Capstone Project

Martin Piják

Bratislava, Slovakia

Abstract

This is project for using Machine Learning (ML) methods in futures trading.

Keywords: Machine Learning, Trading, Udacity, Nanodegree

1. Domain

This project investigates possible futures trading strategies on Chicago Mercantile Exchange (CME) and Intercontinental Exchange (ICE) markets with Machine Learning methods. The goal is to find the trading strategy mostly based on the price, Commitment of Traders report (COT) and seasonality pattern. We will compare this trading strategy to commonly used investing approaches as returns of Nasdaq and fixed returns of interest rate 4%. Fixed rates of 4% is approximation of Microsoft (MSFT) **bonds** yield. We have chosen this benchmark in search of better yield than government bonds but still very secure.

Example of Machine Learning used in futures algorithmic trading.

- Algorithmic Trading of Futures via Machine Learning
  - In this article you can find details about feature engineering, ML-algorithm selection and training
  - Input vector is 2 years of price and volume data.
- A Machine Learning framework for Algorithmic trading on Energy markets
  - This article deals with general pipeline setup for trading

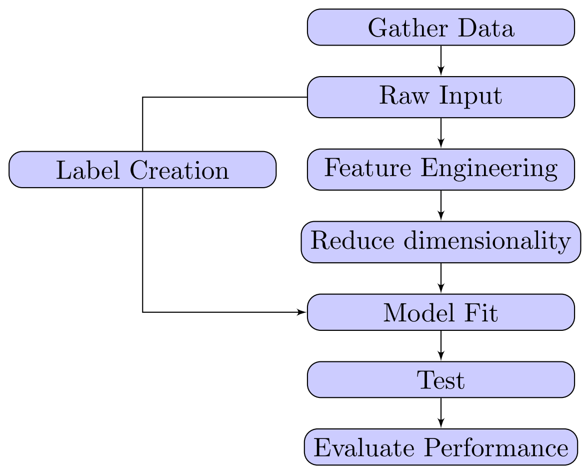
Following commodities were investigated:

- Gold
- Corn
- Coffee

2. Project Overview

2.1. Pipeline

Following schema is showing high level project overview.



2.2. Tools

Following tools were used:

- General ML tools
  - Scikit
- Visualization
  - matplotlib
  - seaborn
- Model building tools
  - Keras
  - TensorFlow (keras backend)
  - H2O
  - LightGBM

2.3. Evaluation Metrics

Model performance will be evaluated on the trading data from year 2018. Based on the trading signals trade will be simulated. Then gains/losses will be accounted.

$$\delta = 0.95 \tag{1}$$
$$fee = 1.5 \tag{2}$$
$$account = 10000 \tag{3}$$
$$f(model_{signal}) \implies \begin{cases} -1 \implies \begin{cases} high - open >= stop_{loss} = -stop_{loss} - fee \\ otherwise = (open - settle) * \delta - fee \end{cases} \\ 0 \implies 0 \\ 1 \implies \begin{cases} open - low >= stop_{loss} = -stop_{loss} - fee \\ otherwise = (settle - open) * \delta - fee \end{cases} \end{cases} \tag{4}$$
$$model(inputs) = model_{signal} \tag{5}$$
$$evaluation = account + \sum^{2018} f(model_{signal}) \tag{6}$$

Note: if throughout the evaluation of model trading (expression 6) we reach 0 or less we stop trading since we have no more capital. Minimum of evaluation is 0 or slightly negative number. Bigger the evaluation number the better model performs. Also note that this evaluation metrics is not directly connected to the training of the model. We are searching for model parameters and loss functions that will maximize this evaluation metrics.

2.4. Where is the code?

All the data and code that was used to generate this report can be found in the **trading_project.ipynb** notebook. When training neural network with keras you can arrive at slightly different results than those in this report because of different seed.

3. Datasets and Inputs

We used data from Quandl. Data contains Open, High, Low, Close and volume (OHLCV) and commitment of traders (COT). Continuous data was generated by taking contract with the highest volume for the trading day. Sample of the data is below. For details please see **data-preparation.ipynb**.

Raw gold OHLCV data example (with calculated label).

Index	Open	High	Low	Settle	Volume	Prev. Day OI	target
2006-06-13	590.5	595.0	565.5	566.8	93899.0	192616.0	0.0
2006-06-14	570.0	575.5	565.4	566.5	68729.0	192917.0	0.0
2006-06-15	573.5	579.5	569.5	570.3	52628.0	193887.0	0.0
2006-06-16	581.2	582.5	570.5	581.7	43947.0	189585.0	0.0
2006-06-19	572.8	578.4	571.0	572.4	27362.0	189168.0	1.0

Raw gold COT data example.

Index	2006-06-20
Open Interest	390281.0
Producer/Merchant/Processor/User Longs	47440.0
Producer/Merchant/Processor/User Shorts	126992.0
Swap Dealer Longs	22404.0
Swap Dealer Shorts	64682.0
Swap Dealer Spreads	25806.0
Money Manager Longs	94632.0
Money Manager Shorts	30963.0
Money Manager Spreads	48730.0
Other Reportable Longs	32947.0
Other Reportable Shorts	11142.0
Other Reportable Spreads	64458.0
Total Reportable Longs	336417.0
Total Reportable Shorts	372774.0
Non Reportable Longs	53864.0
Non Reportable Shorts	17507.0

4. Data analysis

In the proposal I wanted to investigate classification of volatility. Below is 95% of daily volatility. 95% was selected as a simulation of slippage. General trading idea is to keep trade for a day.

4.1. Label Generation for classifier

First let's have a look at the volatility of different commodities. All the data below is gathered from training set. We have put aside performance evaluation set (trading data for year 2018).

Commodity	count	mean	std	min	25%	50%	75%	max
Gold	2921	-6.88	1225.72	-11390.5	-560.5	28.5	627.0	6555.0
Corn	2917	6.12	411.07	-1983.13	-190.0	11.88	213.75	1888.13
Coffee	2908	-11.55	1070.46	-5236.88	-516.56	0.00	498.75	6341.25

Labels were generated based on the trading target. Label -1 is for short trade 0 for no trade and 1 for long trade.

Threshold was chosen for different commodities differently based on the overall gains of the next day considering whole contract.

Commodity	Gains target (USD)
Gold	500
Corn	150
Coffee	375

$$fee = 1.5 \tag{7}$$

$$v_{volatility} = (\text{close} - \text{open}) * \delta \text{ where } \delta = 0.95 \tag{8}$$

$$labels \begin{cases} |v_{volatility}| > t_{threshold} + fee \implies \begin{cases} v_{volatility} > 0 \implies 1 \text{ (long)} \\ v_{volatility} < 0 \implies -1 \text{ (short)} \end{cases} \\ |v_{volatility}| < t_{threshold} + fee \implies 0 \text{ (no trade)} \end{cases} \tag{9}$$

δ constant is used for simulating slippage.

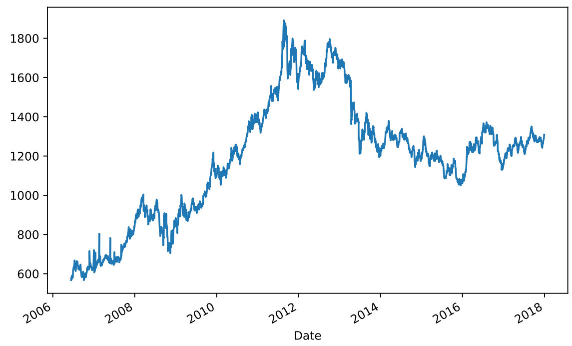
We decided to have a look at regressor as well.

Based on the past 2 years of trading data (OHLCV and COT) classifier is deciding whether to trade or not.

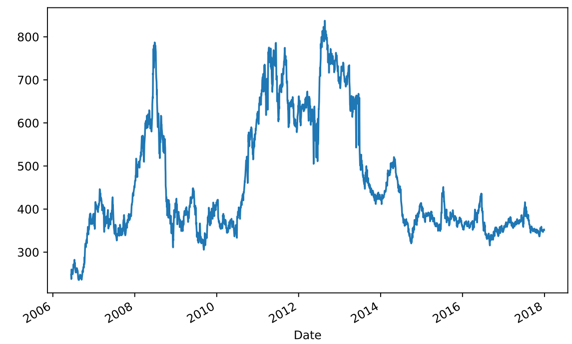
Approach to training classifier/regressor and evaluation does not take into account that stop-loss can still make trade unsuccessful. From Machine Learning perspective the best approach is to have result of classification regression as close to the desired outcome as possible.

Stop-loss was selected so that 90% of trades will be successfully executed (exited on close).

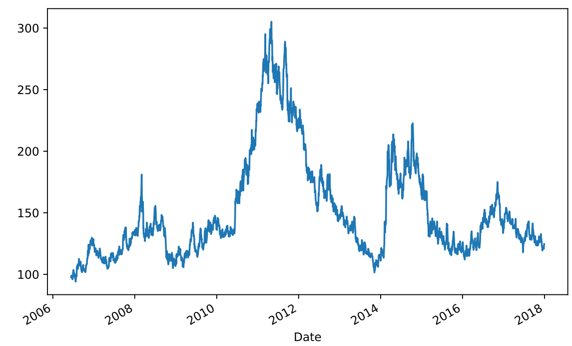
Commodity	Long Stop Loss	Short Stop Loss
Gold	1000	800
Corn	300	300
Coffee	800	800



(a) Gold

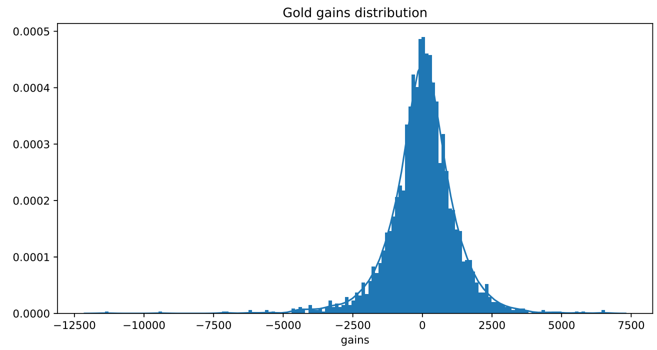


(b) Corn

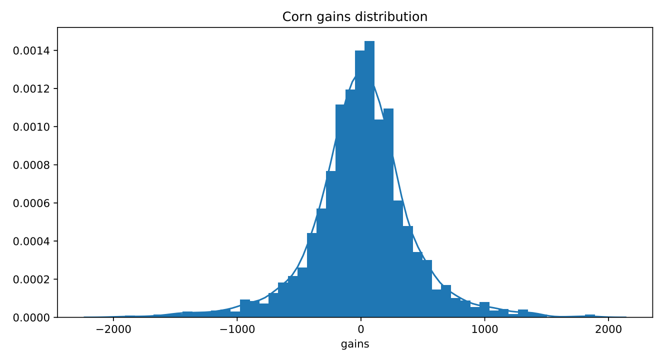


(c) Coffee

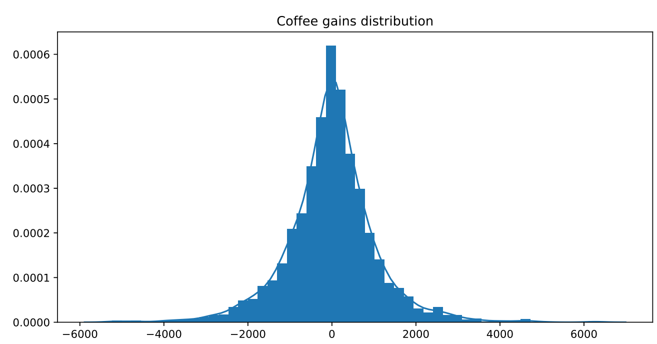
Figure 1: Close price graphs



(a) Gold

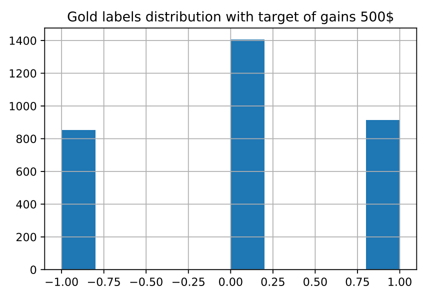


(b) Corn

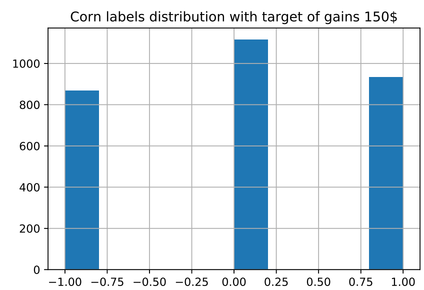


(c) Coffee

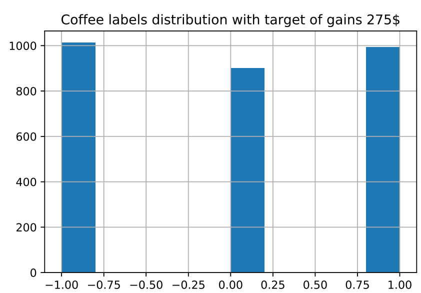
Figure 2: Gains distribution



(a) Gold

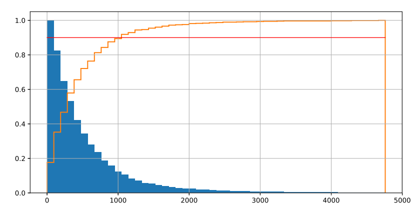


(b) Corn

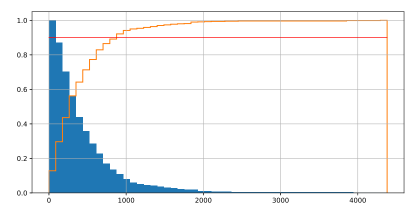


(c) Coffee

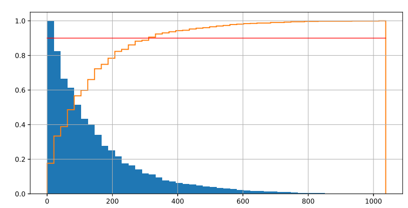
Figure 3: Labels distribution



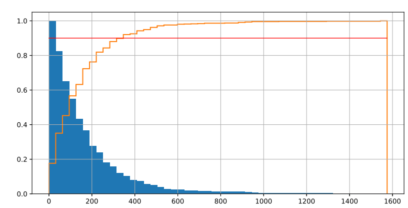
(a) Gold long



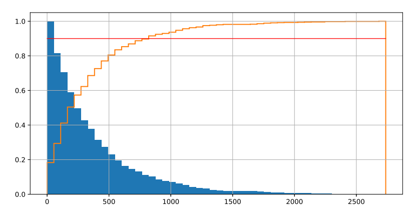
(b) Gold short



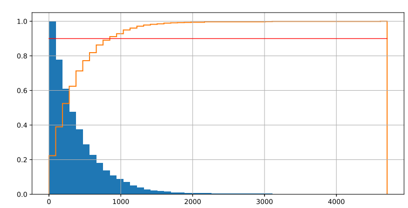
(c) Corn long



(d) Corn short



(e) Coffee long



(f) Coffee short

Figure 4: Stop Loss analysis counter position move

5. Feature Engineering

We transformed last $2 * 252$ trading days into vector with 2520 scalars.

To capture cyclicality, we have transformed features as trading day of month, day of week and quarter of year into *sin* and *cos* values. I am not sure, whether this transformation grants subsequent PCA usage. After adding data features vector now contains 2528 scalars.

We transformed COT to reflect extreme in the index. COT 1 of industrials corresponds to the maximum of traders' positions throughout the last 2 years. Respectively 0 to minimum. We have included COT for commercial (Money managers) and industrial (producers) users from the last 8 weeks which is 16 values. Training vector now contains $2528 + 16 = 2544$ values.

5.1. COT transformation

See below sample of feature engineered COT data.

Index	Prod_net_position_perc	Money_manager_net_position_perc
2006-06-27	1.000000	0.000000
2006-07-03	0.919154	0.246529
2006-07-11	0.792600	1.000000

5.2. PCA

We try to capture about 80% of variance of the data. For corn this corresponds to about 160 components. For gold it is about 180 components. Case of coffee is very strange.

It is interesting Corn is best explained by PCA transformation. It is probably due to clear seasonal patterns in trading.

I am surprised that gold is better explained by PCA transformation than Coffee. I would expect that coffee has stronger seasonal trading patterns than gold because of the growth cycle. Maybe gold mining is subject to the weather in similar way as, agricultural commodities. Gold is still mostly recycled and new production has limited impact on total amount of traded gold.

Possible explanations for lack of variance explanation by PCA for coffee:

- corn traded on CME is mostly US produced with stable harvest season
- production of coffee is very unpredictable depending on the conditions of a given year
- there are multiple producers around the world (coffee is more of a global market with limited US production) with different harvest periods Coffee Harvest
- important difference between coffee and corn is price per unit corn is much less efficient to transport
  - 1 kg of corn is worth about 15¢

– 1 kg of coffee is worth about 230¢

Based on the PCA variance graph I think that PCA transformation is not suitable for coffee. Information in components is growing linearly. If we don't see sharp increase of cumulative explained variance with few first components, then PCA transformation is not suitable. Therefore, coffee should not be considered for trading with PCA transformation. I will continue with coffee as well but based on this transformation I would not go ahead with trading unless I would find different transformation.

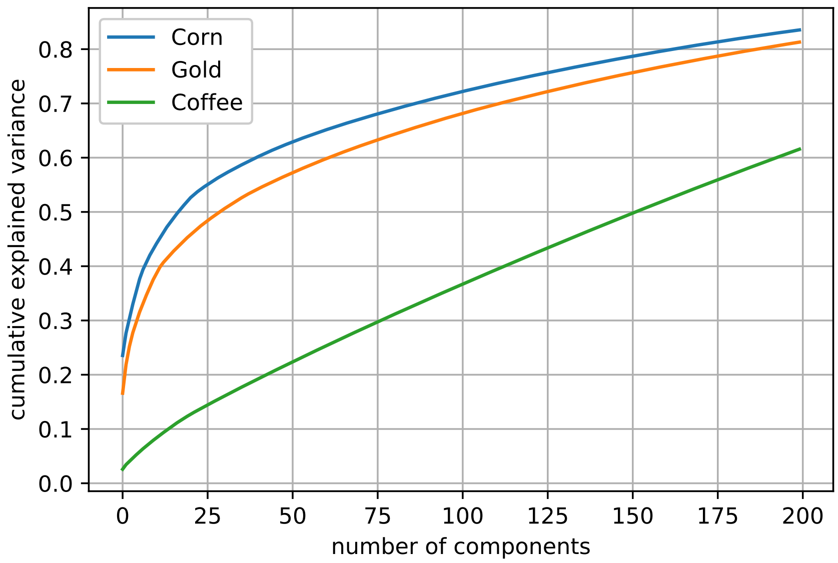
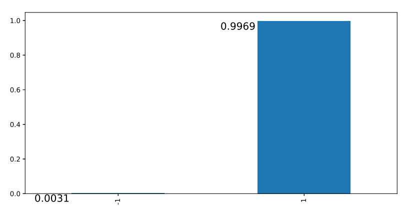


Figure 5: PCA explained variance

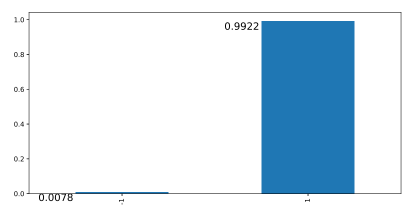
5.3. Outliers

We checked dataset for outliers. I used IsolationForest from scikit with automatic outlier detection after PCA transformation. Data was split into 3 sets training, validation and performance check. Training set contains more that 1930 samples. On this training set outlier detection was trained. Following number of outliers were detected.

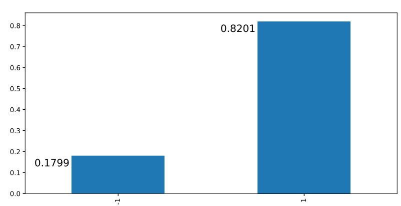
Commodity	Number of Inliers	Number of Outliers
Gold	1927	6
Corn	1915	15
Coffee	1577	346



(a) Gold



(b) Corn



(c) Coffee

Figure 6: Outliers histogram

6. Model

6.1. Model Selection

Gradient boosting on decision trees can have better performance than standard random forest. There can be found some reference to training models with XGB (**Quantisti article**). In this project we will use different type of gradient boosting called LightGBM. Training with gradient boosting is different because next tree we construct is trying to correct for the error. You can read more about GBT vs Random Forest (**article**).

Problem of random forest (as well as gradient boosting) is that this kind of algorithm does not extrapolate well (**freerangestats article**). Neural networks do not suffer from this problem. We will try neural networks for regression.

6.2. General strategy for training

We tried the same model for all three commodities (gold, corn and coffee). We are looking for classifier with the same parameters for any commodity. This way we can be sure that we have found some general classification and results are not just a coincidence for a given model. Generally, we try to find one model that works for most markets.

Following exploration can be split into:

- classifier
- regressor

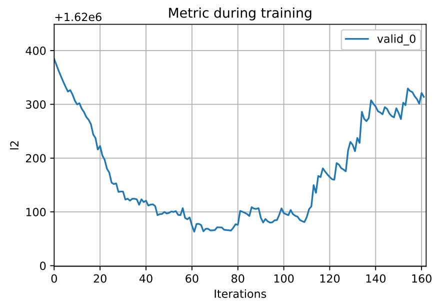
After model training all the models will be validated on the year 2018 trading data. Trading signals from model will be used for simulated trading. Open question is in case of regressor where the threshold for trading should lie? Empirically I chosen to trigger at most around 80 trades.

6.3. LightGBM regressor

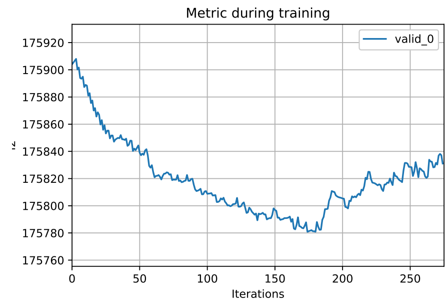
6.3.1. Parameters

We used following parameters to train LightGBM regressor.

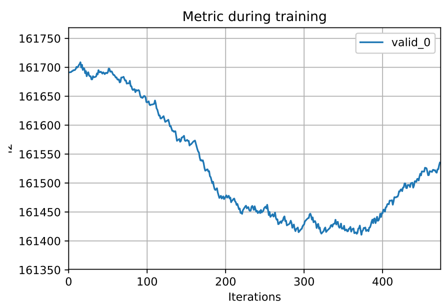
Parameter	Value
num_leaves	40
objective	regression
boosting	dart
metric	l2 (mean squared error)
estimators	1000
learning_rate	0.001
num_class	1
max_bin	30
reg_alpha	5
reg_lambda	10
num_round	1000
early_stopping_rounds	100



(a) Gold



(b) Corn



(c) Coffee

Figure 7: Training Error on The Validation Set

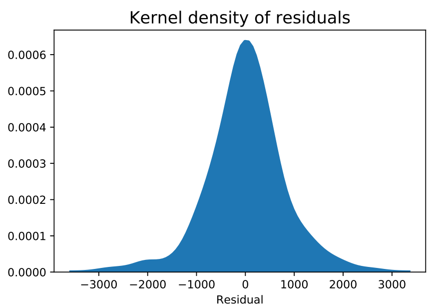
6.3.2. Training

When determining trading performance of the LightGBM regressor we have chosen following trading thresholds.

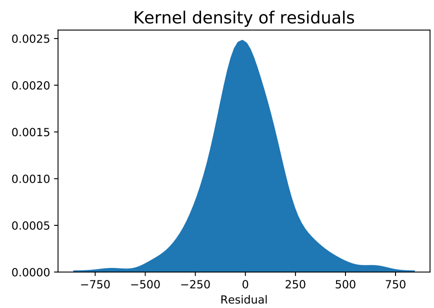
Commodity	Trading threshold	Classifier threshold	Ratio
Gold	10	500	$\frac{1}{50}$
Corn	10	150	$\frac{1}{15}$
Coffee	10	375	$\frac{1}{38}$

This part of trading process is somewhat unknown to me. Regressor is not predicting many days with big volatility. Predictions are much flatter than the actual volatility.

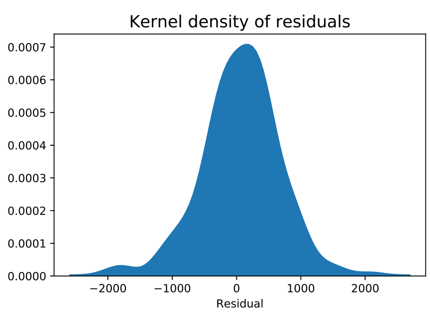
Based on the chosen threshold you can get some idea of how well is the regressor approximating actual volatility.



(a) Gold

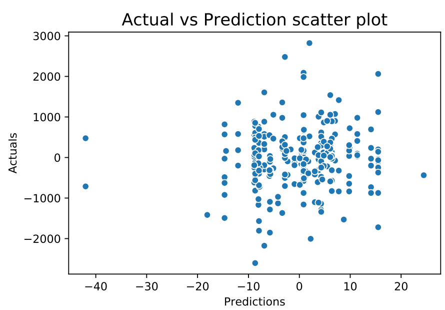


(b) Corn

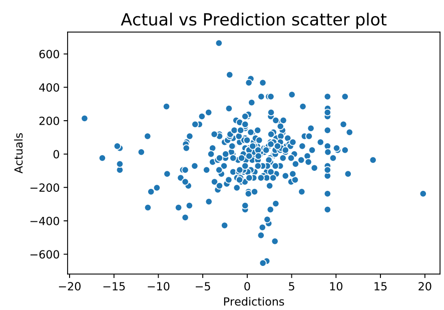


(c) Coffee

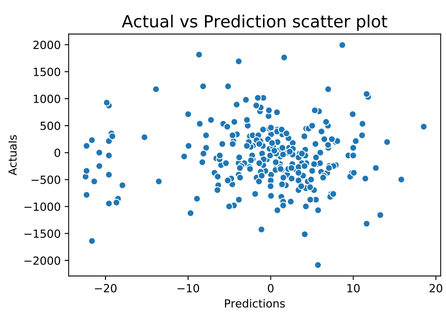
Figure 8: Residual Errors on The Validation Set



(a) Gold



(b) Corn



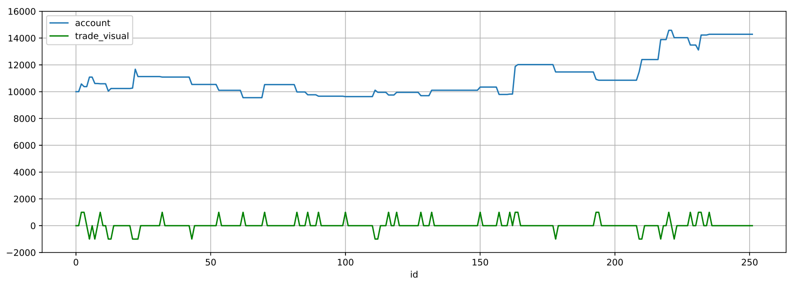
(c) Coffee

Figure 9: Actual vs Prediction Scatter Plot

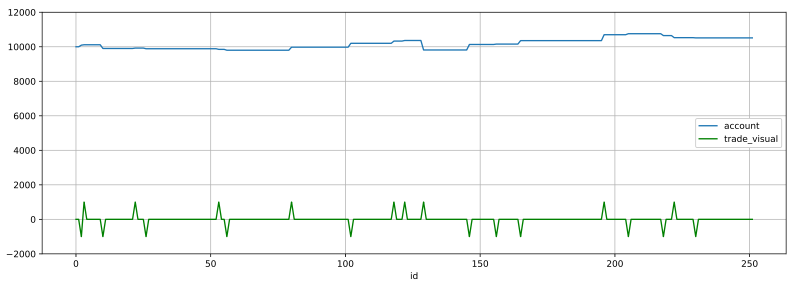
6.3.3. Trading performance

Commodity	Prediction/actual result correlation	Gains
Gold	0.079	42%
Corn	0.078	5%
Coffee	-0.001	62%

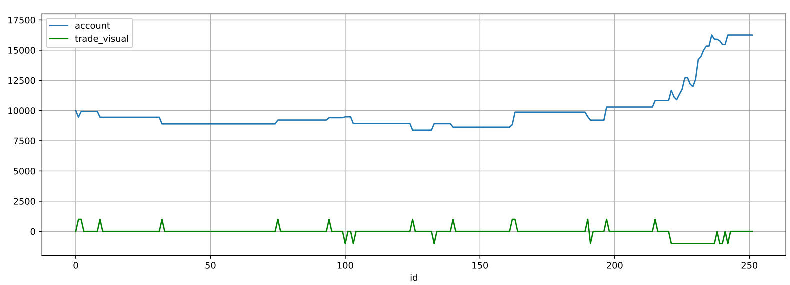
We evaluated trading performance on the year 2018. Below is account performance and histogram of trades.



(a) Gold

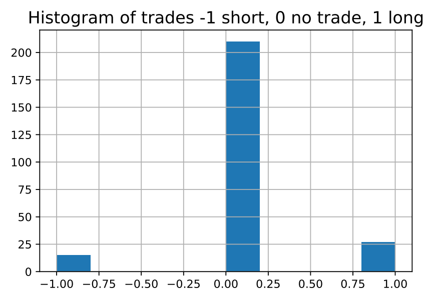


(b) Corn

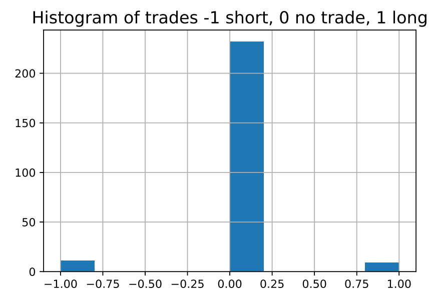


(c) Coffee

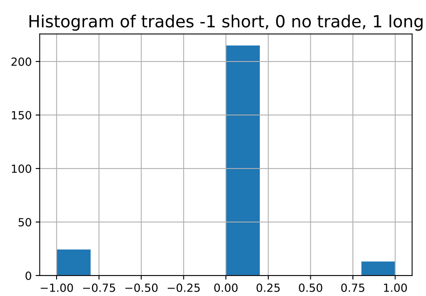
Figure 10: Account performance



(a) Gold



(b) Corn



(c) Coffee

Figure 11: Trade histogram

6.4. Neural Network Classification

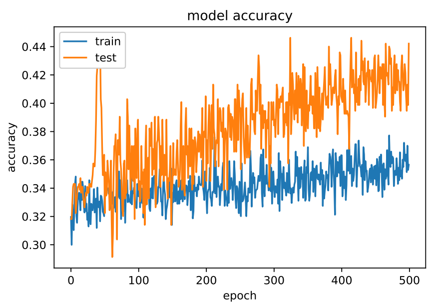
6.4.1. Neural network design

Neural network is fully connected with input layer, 2 hidden layers and output layer consisting of 3 neurons. Output layer of 3 layers represent 3 categories $\{-1, 0, 1\}$.

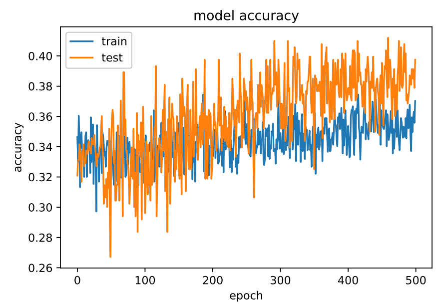
Input	input layer	hidden layers		output layer	output
200	32	16	8	3	3

6.4.2. Classifier Training

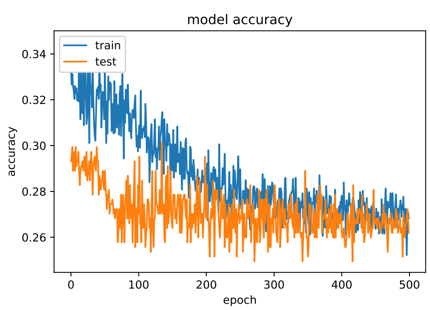
In the training setup we used training data with labels described above. If validation loss improved then new weights were saved. Total number of epochs was 500 and batch size 20.



(a) Gold



(b) Corn



(c) Coffee

Figure 12: Accuracy during training

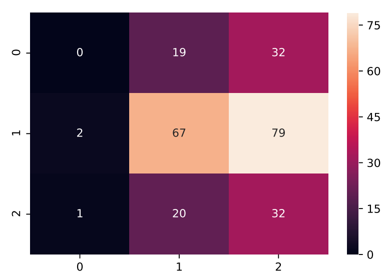
6.4.3. Performance Evaluation

Results below were obtained by running classifier on trading data from year 2018.

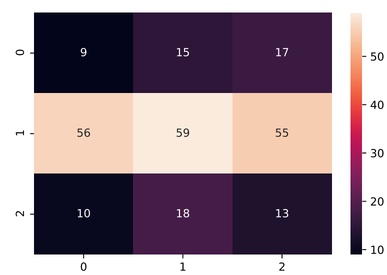
Commodity	Gains
Gold	68 %
Corn	-40 %
Coffee	0 % (no trade)

Precision Score:

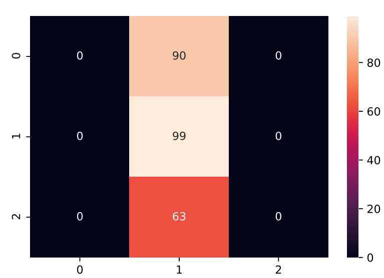
Commodity	Short	No trade	Long
Gold	0.	0.63	0.22
Corn	0.12	0.64	0.15
Coffee	0.	0.39	0.



(a) Gold

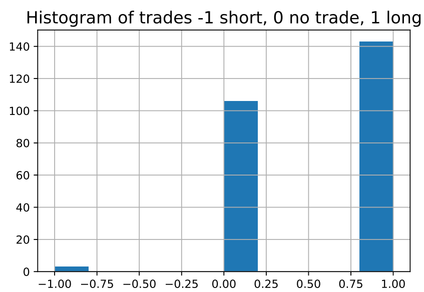


(b) Corn

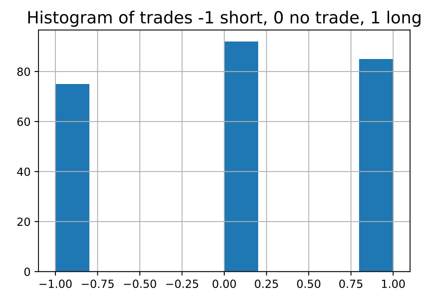


(c) Coffee

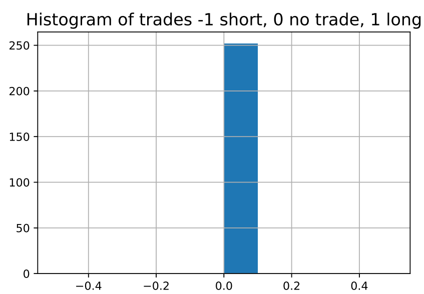
Figure 13: Confusion Matrix



(a) Gold

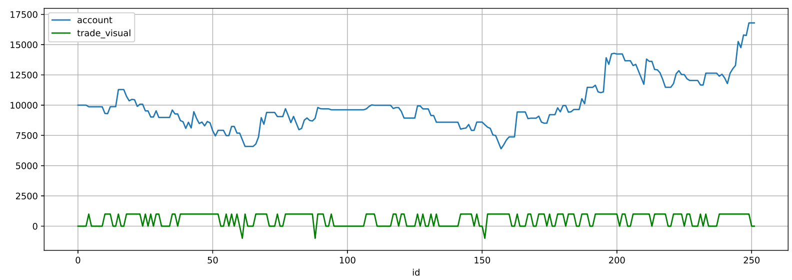


(b) Corn

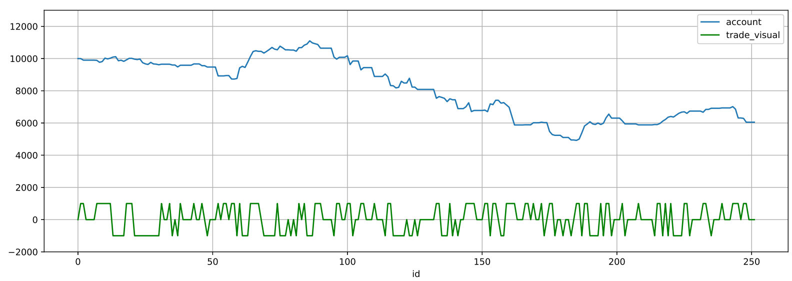


(c) Coffee

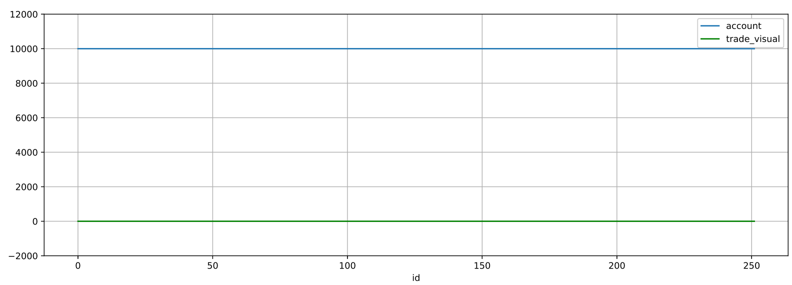
Figure 14: Trade histogram



(a) Gold



(b) Corn



(c) Coffee

Figure 15: Trade histogram

6.5. Neural Network Regressor

6.5.1. Neural Network Design

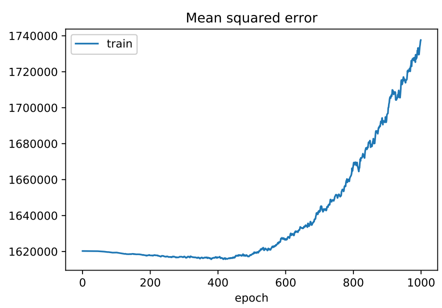
Neural network is fully connected with input layer, 3 hidden layers and output layer consisting of 1 neuron.

Input	input layer	hidden layers			output layer	output
200	500	128	64	32	1	1

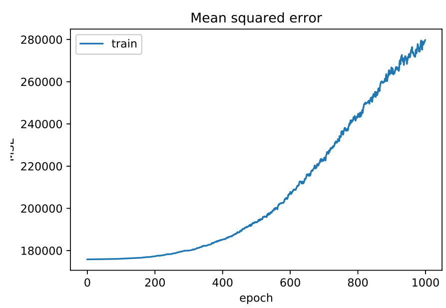
6.5.2. Regressor Training

In the training setup we used training data with labels described above. If validation loss improved then new weights were saved. Total number of epochs was 1000 and batch size 20.

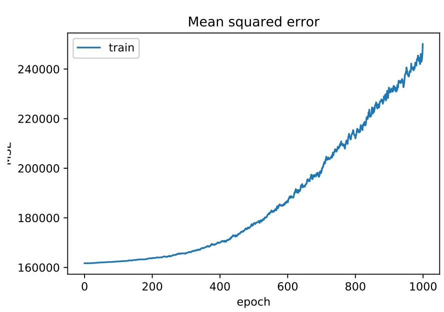
During training only in case of gold we can see improvement of loss on the validation set. For details see validation loss (MSE) below.



(a) Gold



(b) Corn



(c) Coffee

Figure 16: Mean Squared Error

7. Benchmarking Model

We used three comparisons to model performance.

- fixed percentage of 4% with monthly interest calculation (bank/bond deposit comparison)
- Nasdaq performance
- mean reversal trading on given commodity markets with mean of 20 days and mean of 10 days

It is interesting to notice that mean reversal strategy works for gold and corn but does not produce gains for coffee market.

For each commodity we have selected best model out of three trained.

Commodity	Model
Gold	Neural Network Regressor
Corn	LightGBM Regressor
Coffee	LightGBM Regressor

We have taken into consideration not only gains (even though important) but also, other metrics. Out of three commodities gold has best correlation between predictions and actual volatility for neural networks regressor.

Let's have a closer look at the first commodity.

7.1. Gold

Graph below is visualizing our neural networks regressor performance on the trading data for year 2018.

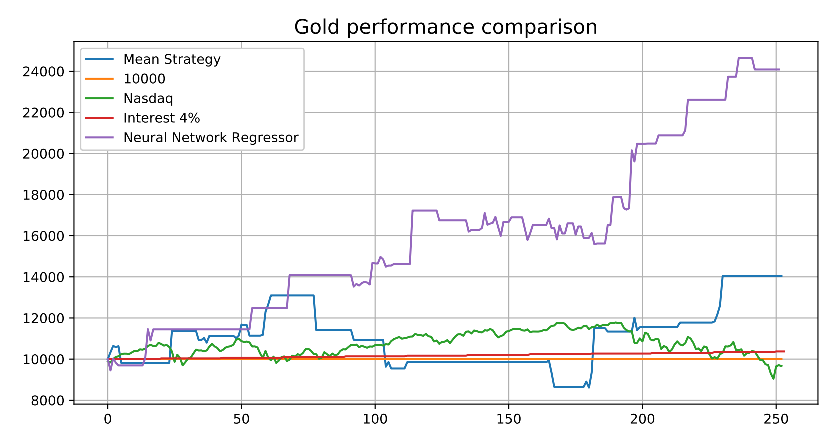


Table below is showing finalized gains/losses for year 2018.

Strategy	Start of year 2018	End of year 2018	Net Gains
trading gold model NN	10000	24000	14000
trading mean reversal	10000	14000	4000
investing in bonds	10000	10400	400
investing in NASDAQ	10000	9720	-280

Gold Neural Network regressor trade details for 2018.

Predictions/Actual volatility correlation	0.16
Returns	140.86 %
Number of trades	72
Number of short trades	15
Number of long trades	57
Earnings per trade	195.65
Earnings per short trade	205.63
Earnings per long trade	193.02

It is important to realize that NASDAQ and bonds investment are requiring least of effort. These are theoretically only 2 (maybe one trade in case of bonds) trades. We buy NASDAQ in the beginning of the year and sell in the end.

Other strategies as trading gold model NN trading and mean reversal require much more effort. We need to trade periodically and make tens of trades.

With each trade the risk is increasing. In case of gold our model has reached high enough performance to be considered for actual trading.

7.2. Corn

Graph below is visualizing our LightGBM regressor performance on the trading data for year 2018.

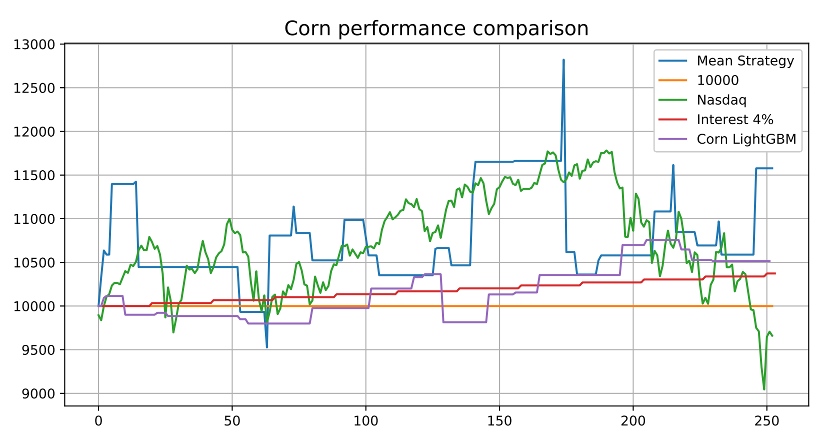


Table below is showing finalized gains/losses for year 2018.

Strategy	Start of year 2018	End of year 2018	Net Gains
trading corn LGBM Regressor	10000	10500	500
trading mean reversal	10000	11550	1550
investing in bonds	10000	10400	400
investing in NASDAQ	10000	9720	-280

Corn LightGBM regressor trade details for 2018.

Predictions/Actual volatility correlation	0.078
Returns	5.14 %
Number of trades	20
Number of short trades	11
Number of long trades	9
Earnings per trade	25.70
Earnings per short trade	44.92
Earnings per long trade	2.21

From the perspective of 10000$ account corn seems like a good trading choice. We haven't reached very high returns but trading strategy turned out profitable for the year 2018. Which would be better than investing into NASDAQ — which encountered decline. Mean reversal trading strategy generated more gains but equity curve seems very sharp.

LightGBM regressor trading strategy is performing only slightly better than much safer investment into bonds. In the current form it might be used as a starting strategy for environment setup. Later on we would have to revisit the data and create better performing model.

7.3. Coffee

Graph below is visualizing our LightGBM regressor performance on the trading data for year 2018.

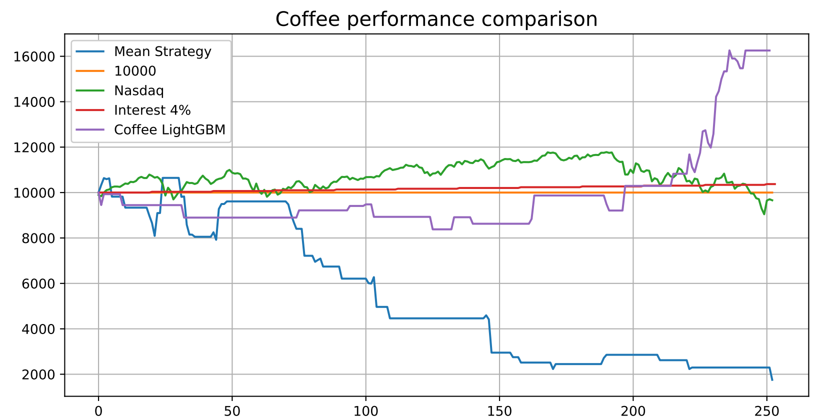


Table below is showing finalized gains/losses for year 2018.

Strategy	Start of year 2018	End of year 2018	Net Gains
trading coffee LGBM Regressor	10000	16200	6200
trading mean reversal	10000	1900	-8100
investing in bonds	10000	10400	400
investing in NASDAQ	10000	9720	-280

Coffee LightGBM regressor trade details for 2018.

Predictions/Actual volatility correlation	-0.001
Returns	62.54 %
Number of trades	37
Number of short trades	24
Number of long trades	13
Earnings per trade	169.04
Earnings per short trade	216.42
Earnings per long trade	81.56

Coffee end of year returns seem very good for coffee. Equity curve throughout the year seems odd. We already mentioned that there is something strange with coffee data (see PCA explained variance and autodetected outliers). Looking at possible gains in coffee market it is a lucrative commodity, but it seems we would have to rethink our approach to this market (which might help with two other commodities as well).

8. Conclusion

Trading is a difficult ML problem. Out of three compared commodities gold, corn and coffee we were able to predict volatility/trade class with gold and corn. Coffee behaved randomly with approximately 0 correlation to the actual volatility.

In the beginning of project, I was thinking of a classifier (short, no trade, long) because it is closer to the usage of model. I tested regressor as well. I think that regressor works better because there is more information preserved for training. I tried different loss functions when training regressor. I decided to use weighted MSE. This could be further modified for better function omitting errors below threshold.

Gold trading is the most capital intensive with very big stop losses (1000 long, 800 short). This can be problem for trading with 10000$ account.

Corn moves are much smaller and seem more suitable for the start of trading. General risk is more acceptable (stop-loss 300$). In case of corn mean reversal strategy is defeating our trading strategy.

In case of coffee the data was almost impossible to classify. I suspect more data transformation is needed to get better results.