Overfitting: A Foe to Many But a Friend to Overconfidence in the Stock Market

Margaret Beisenbek
mbeisenbek2002@gmail.com
Bentley University
Waltham, MA, USA

ABSTRACT

The financial market defies the assumption that investors are rational. As a result, when predicting asset returns, ML models must capture psychological biases, such as overconfidence. Camerer [5] proposes that a "good" ML model loses the capability of accurately capturing overconfidence by correcting for overfitting. This study proves this hypothesis using ML, which relies on a high confidence interval and low accuracy rate as evidence of overconfidence. Thus, this study helps defy the traditional prejudice of overfitting: befriending overfitting when predicting asset returns.

CCS CONCEPTS

• **Machine Learning (ML) → Model overfitting, Regularization techniques (Lasso, Enet, Ridge)**; • **Statistics → Confidence intervals, Out-of-sample accuracy rate**; • **Human-Computer Interaction (HCI) → User overconfidence in machine learning models**.

KEYWORDS

Machine Learning, Time Series Analysis, Behavioral Economics; Overconfidence; Overfitting; Return Prediction

ACM Reference Format:

Margaret Beisenbek. 2024. Overfitting: A Foe to Many But a Friend to Overconfidence in the Stock Market. In *Proceedings of KDD Undergraduate Consortium (KDD-UC).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Financial markets show irrational investor behavior [1, 13]. An example of irrational behavior in the financial market is the active investing puzzle: investors, despite their active trading and decades of experience, underperform the passive strategy on average, based on net costs [7]. As investors trade more, they experience greater losses [20].

This inefficiency stems from psychological biases like overconfidence where investors experiencing high returns become overconfident, attributing their success to their skill. Conversely, when investors experience low returns, they usually blame bad luck rather than their shortcomings, and their confidence does not decrease [10]. Although other biases are present, overconfidence "seems likely to be a key factor in financial decision-making" [7].

Overconfidence is a fundamental bias in behavioral finance models. A vast amount of research finds that overconfidence impacts the stock market in terms of returns, volatility, and other factors [3, 11, 14, 20]. Two key features summarize these impacts: "high trading volumes and predictable returns" [7]. Rational expectations cannot explain these features. The first feature, "high trading volumes," directly relates to the active investing puzzle [7].

Existing literature utilizes machine learning (ML) to model overconfidence. Some of the most common models include regression [14], impulse-response functions (IRFs) [2, 11, 22], or a direct measure to extract a measure of overconfidence to reflect overconfidence [15]. Accurately measuring overconfidence is crucial because it can lead to better stock market predictions.

Despite these methodologies, none consider using the overfitting of ML models. Camerer [5] proposes that the divergence between the accuracy of the training set and the testing set can mimic human overconfidence. To my best knowledge, no study has implemented or tested this hypothesis. Without such a consideration, developing the best methodology for considering overconfidence in the stock market remains an unaccomplished goal.

Such a consideration, however, will counter the common perception of overfitting. Existing literature strives to prevent overfitting [4, 17, 19, 25, 26]. Nevertheless, in the case of overconfidence, based on Camerer [5], overfitting may be beneficial and even increase the predictive power of ML models for stock market prediction. Specifically, Camerer [5] expressed his desire to conduct the following research project: compare human predictions with the ones generated by an ML algorithm and ones generated by a 'bad' ML model that overfits [5]. The predictions from the 'bad' model can have elements of human overconfidence.

This project strives to minimize the costs of the active investing puzzle by helping better represent overconfidence in the stock market. This purpose will test the proposition of Camerer [5] and, by extension, challenge the common notion in ML that overfitting is an unwanted quality that prevents models from realistically modeling real-life issues. To accomplish such a goal, this study will answer the following question: to what extent can overfitting represent overconfidence in the stock market? To accomplish such a goal, and since no study executed the recommendation made by Camerer [5], this study will use stock market return data to test Camerer's hypothesis in the stock market.

KDD Undergraduate Consortium (KDD-UC), August 25-29, 2024, Barcelona, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 PREVIOUS RESEARCH

2.1 Overconfidence as a Bias

As defined by Daniel et al. [8], overconfidence among investors is them overlooking public information and overweighing their private information. This psychological bias is important because it influences trading activity. Thus, if increased information does not necessarily lead to better decision-making, then what factors might influence an investor's choices?

Oskamp [21] found that as participants - psychologists, undergraduate students, and graduate students - received more information about a psychological case, the accuracy of their assessment did not improve [21]. Specifically, in the first stage, where they read three sentences of content about a person, the average accuracy rate was 26 percent (meaning, the participants got 26% of the questions right). In the next three stages, where the participants read 5 pages worth of content, their average accuracy rate did not improve despite being exposed to more content: their average accuracy rate was 28 percent (the participants got 28% correct). In contrast, their confidence level increased by about 20% from the first stage to the next three stages. In other words, the participants think their accuracy increased after consuming more content: in the first stage, the participants believed they had 33% correct; meanwhile, in the last stage, participants thought they had 53% correct when, in reality, their average accuracy rate increased by only 2 percent.

Camerer [5] recommends the application of overfitting using Oskamp's [21] evidence of the miscalibration element of overconfidence. Specifically, Camerer [5] says the following:

> This increase in confidence, combined with no increase in accuracy, is reminiscent of the difference between training and testing set accuracy in AI. As more and more variables are included in a training set, the (unpenalized) accuracy will always increase. As a result of overfitting, however, testing set accuracy will decline when too many variables are included. The resulting gap between training and testing set accuracy will grow, much as the overconfidence in Oskamp's subjects grew with the equivalent of more "variables" (i.e., more material on the single person they were judging).

The relatively constant state of accuracy and increase in confidence present in the experiment by Oskamp [21] may be equivalent to adding variables to the training set, hence increasing the accuracy rate of the training set but decreasing it for the testing set. The accuracy rate for the training and testing set will diverge [5] similar to the overconfidence level diverging from the first stage and the next three stages in the experiment of Oskamp [21].

2.2 Measuring Overconfidence

Such a recommendation has great potential in estimating overconfidence in the market. An analogy of the recommendation of Camerer [5] is adding more variables in an ML algorithm that predicts the stock market returns. Based on the suggestion of Camerer [5], when adding variables, the accuracy rate of the training and testing sets will further diverge. This divergence could be a proxy for overconfidence. Currently, no study seems to verify this claim with empirical evidence.

Instead, studies rely on survey responses from which they applied statistical techniques to decipher a measure of overconfidence [11]. Another popular method relies on nonlinear impulse-response functions with a vector autoregression (VAR) [2, 22]. Coşkun, Kahyaoglu, and Lau [6] is possibly the only study that employs AI techniques and nonlinear considerations when launching impulse responses to consider overconfidence in the stock market of Borsa Istanbul.

2.3 Overfitting

In existing ML literature, overfitting is an unwanted quality. Using the bias-variance trade-off, if H is the capacity of the function class, then, ideally, an ML model would achieve a balance of underfitting and overfitting:

(1) In the case of underfitting, in a too-small H, the predictors of H will most likely underfit the training data, yielding low predictive power.
(2) On the other hand, for overfitting, in a too large H, the empirical risk minimizer of H will most likely over the training data, yielding a low accuracy [4].

The textbook definition of overfitting is the following: a "model with zero training error is overfitted to the training data and will typically generalize poorly" [17]. Due to the problematic nature of overfitting, vast literature strives to minimize it [4, 19, 25, 26]. At best, overfitting can be seen as "benign" [24]. However, no existing research uses overfitting to yield beneficial results, including overconfidence for predicting.

2.4 Significance of this Study

ML algorithms attempt to factor overconfidence when predicting stock market behavior. To accomplish such a function, existing literature primarily used impulse response functions (IRFs), regression, or direct data of overconfidence [2, 14, 15, 22]. Despite such an application of ML, currently, no study considers overfitting as a strategy to reflect overconfidence, as recommended by Cramer [5]. The presented research project aims to fill this gap in the literature. By using overfitting to reflect overconfidence in the stock market, this project will strive to improve upon the current overconfidence models. That way, the costs of the active investing puzzle can be better understood and avoided.

This project will also challenge the traditional notion of overfitting in ML. Overall, in the current literature, overfitting is an undesirable quality in an ML algorithm [4, 9, 25, 26]. By challenging this notion, this project may encourage future research to use overfitting to measure overconfidence.

3 METHODOLOGY

The ML models used in this paper are the ones in Gu et al. [16] listed in Figure 1. Gu et al. [16] determine the effectiveness of each model by minimizing mean squared prediction error (MSE). Overall, each model aims to asset's excess return using the formula:

$$r_{i,t+1} = E_t(r_{i,t+1}) + \epsilon_{i,t+1} \quad (1)$$

where each stock has an index of $i = 1, \ldots, N_t$ and months, of $t = 1, \ldots, T$. Based on these indices, the asset's excess return is denoted as $r_{i,t+1}$. The conditional expected return $E_t(r_{i,t+1})$ and the prediction error $\epsilon_{i,t+1}$ are the components of the asset's excess return. Based on these terms, the conditional expected return is the following:

$$E_t(r_{i,t+1}) = g^*(z_{i,t}) \tag{2}$$

where $g^*(\cdot)$ depends on predictor variables $z_{i,t}$ only, not the past information before t or individual stocks other than the ith. This way, the model has stability when estimating risk premiums for assets while leveraging information for the entire panel of stocks.

3.1 Monte Carlo Simulation

3.1.1 Set the Models. This study simulates data by creating four models based on two dimensions: different models (linear model called "Model A" and non-linear one called "Model B") and different characteristic dimensions (50 and 100 characteristics)[1]. The simulation from Gu et al. [16] is based on the following (latent) 3-factor model for excess returns $r_{i,t+1}$, for $t = 1, 2, \ldots, T$:

$$r_{i,t+1} = g^*(z_{i,t}) + e_{i,t+1}$$

with

$$e_{i,t+1} = \beta_{i,t} v_{t+1} + \epsilon_{i,t+1}, \tag{3}$$

$$z_{i,t} = (1, x_t)' \otimes c_{i,t},$$

$$\beta_{i,t} = (c_{i1,t}, c_{i2,t}, c_{i3,t})$$

where c_t is an $N \times P_c$ matrix of characteristics, v_{t+1} is a 3×1 vector of factors, x_t is a univariate time series, and ϵ_{t+1} is an $N \times 1$ vector of idiosyncratic errors [16]. Excess returns ($r_{i,t+1}$) are explained by three unknown factors (v_{t+1}) and other observable characteristics ($z_{i,t}$). The dependent variable, excess returns ($r_{i,t+1}$), for asset i at time $t+1$ has two components: observable features and unobservable features [16].

The observable features are represented by $g^*(z_{i,t})$ to capture how specific characteristics help drive returns. These specific characteristics are determined by the Knocker product of the overall market trend x_t and the characteristics $c_{i,t}$ of asset i at time t.

The unpredictable explanations for the excess returns are captured by the error term $e_{i,t+1}$. This term is composed of the component explaining how the three factors (v_{t+1}) impact the return of asset i, which is denoted by the weights $\beta_{i,t}$. The three factors v_{t+1} have a normal distribution with a specific standard deviation of 0.052 ($v_{t+1} \sim N(0, 0.05^2 \times I_3)$). The second component of the error term is the idiosyncratic term $\epsilon_{i,t+1}$, which follows a t-distribution with a specific standard deviation of 0.052 ($\epsilon_{i,t+1} \sim t_5(0, 0.05^2)$). The distributions of the three factors (v_{t+1}) and idiosyncratic term ($\epsilon_{i,t+1}$) are calibrated by Gu et al. [16] such that the "average time series R^2 is 40% and the average annualized volatility is 30%."

One of the elements in equation (3) is a simulated dataset of characteristics ($c_{ij,t}$) for each asset i and time t, where j is a characteristic. The simulation follows the formula:

$$c_{ij,t} = \frac{2}{N+1}\text{CSrank}(c_{ij,t-1}) - 1, \qquad c_{ij,t-1} = \rho_j c_{ij,t-1} + \epsilon_{ij,t} \tag{4}$$

with $\rho_j \sim U[0.9, 1]$ and $\epsilon_{ij,t} \sim N(0, 1-\rho_j^2)$. To calculate the simulated characteristic value for asset i, characteristic j, and time period t, the function above first takes the value of the same characteristic for the asset but for the previous time period $t-1$ to capture persistence because financial characteristics tend to perform similarly for an asset over time. Then, this previous value is multiplied by factor ρ_j, which represents the magnitude of the impact of the past value on the current value. A high ρ_j value (when ρ_j is closer to 1) indicates that the characteristic shows greater persistence, meaning that the current value is more likely to be similar to the past value. The CSrank function or the Cross-Section rank function takes the current value of the characteristic and ranks it across all assets for the same time period t, converting this ranking into a value from 0 to 1. Further normalization of the value includes multiplying it with a constant term $\frac{2}{N+1}$ to scale the value between -1 and 1, and then subtracting the resulting value from 1. This normalization is helpful because it sets a standardized scale, making it easier for comparison and implementation of ML models since such models prefer features on a similar scale.

Another necessary component of equation (3) is x_t. It is a time series simulated using the formula:

$$x_t = \rho x_{t-1} + u_t \tag{5}$$

with $u_t \sim N(0, 1-\rho^2)$ and $\rho = 0.95$. The purpose of this equation is to reflect the high persistence of x_t.

The last important component of equation (3) is the $g^*(\cdot)$ functions:

$$g^*(c_{i1,t}, c_{i2,t}, c_{i3,t} \times x_t)\theta_0 \tag{6}$$

$$\text{where} \quad \theta_0 = \begin{pmatrix} 0.02 \\ 0.02 \\ 0.02 \end{pmatrix}$$

$$g^*(c_{i1,t}^2, c_{i1,t} \times c_{i2,t}, \text{sgn}(c_{i3,t} \times x_t))\theta_0 \tag{7}$$

$$\text{where} \quad \theta_0 = \begin{pmatrix} 0.04 \\ 0.03 \\ 0.012 \end{pmatrix}$$

Both versions include 3 covariates to predict excess returns. The input of the functions is $z_{i,t}$, which is a vector of features that help predict excess returns. "Model A" or case (a) sets a linear combination of the three covariates in $z_{i,t}$. "Model B" or case (b), on the other hand, includes non-linear relationships with $c_{i1,t}^2$. "Model B" also considers interaction effects with $c_{i1,t} \times c_{i2,t}$. The element $\text{sgn}(c_{i3,t} \times x_t)$ shows the relationship between the characteristic $c_{i3,t}$ and the time series variable x_t. If both elements have the same sign, then the model might predict a stronger excess return compared to the alternative scenario of both components having the same inverse sign. This element helps capture the interactive effects of the characteristic and time variable on the excess return. In essence, "Model B" or case (b) includes the non-linear and interaction effects.

For both cases, θ_0 is calibrated to achieve a cross-sectional R^2 of 50% and a predictive R^2 of 5%. Meaning, that across all assets for

[1]The reproduction of the data simulation from Gu et al [16] can be found in https://github.com/xiubooth/ML_Codes

a given time period, the R^2 is 50% (the model can predict 50% of the variation in returns across assets for time period t). The second meaningful interpretation is that the model can predict 5% of the variance in future excess returns.

3.1.2 Set the Simulation Parameters. Before running the Monte Carlo simulations, Gu et al. ([16] set the following parameters:

- Number of Assets (N=200) where the model predicts the excess returns for 200 assets.
- Time Series Length (T=180) where the model predicts the excess returns for 180 time periods.
- Total Characteristics ($P_c = 50$ or $P_c = 100$) where the model predicts the excess returns using fixed exogenous characteristics P_x and latent factors.
- 100 number of simulations.

3.1.3 Simulation Loop. With each of the 100 simulation loops, equation (4) generates random values for a panel of characteristics $c_{ij,t}$. Then, equation (5) generates data for x_t. Lastly, using simulated data for excess return drivers v_{t+1}, Gu et al. [16] combine the simulated characteristics data ($c_{ij,t}$), time series (x_t), and excess return drivers (v_{t+1}) into a complete dataset.

For each simulation, that dataset is split into three groups: training, validation, and testing. The training group helps train the respective ML model on the data for the model to learn the relationship between the features $z_{i,t}$ and excess returns $r_{i,t+1}$. Then, the validation set helps tune the hyperparameters of the model. Lastly, to evaluate the performance, the ML model is implemented on the testing data (new data that the ML model has not seen).

This entire process repeats for each loop in the 100 Monte Carlo simulations set by Gu et al. [16] for each Model ("Model A" and "Model B") and each set of characteristics ($P_c = 50$ and $P_c = 100$).

3.2 Regularization Techniques

To achieve the best model performance, Gu et al. [16] rely on hyperparameter tuning as a regularization procedure to prevent overfitting. In other words, by tuning the model's parameters, Gu et al. [16] manage its complexity. Examples of such measures include changing the penalization parameters in "Lasso" and the number of iterated trees in boosting.

The hyperparameters are set based on the most optimal out-of-sample performance. Specifically, the dataset is split into the training, validation, and testing sets. From these three, the hyperparameters are estimated using the training set. Then, using the validation set, the hyperparameters are tuned based on the forecast errors from that sample. Iteratively, Gu et al. [16] search for the hyperparameters that provide the best performance for the model. Figure 2 summarizes the hyperparameters set by Gu et al. [16].

To test the hypothesis of Camerer [5], this study will implement the following methodology: similar to Oskamp's [21] participants, as more features are added to the ML models, the accuracy score of the training set will improve; however, the accuracy score of the testing set will stagnate. At the same time, the confidence interval will increase [5, 21]. Such results will mimic the overconfidence of the participants of Oskamp's [21] study. The reason is that the widening confidence interval can be a sign of a more elaborate prediction without improving the accuracy of the ML model. Similarly, the participants' confidence in Oskamp's [21] study increases, while their accuracy does not improve. So, this study will compare a good model - ML model with regularization technique - and a "bad" model - ML model without/weakened regularization technique - to determine whether their divergence can help measure overconfidence.

I hypothesize that human overconfidence results from a failure to winnow the set of predictors (as in "Lasso" penalties for feature weights). Overconfidence of this type results from not anticipating overfitting. High training set accuracy corresponds to confidence in predictions. Overconfidence is a failure to anticipate the drop in accuracy from training to test. This reasoning stems from Camerer [5]: "In the predictive context, we will define (overconfidence) as having too narrow a confidence interval around a prediction." As a result, I will test Camerer's [5] hypothesis by testing whether (1) the training and testing set accuracy rate divergence and (2) the confidence interval increases as the accuracy rate decreases (overconfidence increases as the actual accuracy rate decreases). Further sections will discuss the methodology behind this regularization technique specific to each ML model.

3.3 ML Models

3.3.1 Simple Linear. The simple linear regression model uses ordinary least squares (OLS) to approximate $g^*(z_{i,t};\theta)$:

$$g^*(z_{i,t};\theta) = z_{i,t}^T\theta$$

where conditional expected returns depend on the linear relationship between the parameter vector θ and predictor variables. Such a model is expected to underperform with high-dimensional data since it does not capture non-linear relationships and interaction effects. This model is a pooled OLS estimator since it minimizes the standard least squares (l_2) objective function (OLS in Figure 1):

$$L(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(r_{i,t+1} - g^*(z_{i,t};\theta))^2$$

[16]

3.3.2 Penalized Linear. Due to that limitation, the number of estimated parameters can be decreased to decrease the risk of overfitting. So, Gu et al. [16] implement a regularization technique that penalizes the objective function:

$$L(\theta;\cdot) = L(\theta) + \phi(\theta;\cdot)$$

(11) where $\phi(\theta;\cdot)$ is the penalty function [16]. It favors parsimonious models, helping balance model complexity and performance by mechanically decreasing the model's in-sample performance such that the model fits poorly to noise and better to signal. Gu et al. [16] utilize the "elastic net" penalty:

$$\phi(\theta;\lambda,\rho) = \lambda(1-p)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2$$

where λ (controlling for the strength of the penalty) and ρ (controls the relative weight of L1 and L2 penalties) are the hyperparameters [16].

(1) "Lasso"
The case of $\rho = 0$ uses l_1 parameter penalization. This regularization technique is a variable importance technique: as λ increases, the coefficient of unimportant variables is set to zero (simply put, it discards the unimportant variables) [23].
(2) "Ridge"
The case of $\rho = 1$ uses l_2 parameter penalization. In this case, "Ridge" penalizes the largest coefficients more, while "Lasso" is uniform. As a result, this technique is one of shrinkage rather than variable importance since the "Ridge" prevents coefficients from being unnecessarily large [18].
(3) "Enet"
The case of the elastic net penalty combines the selection and shrinkage effects by setting intermediate values of ρ. It adaptively tunes both hyperparameters (λ and ρ) [16].

3.4 Performance Evaluation

The ML models' performance is assessed with out-of-sample R^2:

$$R^2_{\text{OOS}} = 1 - \frac{\sum_{(i,t)\in\tau_3}(r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{(i,t)\in\tau_3} r^2_{i,t+1}}$$

where τ_3 is a specific sub-sample of data not used for training or validating. In this case, the higher out-of-sample R^2, the better the model's performance because R^2 compares how well the model's predictions measure against the actual returns from the sub sample τ_3. In this case, the denominator subtracts the predicted and actual excess returns rather than taking the difference between the predicted and the average value (demeaning the data). Gu et al. [16] implement this approach because the current dataset has individual stock returns; the data does not have an aggregate index or long-short portfolios.

To assess the overconfidence of the ML model, this paper calculates the 95% confidence interval of the out-of-sample R^2 using the Bootstrap Resampling (BS) method [12]. It is the preferred methodology because it does not make assumptions about the underlying distribution of the data. As a result, the BS methodology randomly chooses indices with replacements, creating out-of-sample data. Then, the model is trained on that randomly selected sample. Later, the bootstrap sample helps yield predicted values. With such values, out-of-sample R^2 is calculated and stored. In this case, I rely on the rule of thumb: 1000 bootstrap iterations [16]. With each repetition, an out-of-sample R^2 is obtained. From this storage, the confidence interval for the out-of-sample R^2 is obtained by taking the 2.5th percentile and 97.5th percentile from the array of the out-of-sample R^2 from each 1000 bootstrap iterations.

3.5 Interpretation for Both Methodologies

The confidence interval of the out-of-sample R^2 represents the confidence in the out-of-sample R^2. In other words, if the model is certain about the out-of-sample R^2, then the confidence interval will be narrow and vice versa.

3.6 Testing Camerer's [5] Hypothesis

If the hypothesis of Camerer [5] is correct, training and testing set accuracy rates will diverge. This will occur because similar to Oskamp [21] participants, as ML models are trained on data with many features, the accuracy rate of the training set is very high. However, when introduced with new data, the ML model will underperform similar to how Oskamp [**?**] participants were overconfident based on most likely irrelevant features [5].

Since the ML models are in a predictive context, based on Camerer [5], the "bad" model – or the one that overfits – will have a greater confidence interval and lower out-of-sample R^2 relative to the "good" model (or the one that does not overfit).

In the context of the different ML models, if Camerer's [5] theory is correct, the models will have the following order:

- "Oracle" [16]
- "Lasso"
- "Enet"
- "Ridge"
- "Simple OLS"

This order is based on their tendency to overfit (greatest divergence between the training and testing set; greatest to least out-of-sample R^2 and least to greatest confidence interval).

The divergence between the training and testing sets increases based on that ranking ("Oracle" having the least divergence, and "Simple OLS" having the greatest). The reason is that the "Oracle" model represents no overconfidence. Thus, the divergence will be the least for the "Oracle" model since it is the model that shows the true relationship. On the other hand, the "OLS model" will not capture the complex relationships. Thus, it will have high confidence in the training set, but, in reality, the performance of that model is poor since its accuracy rate on unseen data should be low.

The "Oracle" model as defined by Gu et al. [16] has the true covariates in a pooled OLS regression. So, the source uses this model as a benchmark. In other words, in this case, it functions as a substitute for human predictions. By having the true factors, this model represents an ideal human with perfect knowledge of all relevant factors driving stock excess returns. The "Oracle" model is synonymous with a "non-confident" model. Thus, the "Oracle" model must have the highest out-of-sample R^2 and least confidence interval relative to other ML models.

The "Lasso" model discards irrelevant factors by setting their coefficients to zero, effectively removing irrelevant factors. According to Camerer's [5] hypothesis, it must have a slightly smaller out-of-sample R^2 and a larger confidence interval. In the context of Oskamp's [21] experiment, people are susceptible to more noise and, by extension, are less accurate. However, people are more confident in their results (greater confidence interval).

The "Enet" model uses the l_1 and l_2 parameter penalization, thus combining absolute value and squared term penalties on the coefficient. Unlike "Lasso," "Enet" can retain some factors that "Lasso" may deem irrelevant and, thus, discard from the model. Thus, "Enet" will have an out-of-sample R^2 less and a confidence interval greater than the one of "Lasso," based on Camerer's [5] theory.

The "Ridge" regression does not set the coefficients of irrelevant features to zero. Hence, it is more likely to retain irrelevant features than the "Enet" model.

Finally, the "Simple OLS" model includes all features, including the irrelevant features, since it does not perform variable selection. Out of all the ML models, the "Simple OLS" model is most likely to

overfit. As a result, as a model, the "Simple OLS" model will have the smallest out-of-sample R^2 and largest confidence interval.

4 ANALYSIS

The hypothesis of Camerer [5] is mildly supported by Monte-Carlo's simulated data. Figure 2 graphs the in-of-sample R^2, out-of-sample R^2, and the confidence interval for the out-of-sample R^2 and the average out-of-sample R^2 shown in Figure 1.

4.1 Divergence of the Training and Testing Sets

The hypothesis of Camerer [5] is right in terms of the training and testing set accuracy diverging in Figure 2 [5]. The "Oracle" model – or the "non-overconfident" model – has a consistent difference between the in-sample R^2 and out-of-sample R^2 compared to "Lasso", "Enet", and "Ridge." Since the confidence intervals of the "Oracle," "Lasso", "Enet", and "Ridge" models do not overlap, the comparison of the "Simple OLS" model with each one is statistically significant. The "Simple OLS" model has the greatest divergence between the training and testing set accuracy rate relative to each of the models: "Lasso", "Enet", "Ridge," and "Oracle." An important drawback of "Model A" is that it fails to capture non-linear relationships and interaction effects. When those are considered in "Model B", Camerer's [5] hypothesis is clearer. The "Oracle" model has little to no divergence between the in-sample R^2 and out-of-sample R^2. This divergence then steadily increases from the "Lasso" to the "Enet" to "Ridge" to "Simple OLS." Thus, "Model B" supports Camerer's [5] hypothesis that as more features are included, the training-set accuracy increases, but the training-set accuracy decreases, as evidenced by the extreme case of the "Simple OLS." This growing divergence is analogous to the overconfidence of Oskamp's [21] subjects, according to the hypothesis of Camerer [5].

4.2 The Confidence Interval and the Accuracy Score

For further validation, given the predictive context, I include the confidence interval of out-of-sample R^2 as suggested by Camerer [5]. Theoretically, a small confidence interval - the model is certain of the out-of-sample R^2 - represents "overconfidence."

Based on "Model A" with 50 characteristics, the confidence intervals of the "Oracle", "Lasso", "Enet", and "Ridge" overlap. Thus, the range of the confidence intervals is not statistically significant for those models. However, these ranges do not overlap with the confidence interval of the "Simple OLS" model. As a result, the "Simple OLS" model and the other models have a statistically significant difference in the confidence interval for the out-of-sample R^2 (see Figure 2). Since the confidence interval for the "Simple OLS" model is the largest, and its out-of-sample R^2 is the lowest relative to the other models, the hypothesis of Camerer [5] may align with such findings. "Model A" with 100 characteristics gives similar findings but with a greater divergence in the out-of-sample R^2 and confidence interval, supporting Camerer [5] to a greater extent.

"Model B", on the other hand, shows that the confidence intervals of the models "Lasso", "Enet", and "Ridge" do not overlap. Instead, the range of the confidence intervals of the "Oracle", "Ridge", "Simple OLS", and combined range of "Lasso" and "Enet" are statistically different. In this case, the range of the confidence interval of the "Oracle" and "Simple OLS" is similar (with the range of the confidence interval of "Oracle" being slightly larger by about 0.76%) although the out-of-sample R^2 differs by about 10.37%. In other words, despite the greater accuracy rate of the "Oracle" model, its confidence range is larger, challenging the hypothesis of Camerer [5]. The confidence intervals of "Lasso" and "Enet" overlap. Compared to "Ridge", all models – "Lasso", "Enet", and "Ridge" – have a similar confidence range and accuracy rate. Each of those models ("Lasso", "Enet", and "Ridge") has a smaller confidence interval and lower out-of-sample R^2 relative to the "Oracle" model, further challenging the hypothesis of Camerer [5].

A similar finding is present in "Model B" with 100 characteristics. The statistical difference between the models is the same; however, the range of the confidence interval of the "Oracle" model is slightly less than that of the "Simple OLS" model by about 0.4%. However, the range of the confidence interval of the "Lasso," "Enet," and "Ridge" models is about 1% less than for the "Oracle" model although the "Oracle" model has a higher out-of-sample R^2 of about 5%. As a result, the "Model B" with 1000 characteristics continues to challenge the hypothesis of Camerer [5].

5 CONCLUSION

This paper helps relate overconfidence in ML models with human overconfidence. It tests the hypothesis of Camerer [5]: as more features are added, the divergence between the training and testing sets will increase, reflecting human overconfidence; as the out-of-sample R^2 decreases, the confidence interval increases, reflecting the increasing confidence of the Oskamp's [21] participants despite their predictions' accuracy decreasing. Analyzing the out-of-sample accuracy rate and its 95% confidence interval, using the BS method, the evidence was found to partially support the hypothesis of Camerer [5], potentially helping improve market forecasts.

Regarding the divergence of the training and testing sets, "Model B" – the one capturing non-linear relationships – best supports the hypothesis of Camerer [5]. However, the results in terms of the accuracy rate with the confidence rate are less conclusive. For the simple "Model A", the "Simple OLS" model is more confident and less accurate relative to the "Oracle," "Lasso," "Enet," and "Ridge" models. In contrast, "Model B" challenges the hypothesis of Camerer [5], as the "Oracle" model has a large confidence interval despite its higher accuracy rate and the "Oracle and the "Simple OLS" models have similar confidence intervals despite different accuracy rates.

Future studies can further help validate his hypothesis by capturing the inherent complexity of human overconfidence using experiments with humans as participants instead of relying on the "Oracle" model [16]. Secondly, using market data rather than Monte-Carlo simulations can help truly reflect human overconfidence as a bias in the market. Exploring other ML models, different regularization techniques, and studying confidence intervals and out-of-sample accuracy may provide further insights into the nuances of overconfidence in the models.

That way, future studies can help eventually measure overconfidence as a factor when predicting excess returns. As shown by this study, the measure of overconfidence can incorporate overfitting to mimic that psychological bias, proving that potentially overfitting is a foe to many but a friend to overconfidence in the stock market.

6 ACKNOWLEDGMENTS

I am grateful to my advisor, Professor Jeffrey A. Livingston of Bentley University, for his constant support and guidance throughout this research project. His insightful feedback on my research direction and encouragement were instrumental in shaping this work.

I sincerely thank Professors Tom Connors, Jackie Masloff, James Pepe, and Gregory Vaughan (names ordered alphabetically) from Bentley University for their invaluable assistance with the computer science aspects of this research. Their expertise and support have been instrumental in advancing this work.

I also wish to thank Richard Moore for his significant contributions to this project.

Additionally, I am deeply grateful to the Jeanne and Dan Valente Center for Arts and Sciences at Bentley University for their generous financial support, which has made this research possible. Special thanks are due to Professors Johannes Eijmberts and Mark Frydenberg for their encouragement and guidance.

Finally, thank you to my parents always and for everything.

All mistakes are my own.

REFERENCES

[1] Marc Alpert and Howard Raiffa. 1982. 21. A progress report on the training of probability assessors. (1982).

[2] Soleman Alsabban and Omar Alarfaj. 2020. An Empirical Analysis of Behavioral Finance in the Saudi Stock Market: Evidence of Overconfidence Behavior. *International Journal of Economics and Financial Issues* 10, 1 (2020), 73–86. https://www.proquest.com/docview/2485442460?pq-origsite=gscholar&fromopenview=true

[3] Brad M. Barber and Terrance Odean. 2000. The Courage of Misguided Convictions: The Trading Behavior of Individual Investors. *Financial Analysts Journal* 55, 6 (2000), 41–55. https://doi.org/10.2139/ssrn.219175

[4] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2018. Reconciling Modern Machine-learning Practice and the Classical Bias–Variance Trade-off. *Proceedings of the National Academy of Sciences* 116, 32 (2018), 15849–15854. https://doi.org/10.1073/pnas.1903070116

[5] Colin F. Camerer. 2019. Artificial Intelligence and Behavioral Economics. In *The Economics of Artificial Intelligence: An Agenda*. National Bureau of Economic Research, 587–608. https://www.nber.org/system/files/chapters/c14013/c14013.pdf

[6] Esra A. Coşkun, Hakan Kahyaoglu, and Chi K. M. Lau. 2023. Which Return Regime Induces Overconfidence Behavior? Artificial Intelligence and a Nonlinear Approach. *Financial Innovation* 9, 1 (2023). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9845106/

[7] Kent Daniel and David Hirshleifer. 2015. Overconfident Investors, Predictable Returns, and Excessive Trading. *Journal of Economic Perspectives* 29, 4 (2015), 61–88. https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.2009.01443.x

[8] Kent Daniel and David Hirshleifer. 2018. *Overconfident Investors, Predictable Returns, and Excessive Trading*. Technical Report 21945. National Bureau of Economic Research. https://www.nber.org/system/files/working_papers/w21945/w21945.pdf

[9] Kent Daniel, David Hirshleifer, and Avanidhar Subrahmanyam. 1997. Investor Psychology and Security Market Under- and Overreactions. *The Journal of Finance* 53, 6 (1997), 1839–1885. https://www.jstor.org/stable/117455

[10] Kent D Daniel, David Hirshleifer, and Avanidhar Subrahmanyam. 2001. Overconfidence, arbitrage, and equilibrium asset pricing. *The Journal of Finance* 56, 3 (2001), 921–965.

[11] Richard Deaves, Erik Lüders, and Guo Ying Luo. 2009. An Experimental Test of the Impact of Overconfidence and Gender on Trading Activity. *Review of Finance* 13, 3 (2009), 555–575. https://doi.org/10.1093/rof/rfn023

[12] Philip M Dixon. 2006. Bootstrap resampling. *Encyclopedia of environmetrics* (2006).

[13] Baruch Fischhoff, Paul Slovic, and Sarah Lichtenstein. 1977. Knowing with certainty: The appropriateness of extreme confidence. *Journal of Experimental Psychology: Human perception and performance* 3, 4 (1977), 552.

[14] Markus Glaser and Martin Weber. 2007. Overconfidence and trading volume. *Geneva Risk and Insurance Review* 32 (2007), 1–36. https://link.springer.com/content/pdf/10.1007/s10713-007-0003-3.pdf

[15] Mark Grinblatt and Matti Keloharju. 2008. Sensation Seeking, Overconfidence, and Trading Activity. *The Journal of Financial and Quantitative Analysis* 64, 2 (2008), 549–578. https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.2009.01443.x

[16] Shihao Gu, Bryan T. Kelly, and Dacheng Xiu. 2017. Empirical asset pricing via machine learning. *SSRN Electronic Journal* (2017). https://doi.org/10.2139/ssrn.3159577

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

[18] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.

[19] Andrew Y. Ng. 1997. *Preventing 'Overfitting' of Cross-Validation Data*. Technical Report. School of Computer Science Carnegie Mellon University. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.5297&rep=rep1&type=pdf

[20] Terrance Odean. 1998. Volume, volatility, price, and profit when all traders are above average. *The journal of finance* 53, 6 (1998), 1887–1934.

[21] Stuart Oskamp. 1965. Overconfidence in Case-Study Judgements. *Journal of Consulting Psychology* 29, 3 (1965), 261–265. https://doi.org/10.1037/h0022125

[22] Meir Statman, Steven Thorley, and Keith Vorkink. 2006. Investor Overconfidence and Trading Volume. *The Review of Financial Studies* 19, 4 (2006), 1531–1565. https://www.jstor.org/stable/pdf/4123481.pdf

[23] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58, 1 (1996), 267–288.

[24] Alexander Tsigler, Gabor Lugosi, Peter Bartlett, and Phil Long. 2020. Benign Overfitting in Linear Regression. *Proceedings of the National Academy of Sciences of the United States of America* 117, 48 (2020), 30063–30070. https://research.google/pubs/pub47103/

[25] Xue Ying. 2019. An Overview of Overfitting and Its Solutions. *Journal of Physics: Conference Series* 1168, 2 (2019), 022022. https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/pdf

[26] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. 2018. A Study on Overfitting in Deep Reinforcement Learning. https://research.google/pubs/pub47103/

7 APPENDIX

"Model A"										
Parameter	$P_c = 50$					$P_c = 100$				
	R^2 (%)		Confidence Interval			R^2 (%)		Confidence Interval		
	IS	OOS	Absolute Difference			IS	OOS	Absolute Difference		
			Lower	Upper	Difference			Mean	Median	Difference
Oracle	7.15%	4.99%	4.06%	5.91%	1.85%	7.15%	4.99%	4.12%	5.91%	1.79%
Lasso	6.87%	4.68%	3.90%	5.38%	1.48%	6.87%	4.68%	3.94%	5.36%	1.42%
Enet	6.87%	4.68%	3.94%	5.33%	1.40%	6.87%	4.68%	3.92%	5.38%	1.46%
Ridge	6.82%	4.44%	3.88%	5.01%	1.13%	7.00%	3.88%	3.32%	4.39%	1.08%
Simple OLS	8.40%	0.47%	-0.86%	1.79%	2.66%	9.10%	-2.07%	-3.58%	-0.54%	3.03%

"Model B"										
Parameter	$P_c = 50$					$P_c = 100$				
	R^2 (%)		Confidence Interval			R^2 (%)		Confidence Interval		
	IS	OOS	Absolute Difference			IS	OOS	Absolute Difference		
			Lower	Upper	Difference			Mean	Median	Difference
Oracle	5.95%	5.76%	4.91%	6.64%	1.73%	5.95%	5.76%	4.82%	6.63%	1.80%
Lasso	1.07%	0.87%	0.70%	1.05%	0.35%	1.07%	0.87%	0.68%	1.05%	0.37%
Enet	1.07%	0.87%	0.69%	1.05%	0.36%	1.07%	0.87%	0.69%	1.05%	0.35%
Ridge	1.39%	0.19%	0.04%	0.35%	0.31%	1.47%	0.13%	9.80E-05	0.27%	0.26%
Simple OLS	2.93%	-4.61%	-5.52%	-3.64%	0.97%	3.94%	-7.69%	-8.80%	-6.60%	2.20%

Figure 1: These tables show the average in-sample (IS) out-of-sample (OS) R^2 and the lower and upper bound of the 95 percent confidence interval for the out-of-sample R^2 for models (a) and (b) using "Oracle," Lasso, Enet, Ridge, and Simple OLS. When running the Monte Carlo simulation, this study adopts the restraints set by Gu et al. (2017): the number of Monte Carlo repetitions is 100, $N = 200$, $T = 180$, and $P_x = 2$ [16].

Received 27 May 2024

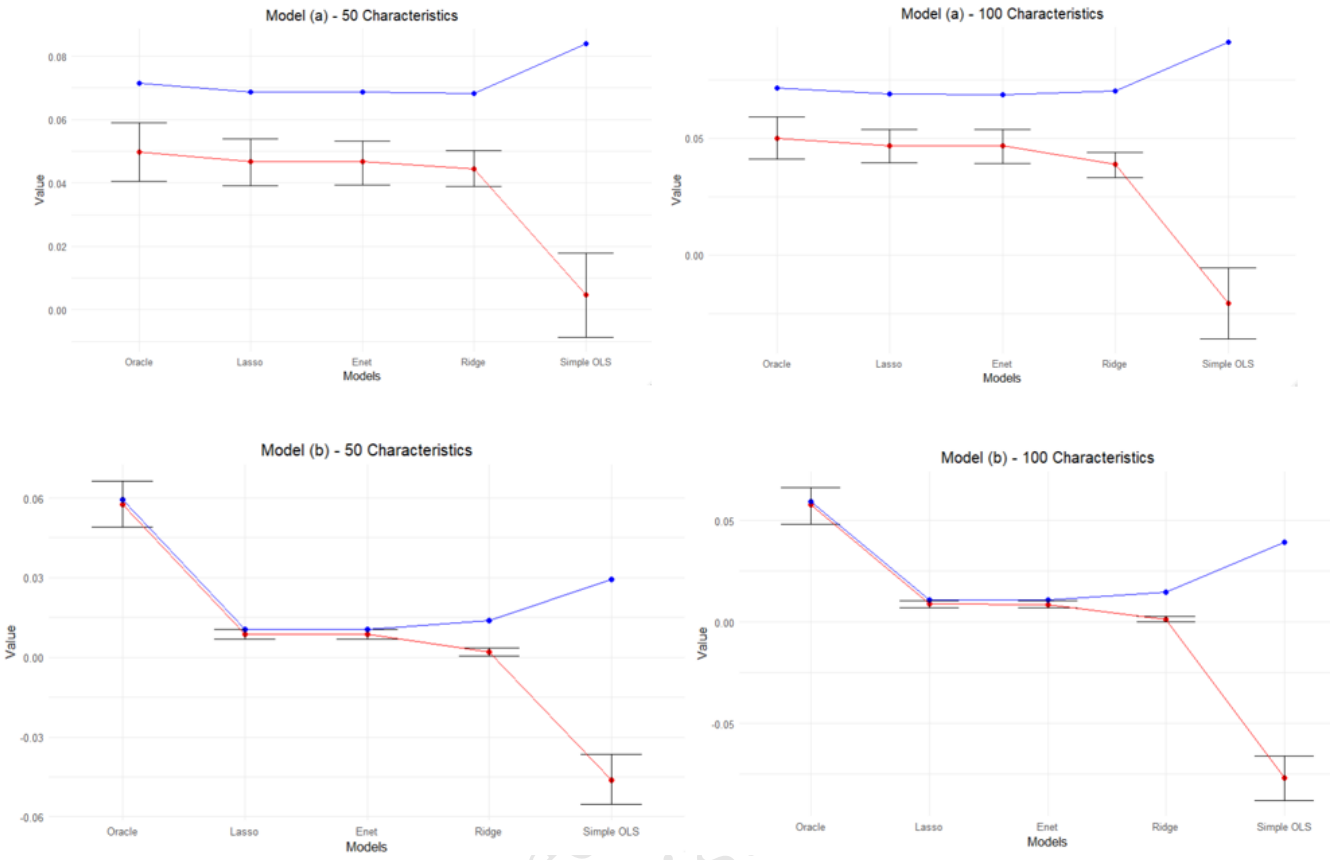

Figure 2: Visual Comparison of the Divergence of the Training and Testing Set and the Confidence Intervals with the Accuracy Rate Using the BS Method.