Forecasting by Traditional and ML Methods:

Implementing Competing Supervised Learning Models to Forecast the Logarithmic Fraction of All Outstanding Shares, a Project Overview

Tanner Woods

21 March 2023
ECON 425T

Table of Contents

Initial Analysis of Dataset

Statement of Purpose

{}

Variable Definitions

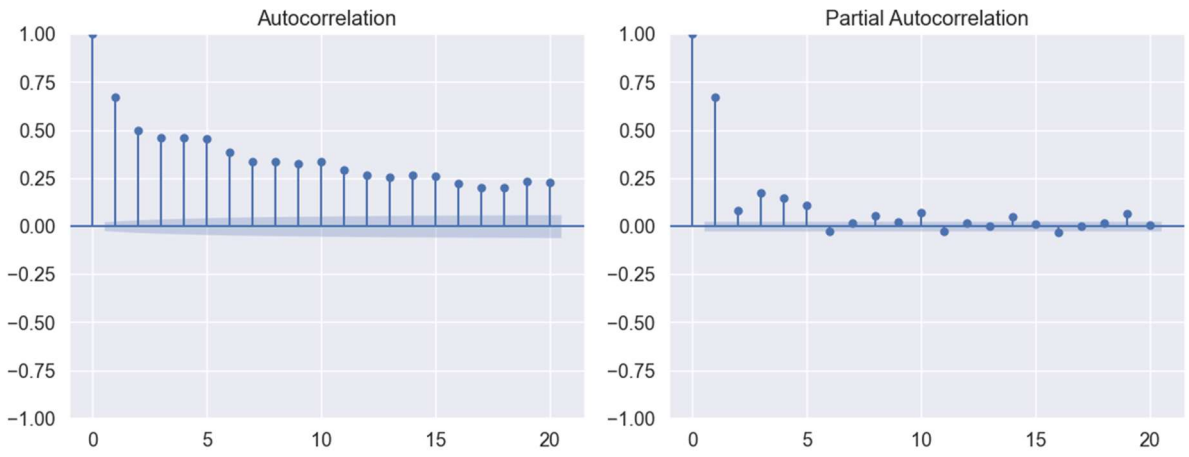
{}

Visualization: Autocorrelation (ACF) and Partial Autocorrelation (PACF)

ACF and PACF of Log Trading Volume

TEXT HERE

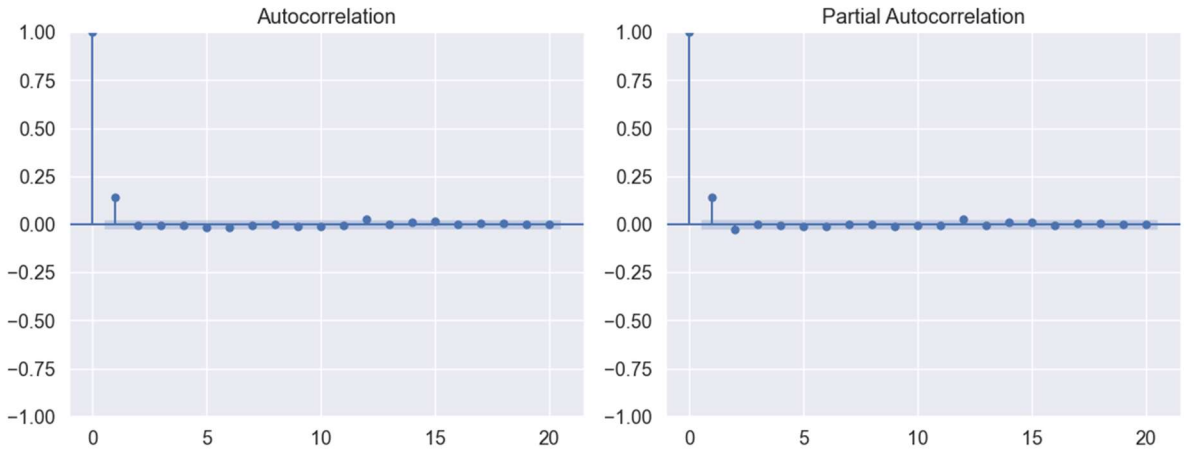
Figure 1. ACF and PACF of Log Trading Volume



ACF and PACF of Log Dow Jones Returns

TEXT HERE

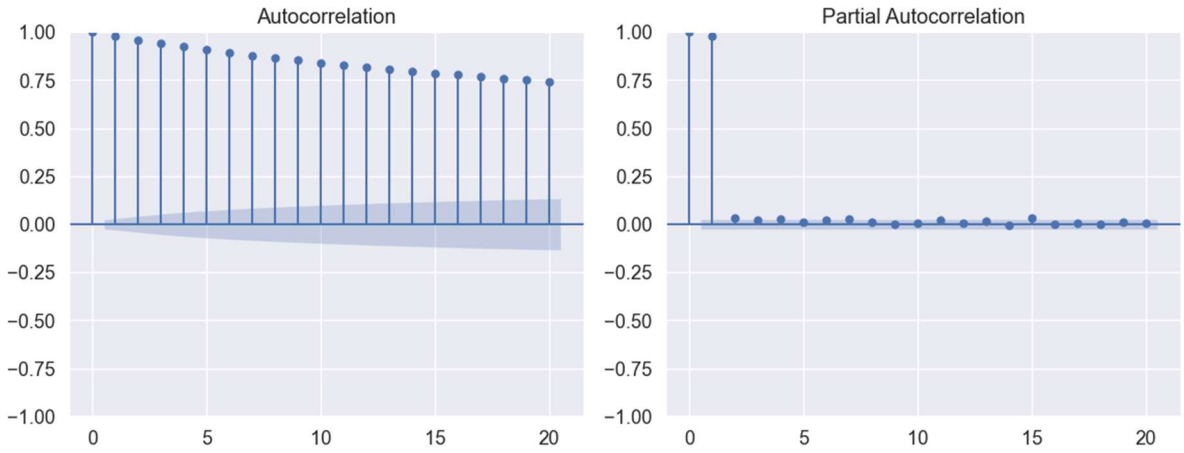
Figure 2. ACF and PACF of Log Dow Jones Returns



ACF and PACF of Log Volatility

TEXT HERE

Figure 3. ACF and PACF of Log Volatility



Baseline (Strawman) Model

Defining the Strawman

Implementing a process for forecasting of v_t by letting $\widehat{v_t} = \beta_0 + v_{t-1}$ — that is, predicting the current log volume via the previous period log volume, with an intercept term to catch the remainder.

Model Performance

Two methods of model performance were implemented here: extracting the R-squared score from the regression of $v_t \sim \widehat{v_t}$, and raw computation of the score. This results in the following scores:

$$R^2_{regression} \approx 0.348$$

$$R^2_{computation} = 1 - \frac{\sum(v_t - \widehat{v_t})^2}{\sum(v_t - E(v_t)^2} \approx 0.334$$

Autoregressive Model with Elastic Net Regularization

Hyperparameters

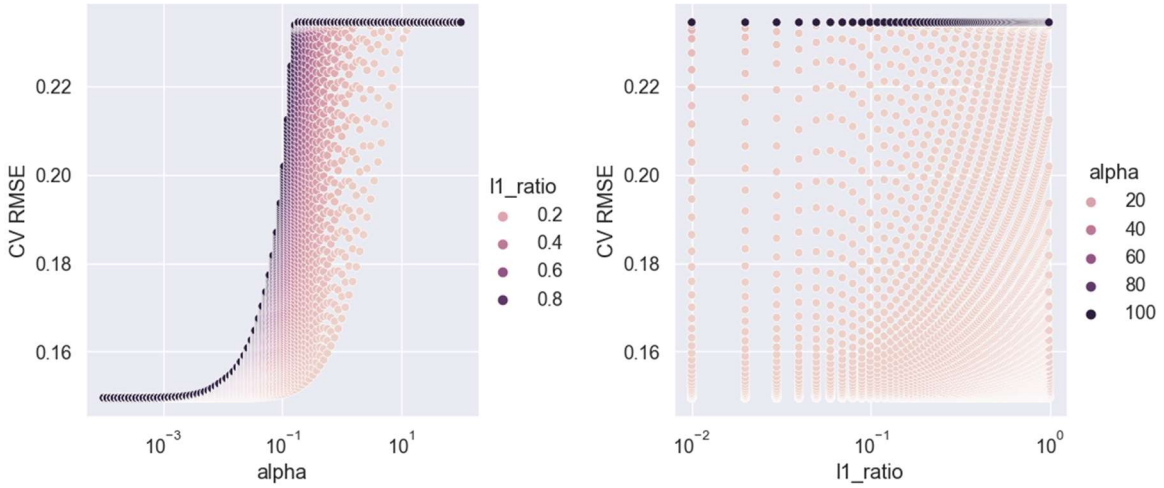
For the baseline autoregressive model, the hyperparameters passed through to the pipeline model search are as follows:

1. Learning rate, $\alpha = [a_1 = 0.01, a_2 = 0.02, \dots, a_{100} = 1.00]$
2. L1 penalty, $L1 = [l1_1 = 0.01, l1_2 = 0.02, \dots, l1_{100} = 1.00]$

Fixed values for parameters are the lag of $L = 5$ and number of cross-validation folds of $CVFolds = 10$.

Visualization: Cross-Validation Results

Figure 4. Cross-Validation Results for AR Model with EN Regularization



Tuned Model Performance

From the tuning process, the most optimal hyperparameter values appear to be $\alpha^* = 0.01$ and $L1^* = 0.01$. Under these hyperparameters, we obtained the following metrics:

Table 2. Performance of AR Model with Elastic Net Regularization

Cross-Validation		**Test**	
$\boldsymbol{R^2}$	**RMSE**	$\boldsymbol{R^2}$	**RMSE**
0.5989	0.1495	0.1781	0.4487

With saved predictions constructed from the model, we can also visualize the performance of our model by comparing the value of true test values against predicted test values:

Figure 5. True vs. Predicted Values for the AR Model with EN Regularization

Autoregressive Model with Multilayer Perceptron Tuning

Hyperparameters

For the autoregressive model with multilayer perceptron tuning, the hyperparameters passed through to the pipeline model search are as follows:

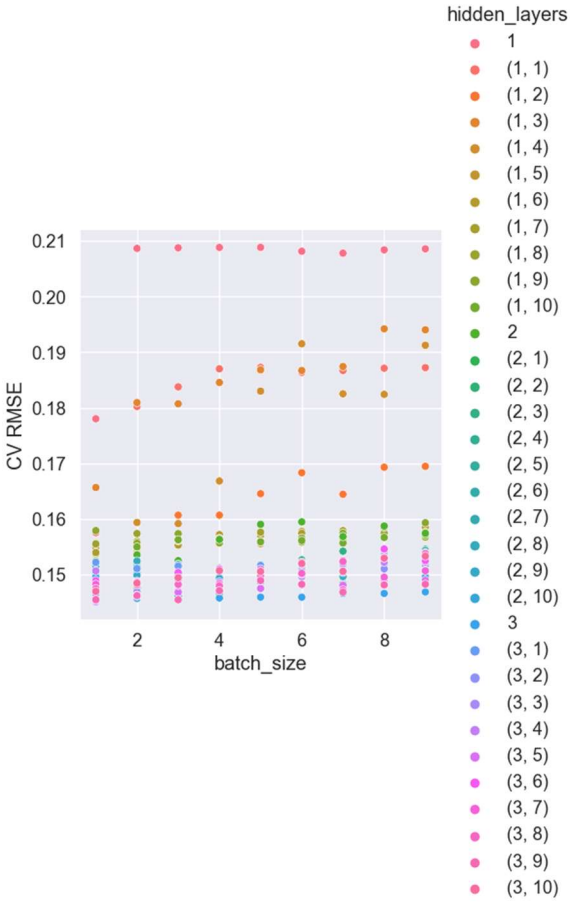
1. Hidden layer size, $HLS = \begin{bmatrix} 1 & (1,1) & \dots & (1,10) \\ 2 & (2,1) & \dots & (2,10) \\ 3 & (3,1) & \dots & (3,10) \end{bmatrix}$
2. Batch size, $B = [b_1 = 1, b_2 = 2, \dots, b_{10} = 10]$

Attempts at tuning for a three-layer perceptron network were frustrated by insufficient computing power. Fixed values for parameters are the lag of $L = 5$ and number of cross-validation folds of $CVFolds = 10$.

Visualization: Cross-Validation Results

(See next page)

Figure 6. Cross-Validation Results for AR Model with MLP Tuning



Tuned Model Performance

From the tuning process, the most optimal hyperparameter values appear to be $hls^* = (3,3)$ and $b^* = 1$. Under these hyperparameters, we obtained the following metrics:

Table 3. Performance of AR Model with MLP Tuning

Cross-Validation		**Test**	
$\boldsymbol{R^2}$	**RMSE**	$\boldsymbol{R^2}$	**RMSE**
0.6342	0.1450	0.1667	0.5170

With saved predictions constructed from the model, we can also visualize the performance of our model by comparing the value of true test values against predicted test values:

Figure 7. True vs. Predicted Values for the AR Model with MLP Tuning

Random Forest Algorithm Model

Hyperparameters

For the random forest algorithm model, the hyperparameters passed through to the pipeline model search are as follows:

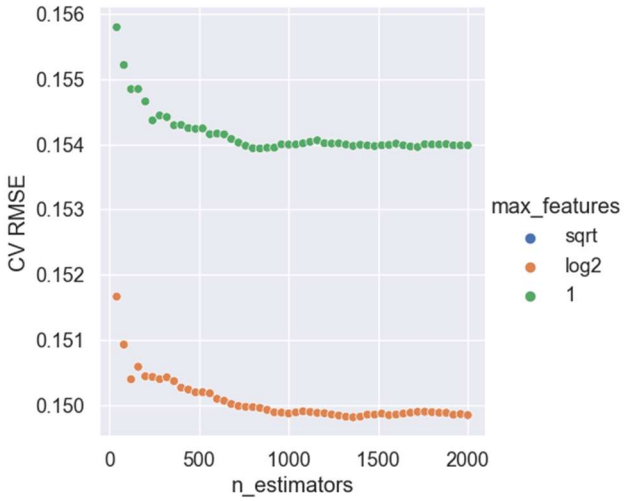
1. Max features, $MF = [mf_1 = "sqrt", mf_2 = "log_2", mf_3 = 1]$
2. Max estimators, $BS = [bs_1 = 40, bs_2 = 80, \dots, bs_{50} = 2000]$

Fixed values for parameters are the lag of $L = 5$ and number of cross-validation folds of $CVFolds = 10$.

Visualization: Cross-Validation Results

Over the cross-validation process for this model, the CV RMSE for the max features hyperparameter values of log_2 and $sqrt$ are approximately equal. This is why **Figure 7** appears to omit the values for the latter: they are effectively the same as the former.

Figure 7. Cross-Validation Results for RF Algorithm Model



Tuned Model Performance

From the tuning process, the most optimal hyperparameter values appear to be $mf^* = "sqrt" = "log_2"$ and $bs^* = 1360$. Under these hyperparameters, we obtained the following metrics:

Table 4. Performance of RF Algorithm Model

Cross-Validation		**Test**	
R^2	**RMSE**	R^2	**RMSE**
0.9482	0.1498	0.1757	0.4631

Boosting Algorithm Models

Boosting Algorithm Model with XGBoost

Hyperparameters

For the autoregressive model with multilayer perceptron tuning, the hyperparameters passed through to the pipeline model search are as follows:

1. Max tree depth, $D = [d_1 = 1, d_2 = 2, \dots, d_6 = 6]$
2. Learning rate, $\alpha = [a_1 = 0.01, \dots, a_{100} = 1.0]$
3. Max estimators, $BS = [bs_1 = 40, bs_2 = 80, \dots, bs_{50} = 2000]$

Fixed values for parameters are the lag of $L = 5$ and number of cross-validation folds of $CVFolds = 10$.

Visualization: Cross-Validation Results

Tuned Model Performance

Boosting Algorithm Model with Scikit-learn

Hyperparameters

Visualization: Cross-Validation Results

Tuned Model Performance

Long/Short-Term Memory Model

Hyperparameters

Visualization: Cross-Validation Results

Tuned Model Performance

Summary of Model Performances

From the table below, we can determine that X model performs the most optimally in forecasting the

Table 8. Summary of Model Performance in Cross-Validation and Testing

Model	**Cross-Validation**		**Test**	
	R^2	**RMSE**	R^2	**RMSE**
Baseline (Straw.) Model				
AR Model with EN Reg.	0.5989	0.1495	0.1781	0.4487
AR Model with MLP Tuning	0.6342	0.1450	0.1667	0.5170
RF Algorithm Model	0.9482	0.1498	0.1757	0.4631
BoostSL Algo. Model				

BoostXG Algo. Model				
LSTM Model				

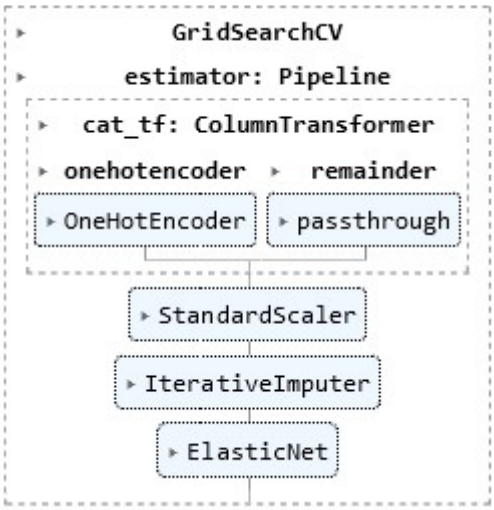
Appendix

Figure 3.

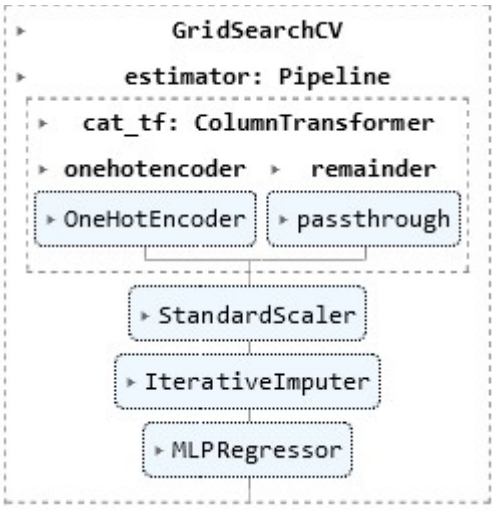
Figure X. Pipelines for All Non-Strawman Models

Pipeline: AR Model with EN Regularization

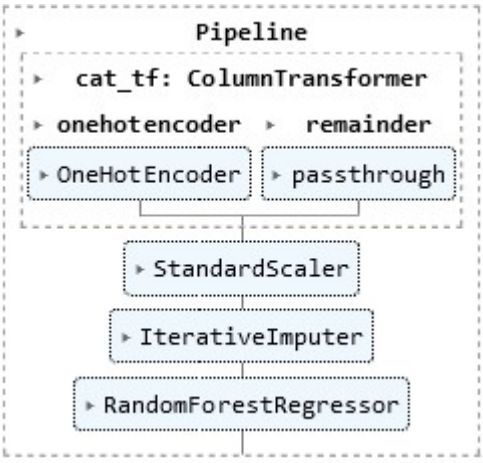
Pipeline: AR Model with MLP



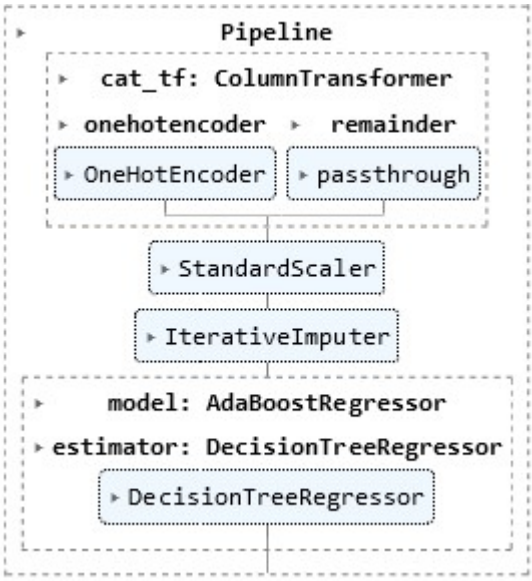
Pipeline: RF Algorithm Model



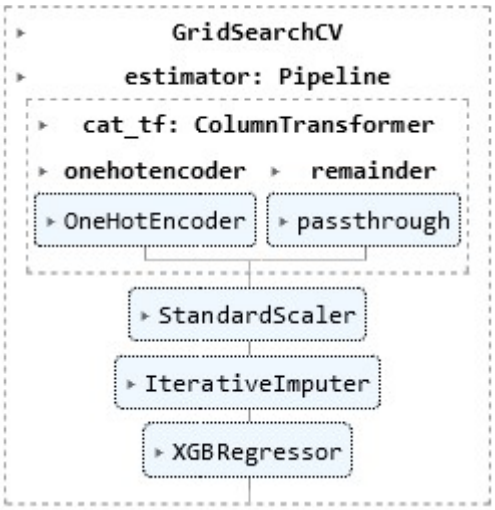
Pipeline: BoostingSL Model



Pipeline: BoostingXG Model



Pipeline: LSTM Model


GridSearchCV
estimator: Pipeline
cat_tf: ColumnTransformer
onehotencoder
remainder
OneHotEncoder
passthrough
StandardScaler
IterativeImputer
XGBRegressor
