IE582 - Short Term Stock Value Forecast

Abdullah Kayacan, Halis Oğuz

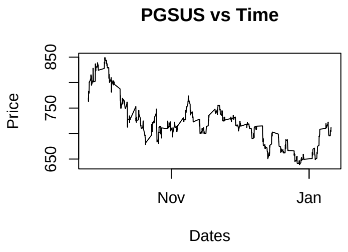
January 2024

1 Introduction

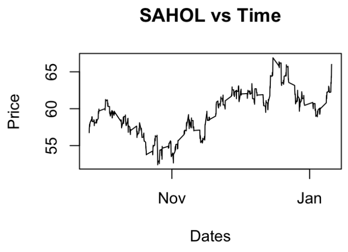
This is an authentic work prepared for IE582(Statistical Learning for Data Mining) under the guidence of Assoc. Prof. Mustafa Gökçe Baydoğan. The main purpose of this project is to form short term stock value forecasts for 30 distinct stocks in BIST(Borsa Istanbul). The error metric is WMAPE(Weighted Mean Absolute Percent Error), for each stock one is provided with data starting from 01/01/2018 up to now. We are expected to form 14 forecasts for the upcoming 14 days, we need to form a forecast for a day ,say t, at the end of the day t-1. Thus, we have all the price and change information available up to the current day. However, the issue is that a stock price may be manipulated heavily by an individual transaction when transaction at hand consists of a considerable shares of the stock. Further, financial markets are influenced through political updates. Therefore, although the availability of the data renders the procedure less cumbersome, there are complex non-linear interactions between a stock price and many exogenous factors which establishes the cumbersome side of the forecasting process.

As a first step, let us form an introductory data exploration on certain stocks without loss of generality. The stock prices are generally known to be non-stationary and highly volatile. As expected, the following data analysis is going to prove us correct. Even though upon checking prices after 25/09/2023, one can observe that data does not seem to be weakly stationary. A time series may be attributed weakly stationary when mean is stable with respect to time and auto-covariance or correlation information may exist but does not change in time. Let us present two examples from earlier periods, after 25/09/2023, and it may easily be observed that mean is not stable with respect to time, i.e. there is a trend associated with respect to time.

In Figure 1, we may observe that after choosing two random stocks weak stationarity does not hold. Further, variance does not seem to be stable as time progresses. Therefore, we have non-stationary data to form a model which provides us with a difficult task. Also, we may observe that scales of the prices differ. As an example, on Figure 1 (a) PGSUS data ranges approximately between 650-850 whereas SAHOL ranges between 55-65. Also, it may be observed from

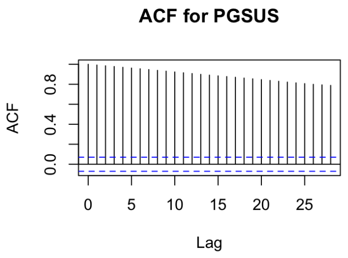


(a) PGSUS original prices

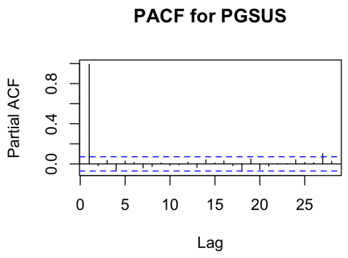


(b) SAHOL original prices

Figure 1: PGSUS and SAHOL vs time



(a) ACF



(b) PACF

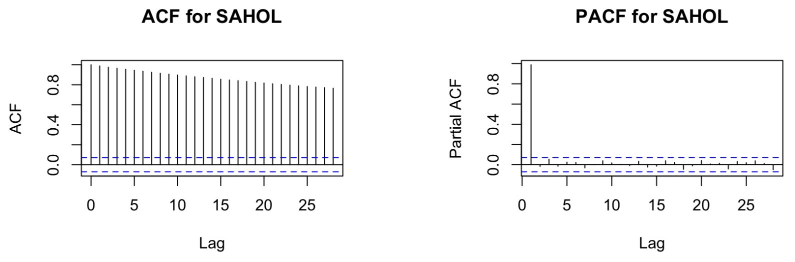
Figure 2: ACF and PACF of PGSUS

Auto-correlation function(ACF) and Partial Auto-correlation function(PACF) for PGSUS(Figure 2) and SAHOL(Figure 3) that either there exists a persistent trend or an auto-regressive term of order 1. This induction may be done by the dying-out behavior of the ACFs and cutting-off behavior of the PACFs at lag 1.

To summarize, the stock prices' behavior are not stationary and some care must be taken while trying to model the behavior of the stock prices. To overcome the issues, two fundamental models will be utilized which are LightGBM and linear regression with auto-regressive components. Following sections will be describing the related literature, methodologies, results on a detailed manner.

2 Related Literature

Gradient Boosting Machines (GBMs) are a set of ensemble models that depend on sequentially implemented decision trees. GBMs are known for their superior



(a) ACF (b) PACF

Figure 3: ACF and PACF of SAHOL

performance in regression and classification tasks. The boosting idea became popular after the publication of AdaBoost [2]. The algorithm proposes the use of weak classifiers to produce a better one by weighting the training data according to the errors caused by the previous trees. The original paper of GBM [3] proposes a method that requires adding new trees based on the error criterion of the remaining trees. Contrary to Adaboost, GBM tries to minimize the residual error of the existing trees with a new tree instead of weighting erroneous data more in the next tree. Most popular GBM algorithms are developed in the 2010s. Extreme Gradient Boosting (XGBoost) [1] is one of the most famous GBM algorithms which brings some enhancements to the classical GBM to boost the performance. In order to prevent overfitting, regularization parameters were added to the model. In addition, the parallel structure of the method provides a faster training period. Light Gradient Boosting Machine (LightGBM) [4] was developed by Microsoft. It uses a smart sampling method that always retains large gradient instances and uses a portion of the small gradient instances. Category Boosting (CatBoost) [5] is another popular GBM algorithm that addresses the problems related to the categorical features in addition to the other enhancements in the algorithm.

Linear Regression models are one of the most essential models in statistics. Given an independent variable Y_t and dependent variables $F_{t,i}$, $i \in \{1...k\}$ it is desired to find a parametric linear function for Y_t given $F_{t,i}$ $i \in \{1...k\}$. Simply put, it is desired to find the best line such that we minimize least squares error, i.e. sum of squares of error terms. It has an algebraically closed form solution and may be obtained easily although requires matrix inversion. We have assumed data to be stationary on a small continuous interval closer towards the last data point. This assumption forms the basis of our Linear Regression model. Taking advantage of superior efficiency of training Linear Regression models, we have formed a grid search and chose models based on the desired evaluation metric WMAPE. Further issues, and approach are discussed in detail on the Regression Model section.

3 Approach

We have focused on two different approaches which have different feature sets and models. In addition, we provide a naive approach which is both a reliable benchmark and a good alternative to use in an ensemble model.

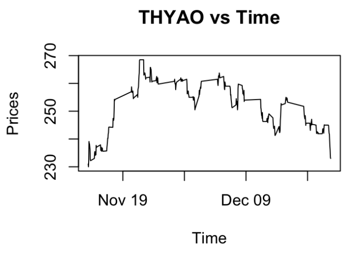
3.1 Naive Model

Naive model simply uses the previous day's closing price as the prediction of price in each hour for the stock of interest.

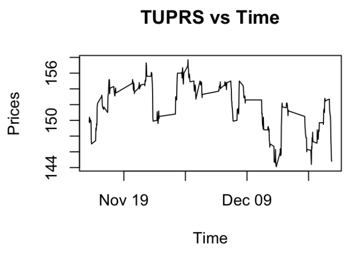
3.2 Regression Model

Linear regression models are widely applied in statistical contexts due to their explainability however they are models with high bias and low variance. Further, while evaluating the models and choosing a model WMAPE is to be utilized which renders the task a little more cumbersome since originally $R^2 = \frac{\text{SSR}}{\text{SSE}}$ is maximized. SSR is the sum of squared explained deviation and SSE is the sum of squared deviation from the mean. Therefore, we need to come up with an algorithm such that we obtain a selection of the model with respect to the desired metric. Although training is going to be performed through maximization of R^2, data is going to be splitted into two disjoint and consequent subsets which are respectively training and test samples. Since this is a time series data, test samples are going to be the last consecutive, say k, points in our data. After analyzing the ACF and PACF plots, it could be observed that a pure auto-regressive(AR) model with $p = 1$ could be utilized. Put another way, each point Y_t will have a feature Y_{t-10} that corresponds to the price of the stock on the subsequent day and the same hour. We have generated a feature that takes value $F_{1,t} = (Y_{t-50}/Y_{t-100} - 1)1_{t=Monday,9a.m.}$. It is similar to log percent change however log percent change has a steeper descent towards negative values since logarithm is a monotonically increasing function that maps $(0, 1) \to (-\infty, 0)$. The feature did not become statistically significant when auto-regressive component was present and thus was omitted for the rest of the work. It would definitely be interesting to transform the results to log-percent changes and then form our model, re-transform results into original form.

For the Linear Regression models, we have assumed that stock prices evolve according to an AR process. That being said, it is required that at least weak stationarity must hold meaning mean should be fixed, variance should be stable and correlation information should not be changing with respect to time for lags. Further, there are at total 30 stocks meaning while forming a forecast for a stock it is possible to extract useful information from the remaining 29 stocks. By taking advantage of the ease of training a linear regression model, it is possible to train 29 models for each stock and selecting the one that yields the lowest WMAPE value. However, validation is somewhat complicated in the sense that data is a time series data. Therefore, one cannot perform arbitrary



(a) THYAO



(b) TUPRS

Figure 4: THYAO and TUPRS recent prices

test-train splits. Also, data is not stationary meaning we have to restrict ourselves with a closer time interval towards the last available time point without performing transformations such that stock prices become stationary all over the horizon. To achieve this goal, we have restricted ourselves with the time period beginning from 13/11/2023 9 A.M. In Figure 4, it may be observed that as we get closer to the last time point the effect of trend vanishes between two distant points. However, it should be noted that this may not and is not the case for all the stocks. This is an assumption and may be argued that could be unnecessarily strong for some stocks. The assumption holds for most of the stocks and that is of concern here. Moreover, there are certain tests such as Kwiatkowski-Philips-Schmidt-Shin(KPSS) that may be utilized to statistically test whether the trend information is stable for a time series on a given interval. Long story short, stocks are assumed to be stationary on a shorter interval towards the last data points which is an acceptable assumption although may be invalidated for certain stocks.

3.2.1 Model Selection

Additionally, taking additional advantage from the ease of training a linear regression model we form 25 to 30 models for each stock by sliding the training window one day to the right each time. Therefore we select one model from 725 to 870 models for each stock considering the best WMAPE performances on each stock. To admit, there is an erroneous approach at this point which was realized at the submission period. The models seem to be an over-fit and not capable of generalization due to the fact that last time point's WMAPE was taken basis while choosing the model. There should be a longer testing period such that the models have generalization capability. It is statistically proven that $MSE(\hat{\theta}) = bias^2(\hat{\theta}) + Var(\hat{\theta})$. Utilizing this approach, although MSE(Mean Squared Error) is not the desired error metric, variance is larger due to the loss of generalization capability which conflicts the innate advantage of the linear regression model. To sum up, although certain advantages of the linear regression models were utilized due to failure of considering the natural characteristics of the linear regression models this approach yielded somewhat larger and generally undesirable WMAPE values which may be observed on Table 2.

All the functions are coded by hand, only the base function for training a linear regression model is utilized with certain data structure packages such as data.table.

3.3 LightGBM Model

Gradient Boosting Machines (GBMs) are well-known models that are highly used for forecasting tasks. LightGBM is also a popular and lightweight alternative among the GBMs.

In the first step, we trained the model with a set of features and default hyperparameters. After the first week of submissions, we conducted cross-validation trials in order to find better hyperparameters and test the new features. Then we compared the initial model with the new model in terms of their WMAPE in the first week of the forecast period. Since the new implementations did not contribute to the model performance significantly, we continued with the initial model for the rest of the submissions.

3.3.1 Feature Engineering

We have initialized our model with the features below:

- Time features
  - *month*: Month of the year for the specific day. The value range is between 1 and 12. It helps model to capture a year-based seasonality if exists.
  - *monthday*: Day of the month. The value range is between 1 and 31. It is used to obtain intra-month seasonality.
  - *weekday*: Day of the week. Takes values between 0 and 4 for the 5 days of week. Week days might have a significant impact on the price changes especially in first and last days of the stock exchange sessions. In addition, some specific days of week might include some events that affect the market. For instance, Money Market Board (PPK) announces the policy interest rates on Thursday, which leads some critical price movements in overall.
  - *hour*: Session hour of the specific timestamp. It indicates the nth hour of the session and takes values between 1 and 10. Price movements might differ among the hours of the exchange session.
- Stock price change features

  - *change_lag_exact*: Price change of the specific stock for the previous days in exact same hour. It is a set of features calculated for last 20 days available. It is an important set of features since it contains stock specific information about the price.
  - *change_lag_prev*: Price change of the specific stock for the previous days in the previous hour. It is a set of features calculated for last 20 days available.
  - *change_lag_next*: *change_lag_prev*: Price change of the specific stock for the previous days in the next hour. It is a set of features calculated for last 20 days available.

- Stock identification feature
  - *starting_price*: Price of the stock at the first timestamp of the data. The value of the feature is the same for all rows of the same stocks. With this feature, it is intended to distinguish the stocks from each other. It is more useful than using binary columns for each stock.

- Stock exchange features
  - *change_mean_lag*: Geometric average of change of the prices from start of the day of all stocks. Its lag is used as feature in order to prevent data leakage. The feature gives model a hint about the overall price movements in the market.

The features indicated above solely depends on the data provided for the project. We also tested some models with additional features which were provided from external data sources such as:

- Macroeconomic features
  - *usd_close*: Close price of USDTRY. The level of price might have a minor contribution to the model.
  - *usd_daily_change*: Daily change of USDTRY in previous day.
  - *usd_weekly_change*: Weekly change of USDTRY in previous day.
  - *usd_monthly_change*: Monthly change of USDTRY in previous day.
  - *avg_int*: Weekly average interest rate announced by the Central Bank

- Stock volume features
  - *volume_try*: Transaction volume of the stock in the previous day in terms of Turkish Liras.
  - *volume_try_daily_change*: Daily transaction volume (TRY) change of the stock in the previous day.
  - *volume_try_weekly_change*: Weekly transaction volume (TRY) change of the stock in the previous day.

  - *volume_try_biweekly_change*: Biweekly transaction volume (TRY) change of the stock in the previous day.
  - *volume_try_monthly_change*: Monthly transaction volume (TRY) change of the stock in the previous day.
  - *volume_usd*: Transaction volume of the stock in the previous day in terms of US Dollars.
  - *volume_usd_daily_change*: Daily transaction volume (USD) change of the stock in the previous day.
  - *volume_usd_weekly_change*: Weekly transaction volume (USD) change of the stock in the previous day.
  - *volume_usd_biweekly_change*: Biweekly transaction volume (USD) change of the stock in the previous day.
  - *volume_usd_monthly_change*: Monthly transaction volume (USD) change of the stock in the previous day.

However, these features could not provide a significant contribution in terms of prediction performance.

We used price changes as the output of the model instead of the prices directly. The reasons are summarized below:

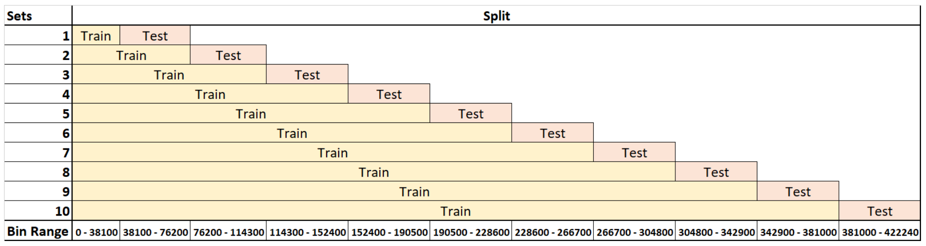
- We use a single model to make predictions for all of the stocks. Each stock has different price levels which might create a difficulty for the model to output sensible results. Even if we use a different model for each stock, the price level of that single stock might also significantly change over time.
- We know that Borsa Istanbul has a quota on price changes. The price of a stock cannot change more than 10% in both directions compared to its daily opening price. It creates a natural boundary for our output. Since the tree-based models have difficulty when they are required to forecast out of the training domain, this boundary makes the tree-based models more applicable.

Due to the reasons above, the change in the price according to the previous day's closing price is used as output for each hour of each stock.

3.3.2 Model Validation

Model validation were made according to a backtesting approach. Backtesting is widely used with time-series forecasting tasks since it prevents data leakage from the future instances which might be a problem in standard k-fold cross-validation approach. We have partitioned the available data into 10 train-test splits as shown in Figure 5. In each split, we used a limited number of test instances that are close to the training period since the absence of recent data might deteriorate the model performance.

The backtesting cross-validation was applied to find the most appropriate LightGBM parameters to construct the model. In order to determine the best



Sets	Split										
1	Train	Test									
2	Train	Train	Test								
3	Train	Train	Train	Test							
4	Train	Train	Train	Train	Test						
5	Train	Train	Train	Train	Train	Test					
6	Train	Train	Train	Train	Train	Train	Test				
7	Train	Train	Train	Train	Train	Train	Train	Test			
8	Train	Train	Train	Train	Train	Train	Train	Train	Test		
9	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test	
10	Train	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test
Bin Range	0 - 38100	38100 - 76200	76200 - 114300	114300 - 152400	152400 - 190500	190500 - 228600	228600 - 266700	266700 - 304800	304800 - 342900	342900 - 381000	381000 - 422240

Figure 5: Backtesting Sets

parameters, we used average WMAPE in the test period as our cost function. Here are the parameters whose best values were sought:

- *reg_alpha*: It is the L1 normalization parameter for the GBM algorithm. It has been search in a continuous interval between 10^{-8} and 10^{1}.
- *reg_lambda*: It is the L2 normalization parameter for the GBM algorithm. It has been search in a continuous interval between 10^{-8} and 10^{1}.
- *num_leaves*: Maximum number of leaves in one tree. It has been chosen between 2 and 256
- *feature_fraction*: Fraction of features to consider in each split. It has been selected between 0.1 and 1.
- *bagging_fraction*: Fraction of instances to consider in each split. It has been selected between 0.1 and 1.
- *bagging_freq*: It indicates the number of iterations between each bagging operation. It is between 1 and 7.
- *min_child_samples*: Minimum number in a leaf. It has been selected between 5 and 100.

Since we have such a large search space, we cannot deal with hyperparameter tuning using a brute-force search algorithm. We used a package called "*optuna*" which provides a more elegant search based on the previous trials in the search history. The algorithm searches for different parameters in each iteration. We run the algorithm for 500 iterations. In return, we obtained the best parameters which are shown below:

- *reg_alpha*: 9.963560916189403
- *reg_lambda*: 2.0970249799778635
- *num_leaves*: 34
- *feature_fraction*: 0.7545810668686815

- *bagging_fraction*: 0.10000380384529119
- *bagging_freq*: 7
- *min_child_samples*: 7

The best parameters lead to a test cost of 0.01581. The default parameters of LightGBM provide 0.02034 for the same test sets.

We conducted the model validation after the first week of the submissions. We included the extra features mentioned in the previous section. Since we have one week of results for the actual submissions, we wanted to compare the results for this period.

Table 1: New Model Comparison

Date	**New Model**	**GBM-1**	**Relative Decrease**
12/26/23	0.0136	0.0127	0.0704
12/27/23	0.0167	0.0145	0.1507
12/28/23	0.0174	0.0159	0.0942
12/29/23	0.0136	0.0160	-0.1531
1/2/24	0.0149	0.0127	0.1676
1/3/24	0.0134	0.0173	-0.2269
1/4/24	0.0134	0.0157	-0.1429
1/5/24	0.0109	0.0110	-0.0128
	Average Decrease:		-0.0066

As shown in Table 1, the new model with hyperparameter optimization and new features does not contribute to the test performance significantly. Thus, we discarded the new model from further discussions.

4 Results

4.1 Model Performances

Table 2 shows the WMAPE of our models during the forecasting period. Unfortunately, Borsa Istanbul began with an unexpectedly huge decline on the first day of the forecasting period. Thus, all models performed below the average compared to the rest of the dates. The GBM model has the lowest average test performance which makes it preferable. However, the naive model performs similarly to the GBM model. We generally submitted an average of the naive and GBM models.

4.2 Submission Performance

Table 3 shows the performance of our submitted results compared with other submissions. Although we were not on the top row on all days, our submitted results have the lowest average WMAPE and rank.

Table 2: Model Performances

Date	**Niave**	**GBM**	**Regression**
12/25/23	2.85%	3.29%	3.93%
12/26/23	2.85%	1.27%	4.11%
12/27/23	1.40%	1.45%	2.02%
12/28/23	1.63%	1.59%	2.24%
12/29/23	1.33%	1.60%	1.76%
1/2/24	1.55%	1.27%	2.16%
1/3/24	1.51%	1.73%	1.79%
1/4/24	1.58%	1.57%	1.61%
1/5/24	1.03%	1.10%	2.15%
1/8/24	2.31%	1.85%	3.58%
1/9/24	1.60%	1.76%	2.93%
1/10/24	1.52%	1.51%	1.81%
1/11/24	1.09%	1.16%	1.74%
1/12/24	1.49%	1.48%	1.97%
Average	1.70%	1.62%	2.42%

Table 3: Submission Performance

	WMAPE	**Rank**
count	11	11
mean	7.74%	5.68
std	11.86%	1.12
min	1.68%	4.01
25%	1.94%	4.88
50%	2.58%	5.43
75%	3.70%	6.69
max	34.98%	7.11
Our Submission	1.68%	4.01

5 Conclusions and Future Works

For the given stock price prediction task, we have conducted the necessary data analyses and proposed different approaches including feature engineering and model development.

We used the initial GBM model in the first week and then tried to empower the models with further feature engineering and hyperparameter optimization techniques. Since these efforts did not bring a significant difference in model performance, we continued with our first model. In our submissions, we used the naive approach, the GBM model, and the average of them mostly since they performed above-average compared to the other teams in the leaderboard.

The GBM model was the most successful among our model approaches. On the other hand, the naive approach performs better from time to time. We used a simple average of their outputs to submit. A smarter approach to mixing the models might provide better outputs. Conducting an extensive analysis to find which model is superior in which situations (date, stock, etc.) might help us to output better results.

A future study might benefit from analyzing the effects of external data in a more detailed way. For instance, although we used transaction volume data, we did not use any intra-day metric related to this feature. In addition, macroeconomic indicators such as currency and interest rates might provide better results if they are investigated more. They might be useful to remove the general market effect on the stock prices. Thus, one can utilize the stock-related features in a much better way after de-trending the data.

Moreover, Linear Regression model could greatly benefit through a more extensive validation approach since as discussed on the section 3.2.1 the performance of the Linear Regression model was deteriorated through poor validation approach. Further, there may be additional information gain through utilization of more than one different stock prices' lag values while trying to form forecasts of a stock price. There may be a stochastic grid search performed since dimension grows exponentially as one tries to utilize more than one different stock. Also, one might try to transform data utilizing Box-Cox, power or logarithmic transformations as well as trying to predict the changes. These procedures are capable of yielding a stationary data structure which aids one on modelling the stochastic behavior in a more meticulous way. Re-stated, the main handicap associated with this study was that stock prices are manipulated through exogenous factors frequently and is affected constantly by the political changes. These property requires one to provide model with necessary variables such that model captures the sudden spikes in behavior and variance information.

6 Code

Here is the list of code files:

- Regression Model

- Initial GBM model (GBM-1)
- GBM Enhancement Notebook (Cross-Validation, Hyperparameter Tuning, Additional Feature Engineering)

References

[1] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[2] Yoav Freund and Robert E Schapire. “A desicion-theoretic generalization of on-line learning and an application to boosting”. In: *European conference on computational learning theory*. Springer. 1995, pp. 23–37.

[3] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[4] Guolin Ke et al. “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30 (2017).

[5] Liudmila Prokhorenkova et al. “CatBoost: unbiased boosting with categorical features”. In: *Advances in neural information processing systems* 31 (2018).