvol	Dollar trading volume	Chordia, Subrahmanyam & Anshuman	2001, JFE	CRSP	Monthly
30	dy	Dividend to price	Litzenberger & Ramaswamy	1982, JF	Compustat	Annual
31	ear	Earnings announcement return	Kishore, Brandt, Santa-Clara & Venkatachalam	2008, WP	Compustat+CRSP	Quarterly

Table A.10: Stock Characteristics continued (Gu et al., 2020)

No.	Acronym	Firm characteristic	Paper's author(s)	Year, Journal	Data Source	Frequency
32	egr	Growth in common shareholder equity	Richardson, Sloan, Soliman	2005, JAE	Compustat	Annual
33	ep	Earnings to price	Basu	1977, JF	Compustat	Annual
34	gma	Gross profitability	Novy-Marx	2013, JFE	Compustat	Annual
35	grCAPX	Growth in capital expenditures	Anderson & Garcia-Feijoo	2006, JF	Compustat	Annual
36	grltnoa	Growth in long-term net operating assets	Fairfield, Whisenant & Yohn	2003, TAR	Compustat	Annual
37	herf	Industry sales concentration	Hou & Robinson	2006, JF	Compustat	Annual
38	hire	Employee growth rate	Bazdresch, Belo & Lin	2014, JPE	Compustat	Annual
39	idiovol	Idiosyncratic return volatility	Ali, Hwang & Trombley	2003, JFE	CRSP	Monthly
40	ill	Illiquidity	Amihud	2002, JFM	CRSP	Monthly
42	invest	Capital expenditures and inventory	Chen & Zhang	2010, JF	Compustat	Annual
43	lev	Leverage	Bhandari	1988, JF	Compustat	Annual
44	lgr	Growth in long-term debt	Richardson, Sloan, Soliman & Tuna	2005, JAE	Compustat	Annual
45	maxret	Maximum daily return	Bali, Cakici & Whitelaw	2011, JFE	CRSP	Monthly
46	mom 12 m	12-month momentum	Jegadeesh	1990, JF	CRSP	Monthly
47	momlm	1-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
48	mom 36 m	36-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
49	mom6m	6-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
50	ms	Financial statement score	Mohanram	2005, RAS	Compustat	Quarterly
52	mve$_i a$	Industry-adjusted size	Asness, Porter & Stevens	2000, WP	Compustat	Annual
53	nincr	Number of earnings increases	Barth, Elliott & Finn	1999, JAR	Compustat	Quarterly
54	operprof	Operating profitability	Fama & French	2015, JFE	Compustat	Annual
55	orgcap	Organizational capital	Eisfeldt & Papanikolaou	2013, JF	Compustat	Annual
56	pchcapx$_i a$	Industry-adjusted % change in capital expenditures	Abarbanell & Bushee	1998, TAR	Compustat	Annual
57	pchcurrat	% change in current ratio	Ou & Penman	1989, JAE	Compustat	Annual
58	pchdepr	% change in depreciation	Holthausen & Larcker	1992, JAE	Compustat	Annual
59	pchgm$_p$*chsale*	% change in gross margin - % change in sales	Abarbanell & Bushee	1998, TAR	Compustat	Annual
60	pchquick	% change in quick ratio	Ou & Penman	1989, JAE	Compustat	Annual
61	pchsale$_p$*chinvt*	% change in sales - % change in inventory	Abarbanell & Bushee	1998, TAR	Compustat	Annual
62	pchsale$_p$*chrect*	% change in sales - % change in A/R	Abarbanell & Bushee	1998, TAR	Compustat	Annual
63	pchsale$_p$*chxsga*	% change in sales - % change in SG&A	Abarbanell & Bushee	1998, TAR	Compustat	Annual
64	pchsaleinv	% change sales-to-inventory	Ou & Penman	1989, JAE	Compustat	Annual
65	petace	Percent accruals	Hafzalla, Lundholm & Van Winkle	2011, TAR	Compustat	Annual
66	pricedelay	Price delay	Hou & Moskowitz	2005, RFS	CRSP	Monthly
67	ps	Financial statements score	Piotroski	2000, JAR	Compustat	Annual
68	quick	Quick ratio	Ou & Penman	1989, JAE	Compustat	Annual
69	rd	R&D increase	Eberhart, Maxwell & Siddique	2004, JF	Compustat	Annual
70	rd$_m ve$	R&D to market capitalization	Guo, Lev & Shi	2006, JBFA	Compustat	Annual
71	rd$_s ale$	R&D to sales	Guo, Lev & Shi	2006, JBFA	Compustat	Annual
72	realestate	Real estate holdings	Tuzel	2010, RFS	Compustat	Annual
73	retvol	Return volatility	Ang, Hodrick, Xing & Zhang	2006, JF	CRSP	Monthly
74	roaq	Return on assets	Balakrishnan, Bartov & Faurel	2010, JAE	Compustat	Quarterly
75	roavol	Earnings volatility	Francis, LaFond, Olsson & Schipper	2004, TAR	Compustat	Quarterly
76	roeq	Return on equity	Hou, Xue & Zhang	2015, RFS	Compustat	Quarterly
77	roic	Return on invested capital	Brown & Rowe	2007, WP	Compustat	Annual
78	rsup	Revenue surprise	Kama	2009, JBFA	Compustat	Quarterly
79	salecash	Sales to cash	Ou & Penman	1989, JAE	Compustat	Annual
80	saleinv	Sales to inventory	Ou & Penman	1989, JAE	Compustat	Annual
81	salerec	Sales to receivables	Ou & Penman	1989, JAE	Compustat	Annual
82	secured	Secured debt	Valta	2016, JFQA	Compustat	Annual
83	securedind	Secured debt indicator	Valta	2016, JFQA	Compustat	Annual
84	sgr	Sales growth	Lakonishok, Shleifer & Vishny	1994, JF	Compustat	Annual
85	sin	Sin stocks	Hong & Kacperczyk	2009, JFE	Compustat	Annual
86	sp	Sales to price	Barbee, Mukherji, & Raines	1996, FAJ	Compustat	Annual
87	std$_d olvol$	Volatility of liquidity (dollar trading volume)	Chordia, Subrahmanyam & Anshuman	2001, JFE	CRSP	Monthly
88	std$_t urn$	Volatility of liquidity (share turnover)	Chordia, Subrahmanyam & Anshuman	2001, JFE	CRSP	Monthly
89	stdace	Accrual volatility	Bandyopadhyay, Huang & Wirjanto	2010, WP	Compustat	Quarterly
90	stdef	Cash flow volatility	Huang	2009, JEF	Compustat	Quarterly
91	tang	Debt capacity/firm tangibility	Almeida & Campello	2007, RFS	Compustat	Annual
92	tb	Tax income to book income	Lev & Nissim	2004, TAR	Compustat	Annual
93	turn	Share turnover	Datar, Naik & Radcliffe	1998, JFM	CRSP	Monthly
94	zerotrade	Zero trading days	Liu	2006, JFE	CRSP	Monthly

B Shapley Values Distributions and Variable Importance

B.1 Expansions

Figure B.1: NN1 Model

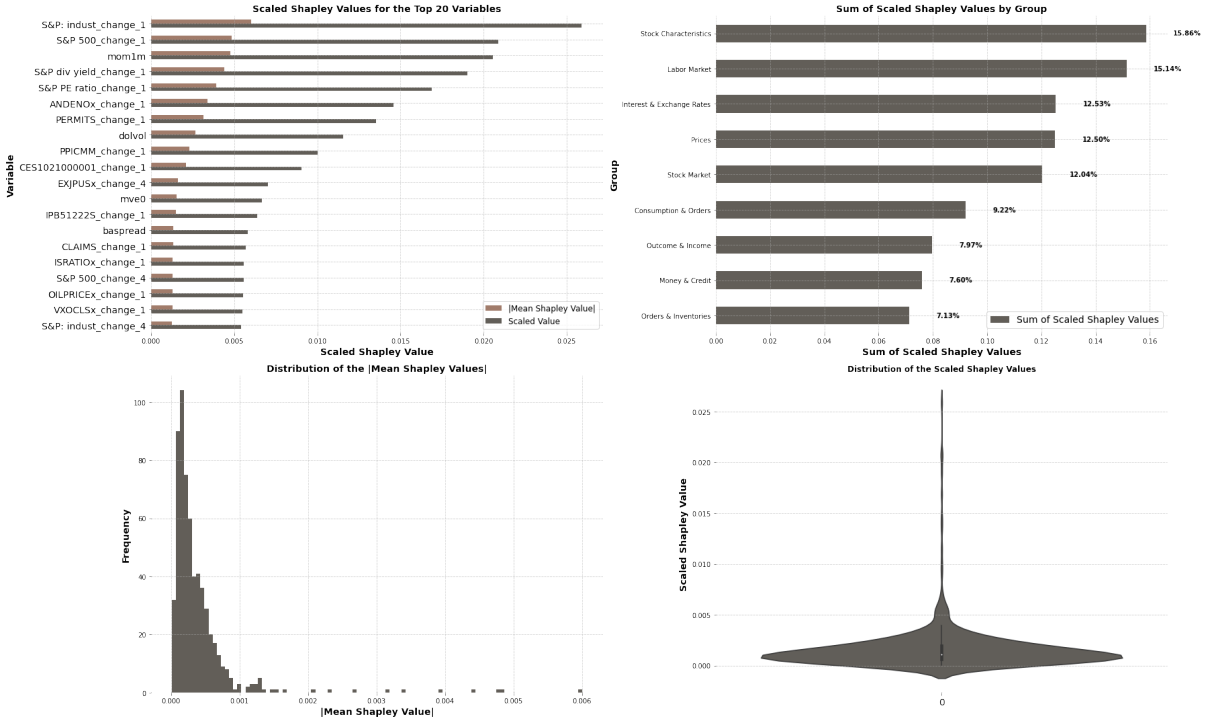


Figure B.2: NN2 Model

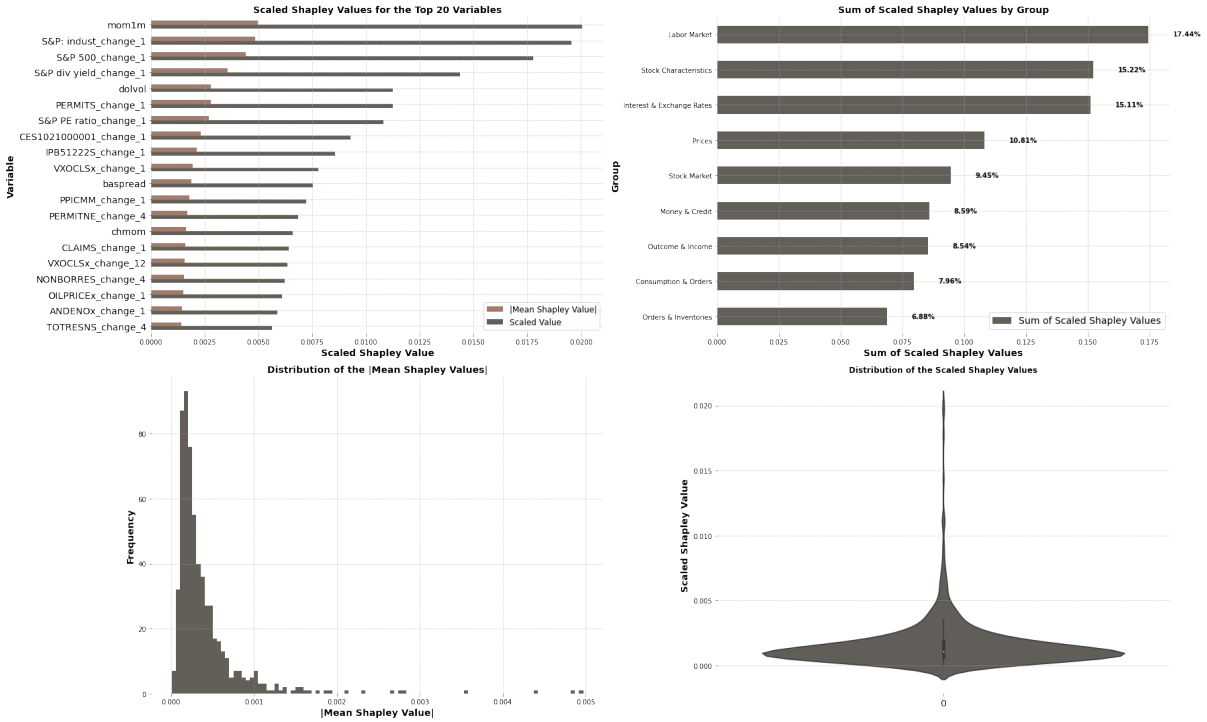


Figure B.3: NN3 Model

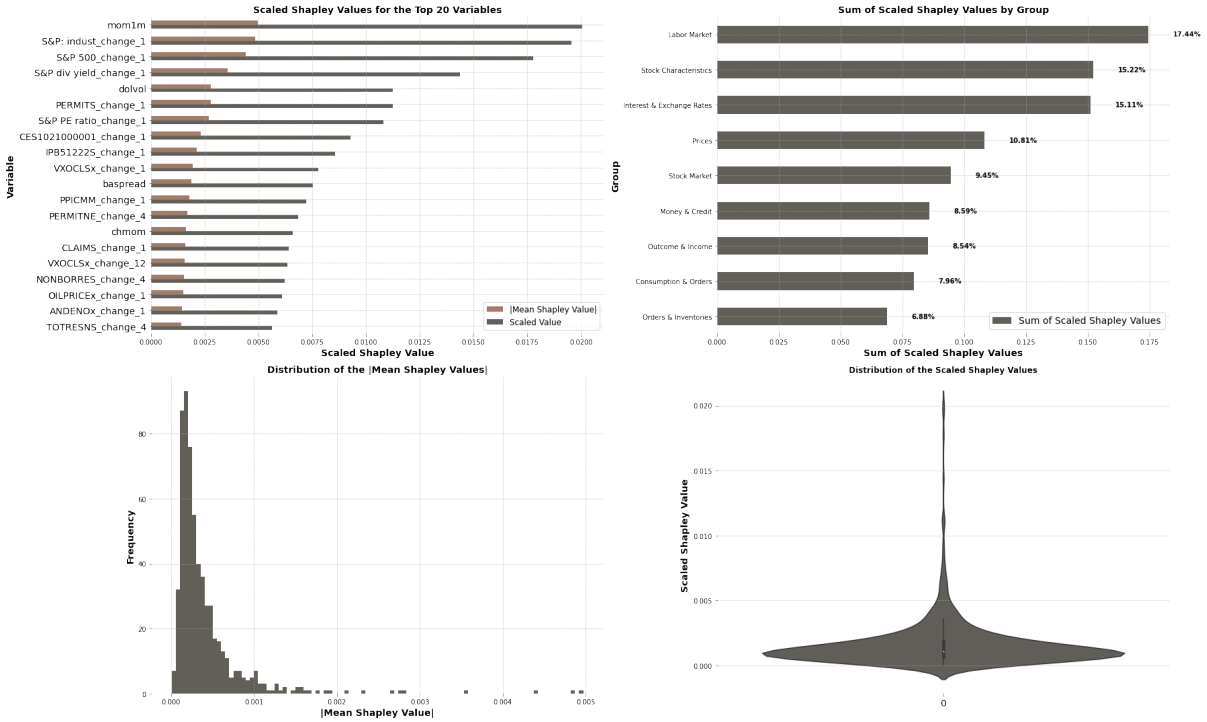


Figure B.4: NN4 Model

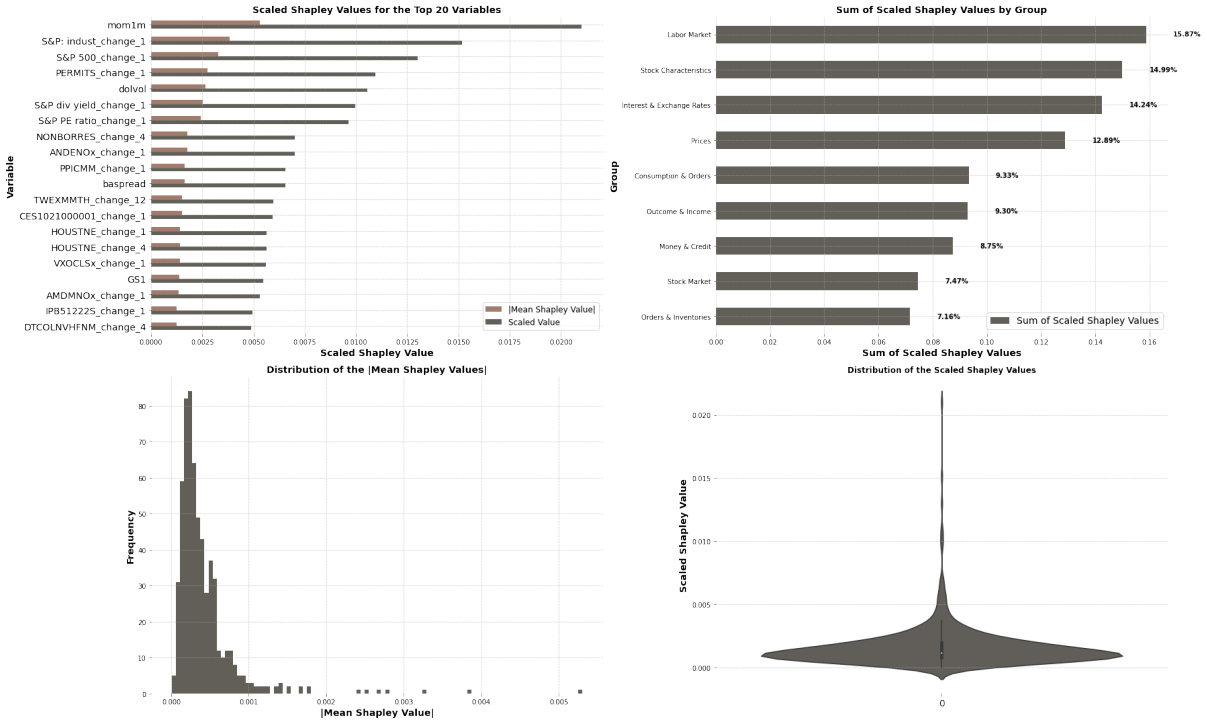
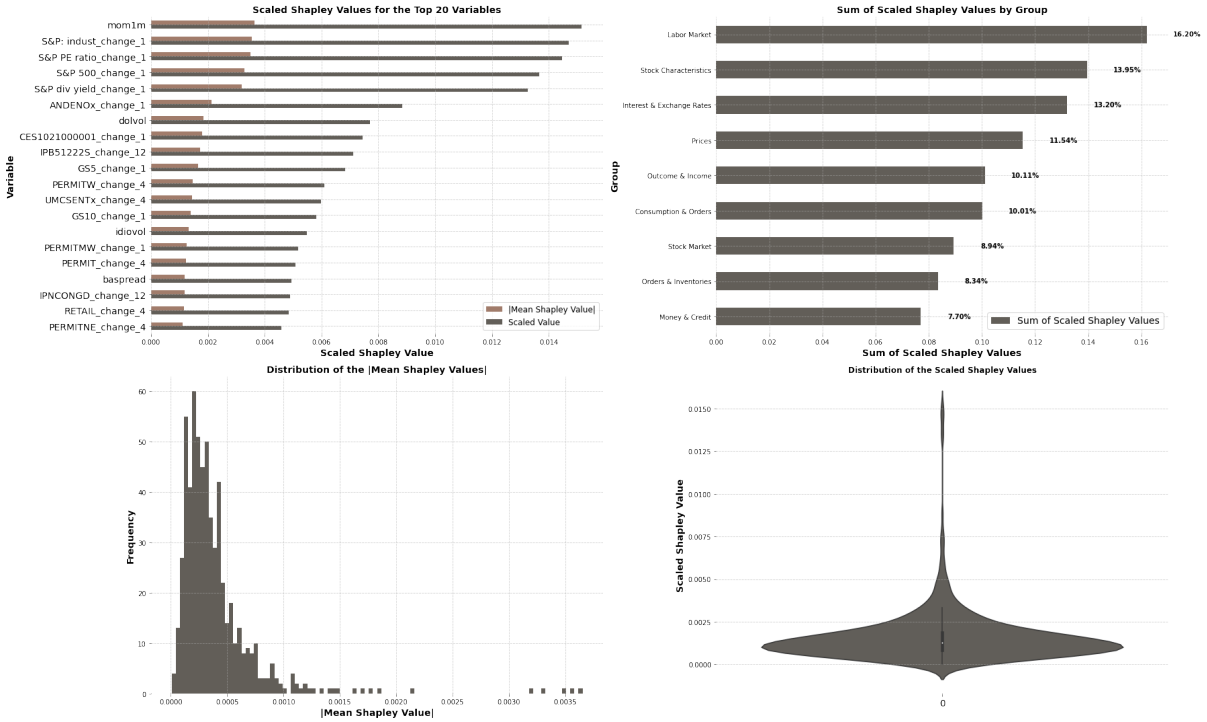


Figure B.5: NN5 Model



B.2 Recessions

Figure B.6: NN1 Model

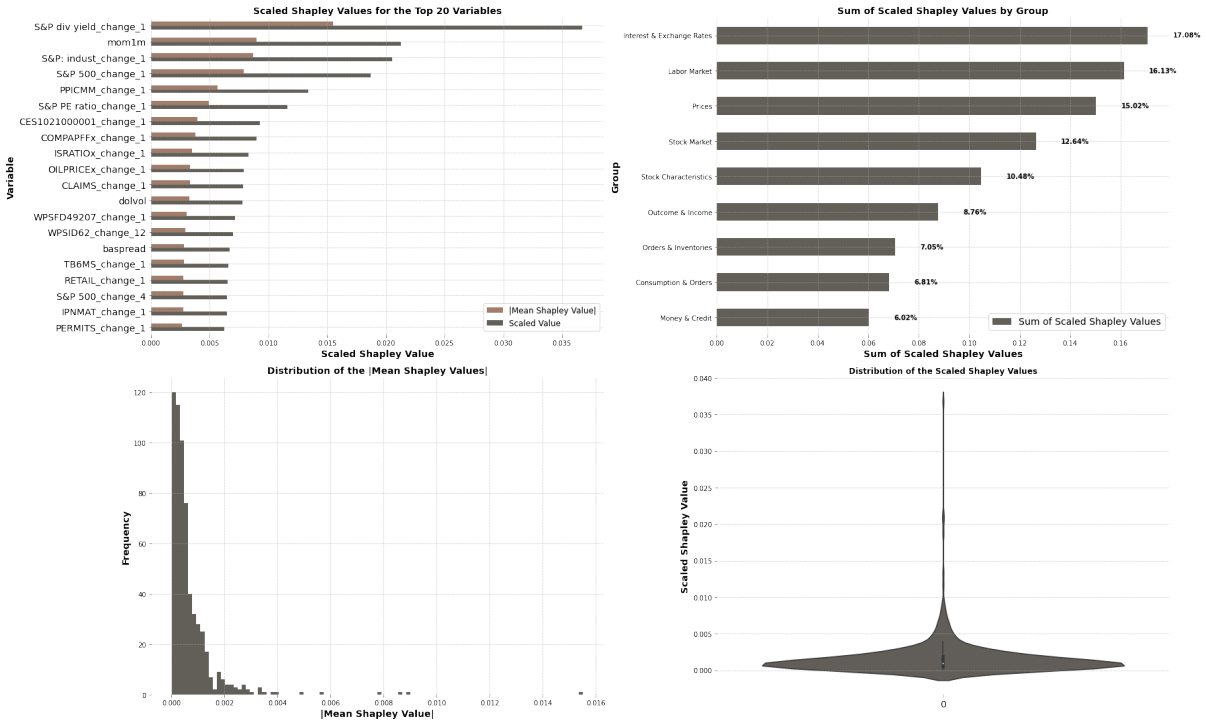


Figure B.7: NN2 Model

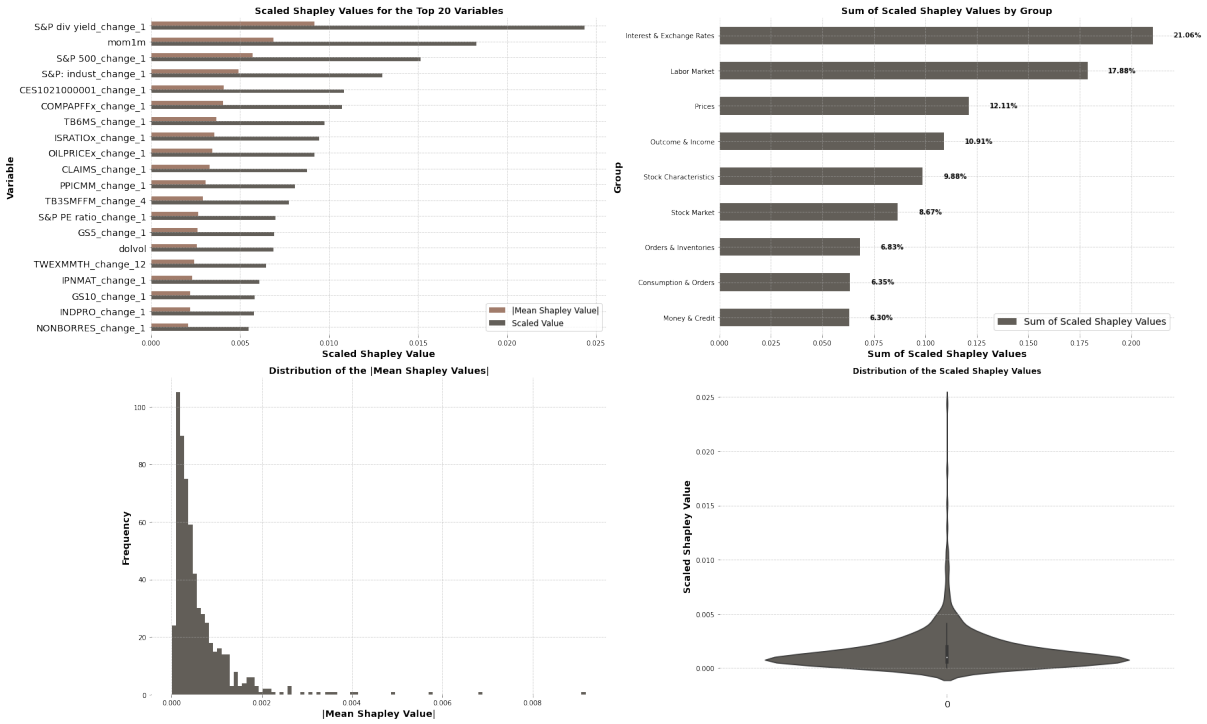


Figure B.8: NN3 Model

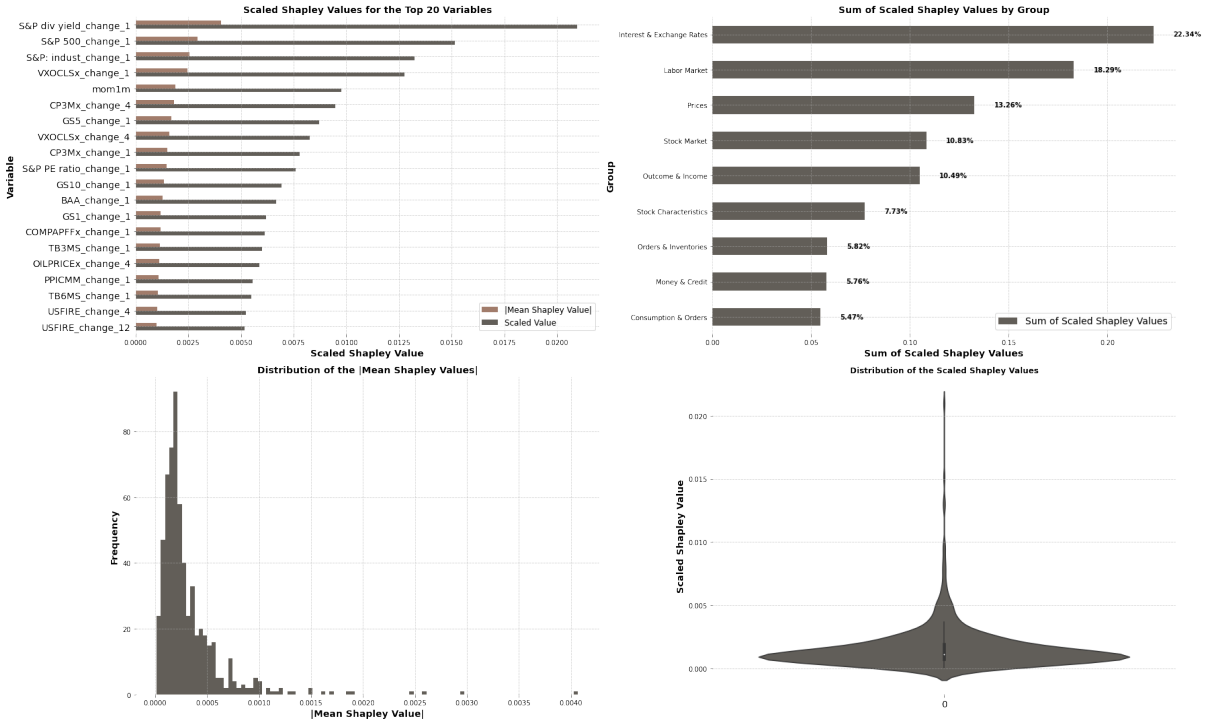


Figure B.9: NN4 Model

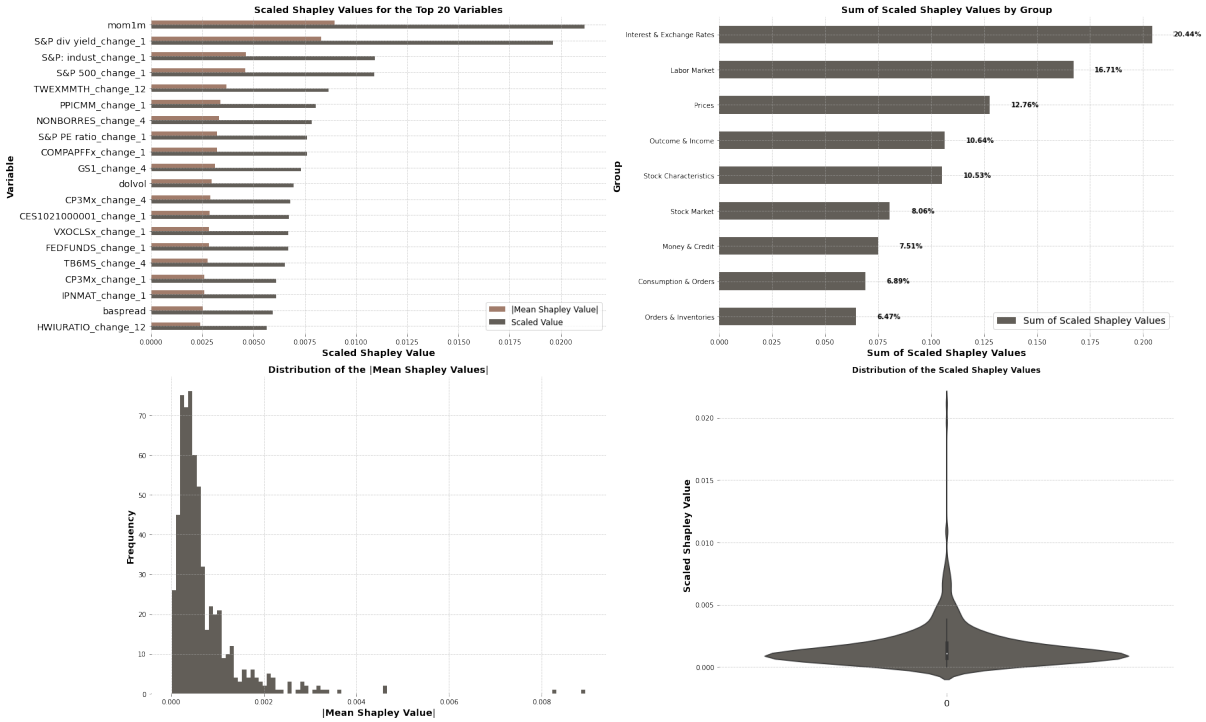


Figure B.10: NN5 Model