CIS 5200 Machine Learning
Fall 2022
Final Project Report

Patrick Cappello, Sriya Reddi, Aditya Jayant Ganapathiraju
Pod TA: Matthew Pressimone

December 12, 2022

1 Abstract

The goal of the project is to predict the prices of gold from various economic indicators using a supervised learning approach and then to use the same logic to predict the prices of silver by transfer learning.

We have used a wide variety of economic indicators to get stronger predictions, such as the Gross Domestic Product (GDP), Consumer Price Index, Unemployment Rate, and other critical indicators. These indicators are also uniformly reported all around the globe for different countries, which gives our model extreme transferability.

We used a linear regression model as our baseline, and then we used the results to compare the results obtained from other methods such as XGBoost, Elastic Net, and a self-designed Dense Feedforward Neural Network which are known to perform better. The following paper documents our results, findings, inferences, and conclusions, along with some future work that can help further explore this idea.

2 Motivation

Gold, one of the few metals which are considered to be precious, has its reputation and power go back to ancient times. Our ancestors used to fashion themselves, for trade, and to display wealth and influence. It was known to be one of the earliest currencies and was the backbone of world trade for centuries, and still is one of the key financial tools that countries have at their disposal. It is known to carry countries through recessions and rescue entire economies. Before the US dollar was made the standard currency of the globe in the 1970s, gold held that title for thousands of years.

Even though it is now not the primary currency for most countries, the appeal, and influence of the yellow metal still affects currencies of the world. Unlike the paper currencies that can be printed to influence inflations and other factors, gold is known to be less volatile and hence is still used as a reserve currency in many countries.

The current financial and economic conditions of the world have not been looking great, leading to the impending and ever-increasing likelihood of recession, especially after the Ukraine-Russia Crisis and Covid-19. Most countries have already registered many 'successful' quarters of negative GDP growth, increasing inflation, and high-interest rates. Continuous increase in layoffs and restricted hiring has unemployment rates skyrocketing, and decreased credit from banks has businesses and people scrambling for finances, which in turn leads to more economic downturn. Hence, examining and predicting the price of this illustrious metal will be a key indicator in predicting the world's economic health and help us better prepare ourselves against black swan events and unpredictable recessions and shocks.

3 Related Work

We wanted to use a dataset that involved economic indicators to predict the prices of gold. Most of the other implementations used stock prices of gold to predict it such as [1]. There were a few interesting approaches to predicting gold prices. [2] used random forests to predict the prices of gold and the model did a very good job, with both the actual and predicted prices overlapping with great accuracy and precision, it was very impressive. Again they used stock prices of gold to predict. Also, they used the R squared score metric to validate their model. [3] used the old but gold linear regression, and the model actually performed quite well, they used the standard 80 percent training dataset and 20 percent test dataset. The IEEE paper on Deep Learning techniques to predict gold prices, primarily used CNNs, RNNs, and LSTM [4]. Another interesting project paper [5] predicted the prices of Crude oil and Gold to see if they are interdependent. They mention that peaks of the gold prices in the US coincide with two significant economic recessions in history, in 1980 and 2008. They worked with timeseries data and used metrics such as RME, MAE and RMSE to test the performance of their models. They used both regressive and autoregressive models.

4 Data Set

The different economic indicators that were collected are given below. Since most of the ideal variables are collected on a monthly basis even the gold and silver prices dataset used was collected on a monthly basis. The final dataset included the economic indicators, gold prices, and silver prices from April 1968 to July 2021 which gave us around 637 rows/data points.

1. Price adjusted GDP - Dataset

2. Median CPI - Dataset

3. Unemployment Rate - Dataset
4. Non-farm payroll - Dataset
5. Consumer Opinion Surveys: Confidence Indicators - Dataset
6. Personal Consumption Expenditures - Dataset

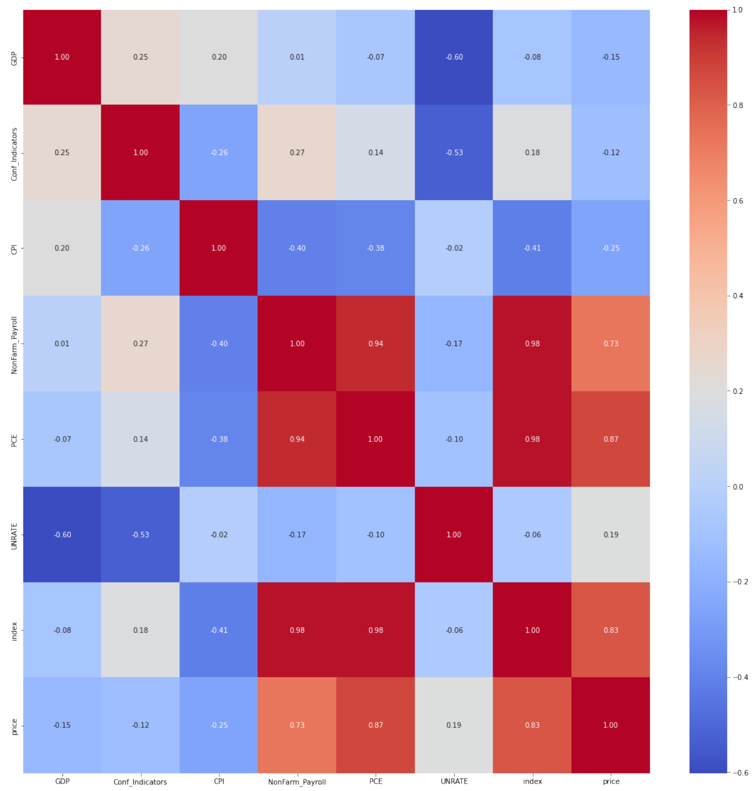


Figure 1: Heat Map of Economic Indicators with Gold Price

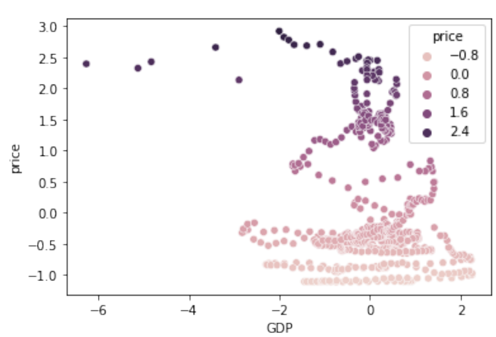


Figure 2: GDP vs Price of Gold

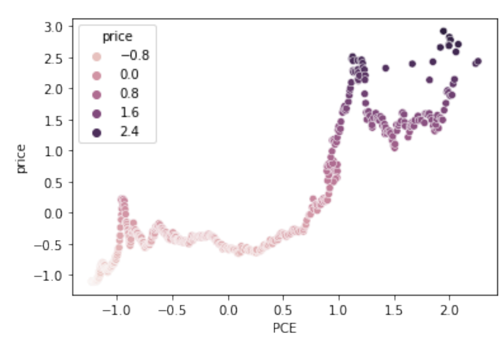


Figure 3: PCE vs Price of Gold

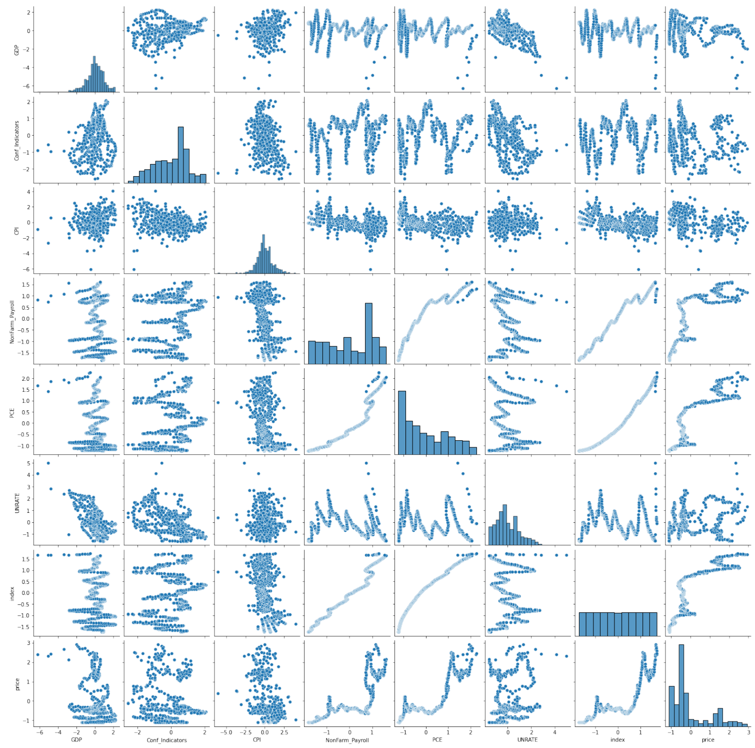


Figure 4: Pairwise Relationships of Economic Indicators with Gold Price

5 Problem Formulation

Our goal is to predict the price of gold using economic indicators such as GDP, CPI, PCE, etc (as mentioned in the previous section). Since this is a regression problem we will be using Linear Regression as the baseline model. We will also use models such as Elastic Net, and XGBoost and design and implement our own Neural Network. To evaluate and infer the results of all these models similarly we will be calculating the Mean Squared Error (MSE) Loss. We would be doing the same on a similar but different dataset with the prices of silver.

Our goal is to also try to implement transfer learning by using the trained Neural Network model on another dataset with the prices of Silver.

Though we started with 8 economic indicators initially, upon feature engineering and sampling the data we decided to use the 6 economic indicators that are mentioned in the previous section. This was based on the importance of the feature and the data availability. We will also be taking into consideration an additional feature, the date as an index.

6 Methods

6.1 Baseline Model

Since we have a regression problem, we figured it would be best to use a linear regression model as the baseline. Although the label (price) is not necessarily linear with respect to our features, this model would show us the minimum requirements for our more sophisticated models to outperform.

6.2 Other Methods

Our other models consist of XGBoost, Elastic Net, and Dense Feedforward Neural Network. We chose XGBoost because it is known to be one of the best machine-learning algorithms and our problem formulation allows us to use this model effectively. We chose the Elastic Net model because it will almost surely perform better than linear regression since it is the same model but with regularizing terms to avoid overfitting. Lastly, we chose to run a neural network as well because it is known for being able to approximate very complicated non-linear functions well and this problem would be a perfect example of how the neural net could be applied in the context of regression.

6.3 Implementation

Our linear regression and elastic net models were implemented with Scikit-learn. Our XGBoost regressor model was implemented with the XGBoost library. Our neural network was implemented using PyTorch.

7 Experiments and Results (Evaluation and Analysis)

7.1 Experimental/Evaluation Framework

We optimized our models by trial and error and chose the hyperparameters with the lowest validation error. The baseline linear regression model did not have any hyperparameters to tune. For XGBoost, Elastic Net, and Neural Net, we tuned the hyperparameters by starting with a default value, running and evaluating the model, making small changes in the direction of decreasing error, then running and evaluating the model again. This method of hyperparameter optimization assumes that the mean-squared error is convex with respect to each hyperparameter (learning rates, regularization coefficients, number of estimators, etc.); meaning there is a value for each of the hyperparameters such that the mean-squared error is at a minimum.

7.2 Performance Metrics

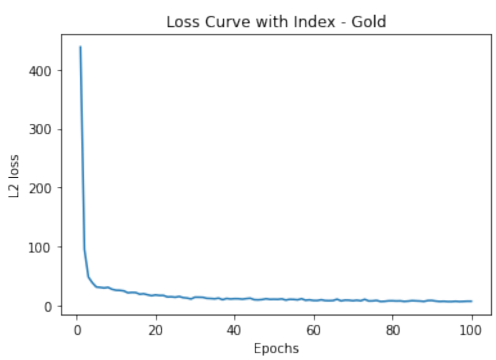
We evaluated each of our models with mean-squared error. We chose mean-squared error because this is the standard performance metric for regression problems. Also, each of our models are optimized to minimize square error so evaluating each of the models on their mean-squared error is a valid performance metric. Since our labels have 0 mean and 1 variance, the mean-squared error has an interpretable meaning: the square root of the MSE is the number of standard deviations from the mean that the model predicts on average.

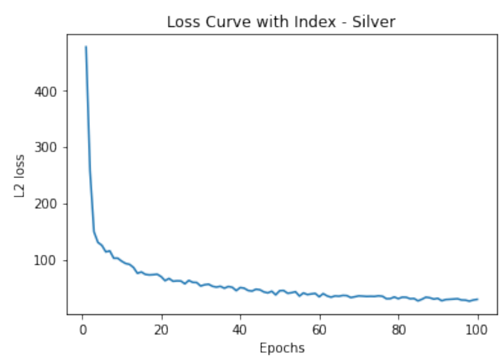
7.3 Performance Results

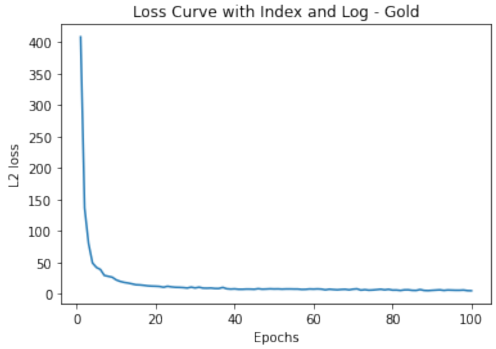
Data\Model		Linear Regression		Elastic Net		XGBoost		Neural Network	
X Data	Y Data	Gold	Silver	Gold	Silver	Gold	Silver	Gold	Silver
With Index	Without log values	0.0635	0.1984	0.0629	0.1971	0.0042	0.0286	0.0135	0.0637
With Index	With log values	0.1096	0.1769	0.1103	0.1767	0.0031	0.0147	0.0116	0.0299
Without Index	Without log values	0.0808	0.1970	0.0806	0.1969	0.0052	0.0271	0.0133	0.0690
Without Index	With log values	0.1242	0.1769	0.1240	0.1767	0.0031	0.0145	0.0099	0.0279

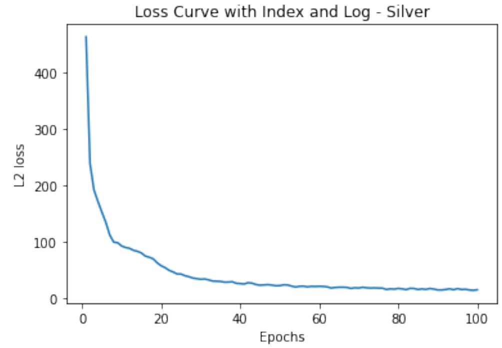
Neural Network Loss Curves

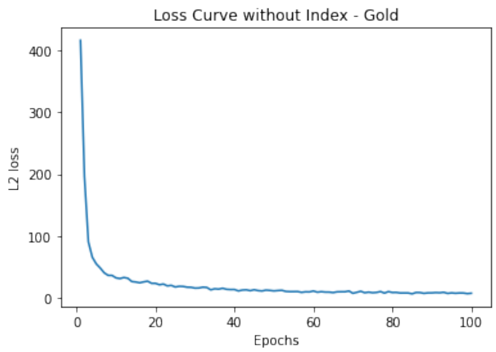
The below plots are the loss curves generated by the neural network for the different aspects tested on the data as shown in the table. The first two plots represent the loss values when the X data includes the index as a feature and the Y data is standardized. The next two plots represent the loss values when the X data includes the index as a feature and the Y data is standardized after taking the log of the values. The next two plots represent the loss values when the X data excludes the index as a feature and the Y data is standardized. The next two plots represent the loss values when the X data excludes the index as a feature and the Y data is standardized after taking the log of the values.

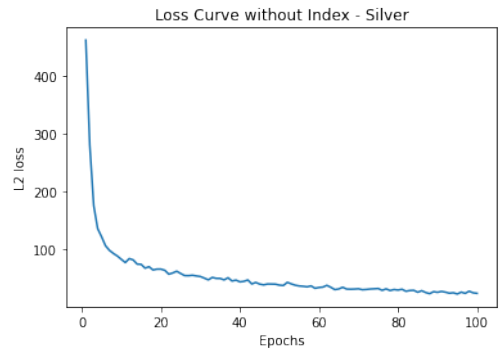


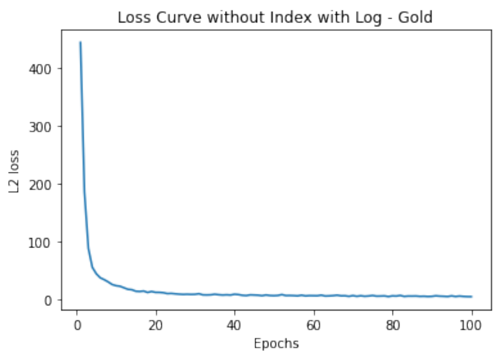


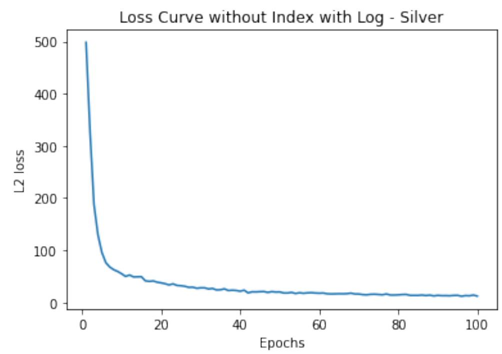












8 Conclusion and Discussion

8.1 Summary of Results

The model with the best performance is XGBoost. We initially thought that the neural net would outperform all of the models; however, neural networks require a lot of data and there wasn't enough data available to train the neural net to its greatest potential. The linear regression model performed the worst of all. This may be due to the fact that the price labels are not linear with respect to our features and with such a simple model, the linear regression could not fit the data well. The elastic net model performed very similarly to the linear regression model. This is due to the fact that the optimal regularization hyperparameters are extremely small resulting in a model that acts nearly the same as linear regression.

8.2 Qualitative Interpretations of Results

Based on how our models performed, it is reasonable to conclude that our models are underfitting. This is likely due to not having an abundance of data to train our models on. The small regularization hyperparameters for the elastic net model also lead us to believe that linear regression and elastic net models are underfitting. The dropout layer ($p = 0.1$) in our neural network made only very slight improvements to the validation error which is an indication that the neural network may be underfitting. Hypothetically, we could expect that if the dropout layers with larger p values improve the model performance by a significant factor, we may conclude that the original model is overfitting.

8.3 Takeaway

The primary lesson from this project is the importance of having an abundance of data so there can be more room for intervening to find the balance between bias and variance. We also found that the neural network is not the ultimate problem solver, it has its limits, one of which is that it requires a lot of data to perform extremely well. Another lesson is that transfer learning does not necessarily improve and apply to every kind of problem. The fact that gold and silver display different patterns over time (given the same economic indicator data) implies that it likely would not have much of an improvement to train a model with a full dataset and half of the other and test on the other. One thing that helped a lot was taking the time to manually find the optimal hyperparameters for each of the models that had hyperparameters to optimize.

8.4 Future Directions

An extension to this project would be to predict how the prices of gold and silver behave in the future. A prediction problem like that would likely require more data and additional features that might be better indicators of the prices of gold and silver. This would also likely require the use of time series models such as recurrent neural networks and long short-term memory networks.

9 References

[1] Kaggle - Price Prediction using ML

[2] Building a gold price prediction model using ML

[3] Another gold price prediction python implementation

[4] V. G. S and H. V. S, "Gold Price Prediction and Modelling using Deep Learning Techniques," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2020, pp. 28-31, doi: 10.1109/RAICS51191.2020.9332471.

[5] Predicting the prices of Oil and Gold

10 Appendix

Figure 5: Heat Map of Economic Indicators with Silver Price

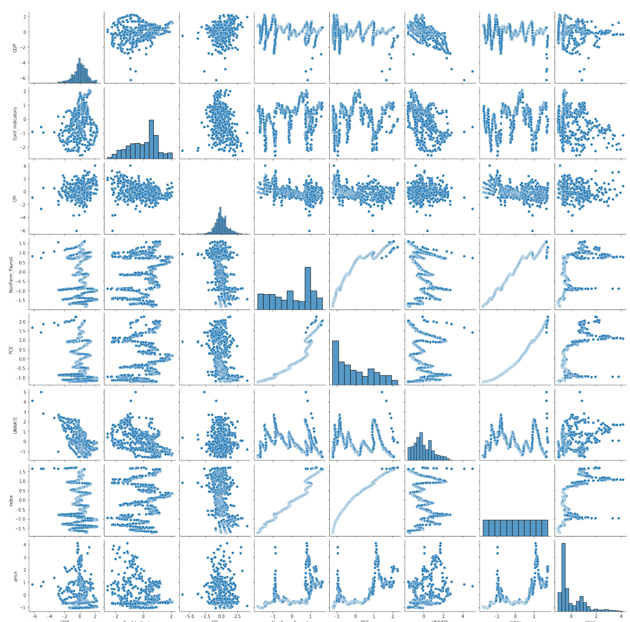


Figure 6: Pairwise Relationships of Economic Indicators with Silver Price