COURSE PROJECT IN
STATISTICAL MACHINE LEARNING

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Statistical Machine Learning Project

Variable Importance Measures on Random Forest

Antoine Zacharie Launay, Cheng Zilan, Luo Yongyi

January 5, 2021

Report Type:	Course Project
Master Students:	Antoine Zacharie Launay Cheng Zilan Luo Yongyi
Professor:	Dr. Guillaume Obozinski
Date:	January 5, 2021

1 Introduction

In this report, we summarize the main ideas for computing variable importance in classification random forest. First, we introduce the concept and the priciples of 3 kinds of variable importance. Then, We apply the measures on BTCUSDT Price datasets using R and analyze the results and find that the measures work efficiently.

2 Variable Importance Measures

2.1 Gini Importance

The basic idea of Gini Importance is to access the variable importance by accumulating over each tree the improvement in the splitting criterion metric in each split. For the regression case, the splitting criterion metric would be simply the squared error. For the classification case, we introduce the concept of Gini Impurity, which measures how often a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the frequency of the levels in the subset.

Gini impurity is formally defined as follows:

$$GI = \sum_{i=1}^{I} p(i) * (1 - p(i))$$

If there are I divisions and the probability of an element in the i^{th} subset is $p(i)$.

For the binary target variable, we define the weighted Gini impurity at node k:

$$G_k := \min_{j,s}\{GI_1(j,s) \cdot \frac{|R_1(j,s)|}{|R_1(j,s)| + |R_2(j,s)|} + GI_2(j,s) \cdot \frac{|R_1(j,s)|}{|R_1(j,s)| + |R_2(j,s)|}\},$$

where the right hand side depends on k through the observations considered.

For the multiple target variable, We define the weighted Gini impurity at node k:

$$G_k := \min_{j,s} \sum_{i=1}^{n} \{GI_i(j,s) \cdot \frac{|R_i(j,s)|}{\sum_{i=1}^{n} |R_i(j,s) \sum_{i=1}^{n}}\},$$

where the right hand side depends on k through the observations considered.

Let M_b be the number of nodes in the b-th tree of the random forest $\{h(T_b\}_{b \in B \subseteq \mathbb{N}}$ (not including terminal nodes i.e. leafs). Then the Gini Importance for the feature X_j is defined as

$$I_{gini(j)} = \sum_{b=1}^{B} \{ \sum_{m=1}^{M_b} (GI_m^{parent} - G_m) \mathbb{1}_{\{split\ is\ made\ upon\ X_j\}} \}$$

where GI_m^{parent} is defined as the Gini Impurity in the node m i.e. parent node w.r.t to the split and G_m is defined as the weighted Gini Impurity resulting from the split.

Gini Importance or Mean Decrease in Impurity (MDI) directly calculates each feature importance as the improvement in the Gini Impurity (across all trees) that include the feature, proportionally to the number of samples it splits. Sadly, it is often computed on in-bag samples, which leads to the risk of overfitting. Besides, it has a bias in favor continuous variables and discrete variables with many values.

2.2 Permutation Importance

Unconditional Permutation Importance is defined to be the decrease in a model score when a single feature value is randomly shuffled[1]. This procedure breaks the relationship between the feature and the target, therefore the drop in the model score is indicative of how much the model depends on the feature[1]. Based on experiments on out-of-bag(OOB) samples, we calculate permutation importance.

Algorithm 1: Pseudo code for calculating permutation importance for a single feature X_j

```
Fit a random forest $h(T_b) : b \leq B$ on the training set $T$.
for $b = 1$ to $B$ do
    (a)Compute the OOB prediction accuracy of the $b$-th tree $h(T_b)$.
    (b)Permute randomly the observations of the feature $X_j$ in the OOB sample $O_b$ once.
    (c)Recompute the OOB prediction accuracy of the $b$-th tree $h(T_b)$ using the permuted input.
    (d)Compute $I^b_{permute}(j)$.
end
Compute the average decrease of prediction accuracy over all trees i.e. $I^b_{permute}(j)$.
```

2.3 Conditional Permutation Importance

Unconditional permutation importance favors correlated predictor variables due to two reasons. Firstly, it has preference of correlated predictor variables in (early) splits when fitting the random forest[1]; also, it has the inherent null hypothesis[1]. Therefore, to avoid bias in a setting with highly correlated features, we distinguish the marginal effect and the conditional effect of a single predictor variable on the target by applying conditional permutation importance.

Algorithm 2: Pseudo code for calculating conditional permutation importance for feature X_j

```
Fit a random forest $h(T_b) : b \leq B$ on the training set $T$.
for $b = 1$ to $B$ do
    (a)Compute the OOB prediction accuracy of the $b$-th tree $h(T_b)$.
    (b)Determine the variables Z to be conditioned on, extract all cutpoints for each variable in
    the current tree and construct the grid by bisecting the feature space in each cutpoint.
    (c)Within this grid permute the observations of the feature $X_j$ in the OOB sample $O_b$.
    (d)Recompute the OOB prediction accuracy of the $b$-th tree $h(T_b)$ using the permuted input
    i.e. (for classification) compute
        $$\frac{\sum_{i \in O_b} \mathbb{1}_{\{y_i = \hat{y}^b_{i,\pi_j|Z}\}}}{|O_b|}$$
    where= $\hat{y}^b_{i,\pi_j|Z} = h(T_b)(x_{j,x_j|Z})$ is the prediction of the $b$-th tree after permuting the
    observations of $X_j$ within the grid defined by Z.
    (e)Compute $I^b_{permute,conditional} j$ using (a) and (d) analogously as for $I^b_{permute}(j)$.
end
Compute the average decrease of prediction accuracy over all trees which we will denote by
$I^b_{permute,conditional} j$ analogously using $I^b_{permute,conditional} j$ as for $I_{permute}(j)$
```

3 Application on BTCUSDT Price Data

In this section we will discuss our application of variable importance measures on BTCUSDT data set and the experiments we have done.

3.1 Data Description

BTCUSDT, stands for the exchange rate of Bitcoin v.s. USDT. BTC refers to bitcoin. USDT represents the token Tether USD which is a stable value currency worth 1USDT = 1USD. Our dataset consists of the bitcoin price expressed in USDT, from the 2020-07-03 07:00:00 to 2020-12-02 22:00:00, GMT Time, on a 5-minutes timeframe. The original variables of our dataset were: timestamp, opening price, highest price, lowest price, close price and the volume. From this first set of variables we use ta-lib python package to generate more than 100 consequent variables of the following types: volume-based, volatility-based, trend-based, momentum-based and 5-min-return indicators, which are explicitly described in **Table 1** in **Appendix**. We built our response variable Y as follow: $Y = 1$ if the 5-min return is higher than 0.2%, otherwise $Y = 0$.

3.2 Construction of Random Forest

Based on 3664 observations, 51 predictors and a binary classification response variable, we construct two random forests. The first one is used to compute Gini Importance and Permutation Importance, constructed by function **randomforest** in R package **randomForest**. Applying **importance(...,type=2)** on it, we compute the Gini Importance; while **importance(...,type=1)** computes the Unconditional Permutation Importance. In the first tree, we need to set hyperparameters: $mtry$ and $ntree$, which stand for the number of variables randomly sampled as candidates at each split and the number of trees respectively. After trying, we set $mtry = 7$ and $ntree = 40$, where $mtry$ is nearly the square root of the number of predictors. Its OOB estimate of error rate is 0.05%, while any change of the hyperparameters does not improve the error.

The second random forest is constructed by function **cforest** in R package **party**. The Unconditional Permutation Importance can be computed by **varimp(..., conditional=FALSE)**, we refer it as Standard Importance in the conclusions and plots. The Conditional Permutation Importance can be computed by **varimp(..., conditional=TRUE)**. We compute the Permutation Importance twice in these two random forests to see if they have the same results.

3.3 Experiments

Besides computing the Gini Importance, Permutation Importance and Conditional Permutation Importance on the entire data, we conduct two experiments. In the first experiment, we divide the data into two parts with equal size of 1832 observations and run the same code. We keep the $mtry = 7$ and $ntree = 40$ and both random forests' constructed by **randomforest** have increased OOB estimate of error rates, 0.11%, due to the reduction of sample size. In the second experiment, we use the 3664 observations with removing the most important variables, PC, DR and DLR, and see what will happen to the rest variables' importance.

4 Conclusions

These experiments led us to notice that some variables entice a much higher variable importance than others. If time would have allowed us, it would have been worth building an actual prediction model based on the variables with the highest variable importances.

In **Figure 1**(see: **Appendix**) we have the correlation matrix which shows us strong correlation between some subsets of variables. This makes sense as we have generated our variables from the ta-lib python packages, which generate variables of certain type. For example volume-based variables (i.e. we should expect volume-based variables to entice strong correlations between one another). Similarly for the other types of variables (volatility, momentum, trend, and log-return types). Also, it indicates that we need to consider conditional permutation importance, since unconditional permutation importance shows bias in this setting.

We then proceed to the computation of the three types of variable importance's measures, respectively Gini importance, Permutation Importance, Standard Importance and Conditional Permutation Importance. The results are obtained in **Figure 2** (see: **Appendix**) and one can see that 3 variables stand out, namely PC (Percentage change), DR (5-min return), and DLR (5-min log return) on all 4 measures. This is to be expected since the response variable build upon PC (Percentage change), which yields a strong correlation to the return and the log-return.

In order to investigate the correctness of our algorithm, we then decide to split the data into two parts, re-run the variable importance's measures algorithms on both parts, as per **Figure 3, 4, 5 and 6** (see: **Appendix**), and see if both parts entail similar results.

We do indeed have similar results after the split of the data. This is a sign of good health of the algorithms we used.

Again, in **Figure 3, 4, 5, and 6,** as in Figure 2, the 3 variables, PC, DR, and DLR still stands out.

Eventually, in order to apply our current methodology to a real world scenario (assuming we which to accurately predict the BTCUSDT exchange rate) one would not know in advance the percentage change nor the 5-min return variables. Thus we decided to re-run the experiments, this time only, without the 3 variables PC, DR, and DLR.

This time, our Conditional Permutation Importance algorithm would crash. We assume that the computational power necessary was not sufficient. (note: the 3 former variables PC, DR, and DLR were so good in predicting the model this explains why Conditional Permutation Importance algorithm could previously perform well under the former set of variables)

As **Figure 7**(see: **Appendix**), this last experiment entails encouraging results as 5 variables stands out on all 3 measures, namely two volume-based indicators; MFI (Money Flow Index), EM (Ease of Movement), and three moment-based indicators; SRSI (Stochastic Relative Strenght Index), SR (Stochastic Oscillator), and WR (Williams R).

Since all 3 measures agrees on these 5 variables one may assume that they make strong indicators to accurately predict a change in our response variable (and on broader perspective a change in the BTCUSDT exchange rate)

As a further experiment, if we had more time it would have been interesting to go beyond and build an actual predictive model, and see how accurately it performs.

References

[1] Dipl.-Ing. Jakob Weissteiner. Variable importance measures in regression and classification methods. Institute for Statistics and Mathematics Vienna University of Economics and Business.

Appendix

Table 1: Variable Description

Category	Notation	Description
Volume	ADI	Accumulation/Distribution Index (ADI)
	OBV	On-Balance Volume (OBV)
	CMF	Chaikin Money Flow (CMF)
	FI	Force Index (FI)
	MFI	Money Flow Index (MFI)
	EM	Ease of Movement (EoM, EMV)
	VPT	Volume-price Trend (VPT)
	NVI	Negative Volume Index (NVI)
	VWAP	Volume Weighted Average Price (VWAP)
volatility	ATR	Average True Range (ATR)
	BBM	Bollinger Bands (BB)
	BBH	
	KCC	Keltner Channel (KC)
	KCH	
	DCL	Donchian Channel (DC)
	DCH	
	UI	Ulcer Index (UI)
Trend	MACD	Moving Average Convergence Divergence (MACD)
	SMA$_{fast}$	Simple Moving Average (SMA)
	SMA$_{slow}$	
	EMA$_{fast}$	Exponential Moving Average (EMA)
	EMA$_{slow}$	
	ADX	Average Directional Movement Index (ADX)
	VI	Vortex Indicator (VI)
	TRIX	Trix (TRIX)
	MI	Mass Index (MI)
	CCI	Commodity Channel Index (CCI)
	DPO	Detrended Price Oscillator (DPO)
	KST	KST Oscillator (KST)
	Ichimoku$_{conv}$	Ichimoku Kinkō Hyō (Ichimoku)
	Ichimoku$_{base}$	
	AROON$_{up}$	Aroon(Aroon)
	AROON$_{down}$	
	PSAR$_{up}$	Parabolic Stop And Reverse (Parabolic SAR)
	PSAR$_{down}$	
	STC	Schaff Trend Cycle (STC)
Momentum	RSI	Relative Strength Index (RSI)
	SRSI	Stochastic RSI (SRSI)
	TSI	True strength index (TSI)
	UO	Ultimate Oscillator (UO)
	SR	Stochastic Oscillator (SR)
	WR	Williams %R (WR)
	AO	Awesome Oscillator (AO)
	KAMA	Kaufman’s Adaptive Moving Average (KAMA)
	ROC	Rate of Change (ROC)

Table 1: Variable Description

Category	**Notation**	**Description**
	PPO PPO_{hist}	Percentage Price Oscillator (PPO)
Others	DR DLR CR	5-min Return (DR) 5-min Log Return (DLR) Cumulative Return (CR)
Outcome	PC	Percentage Change
Response Variable	Y	Y = 1 if DR (5-min Return) > 2% , 0 otherwise

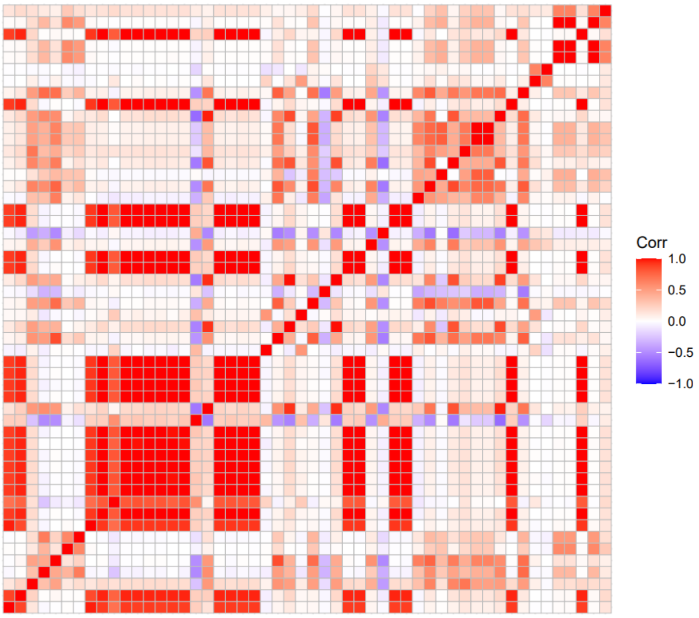
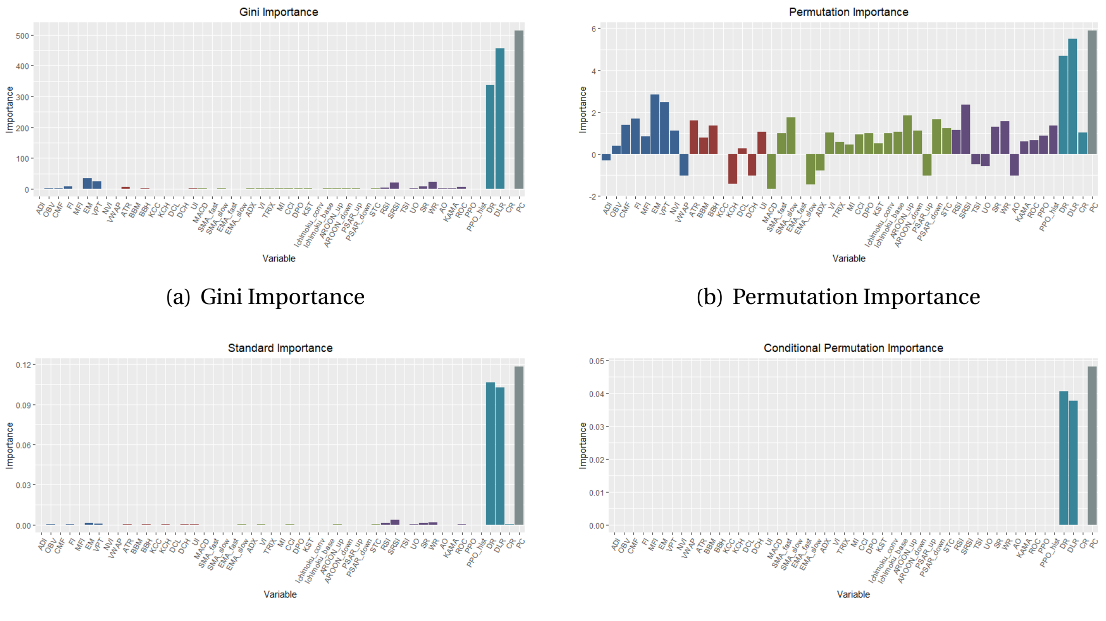


Figure 1: Correlation Matrix



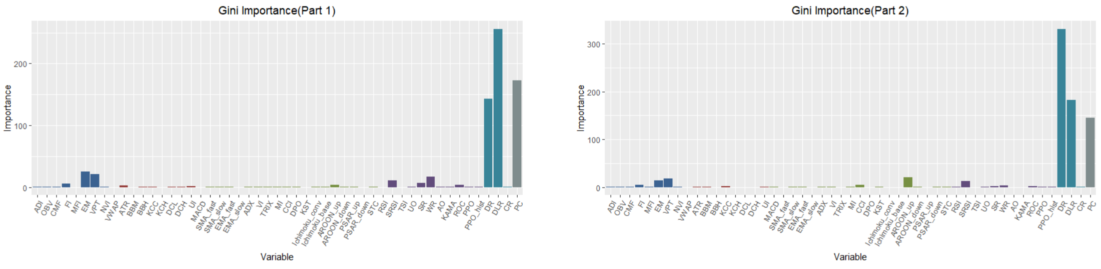
(a) Gini Importance

(b) Permutation Importance

(c) Standard Importance

(d) Conditional Permutation Importance

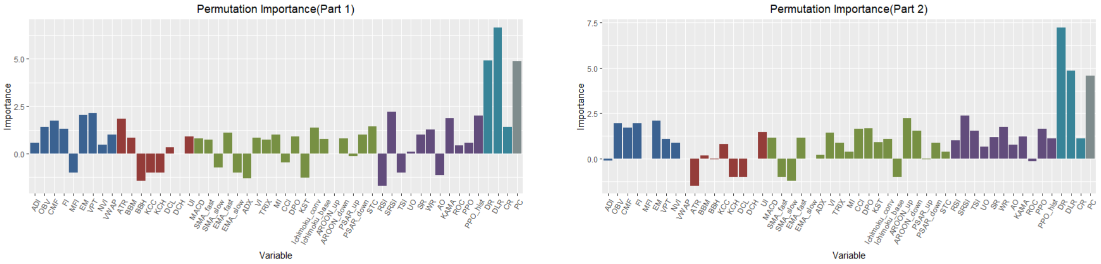
Figure 2: Variables' importance over the whole data



(a) Gini Importance(Part I)

(b) Gini Importance(Part II)

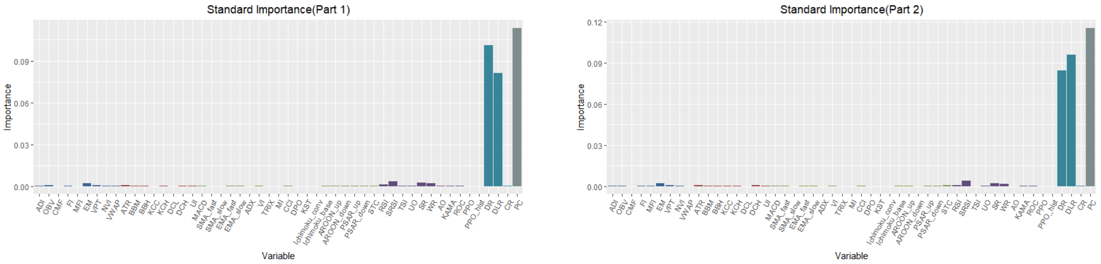
Figure 3: Experiment I: split the data into two parts and compute Gini Importance respectively using **randomForest**



(a) Permutation Importance(Part I)

(b) Permutation Importance(Part II)

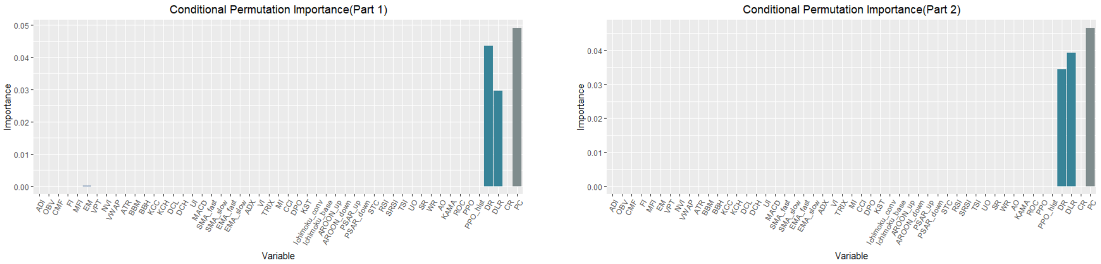
Figure 4: Experiment I: split the data into two parts and compute Unconditional Permutation Importance respectively in the first random forest using **randomForest**



(a) Standard Importance(part I)

(b) Standard Importance(part II)

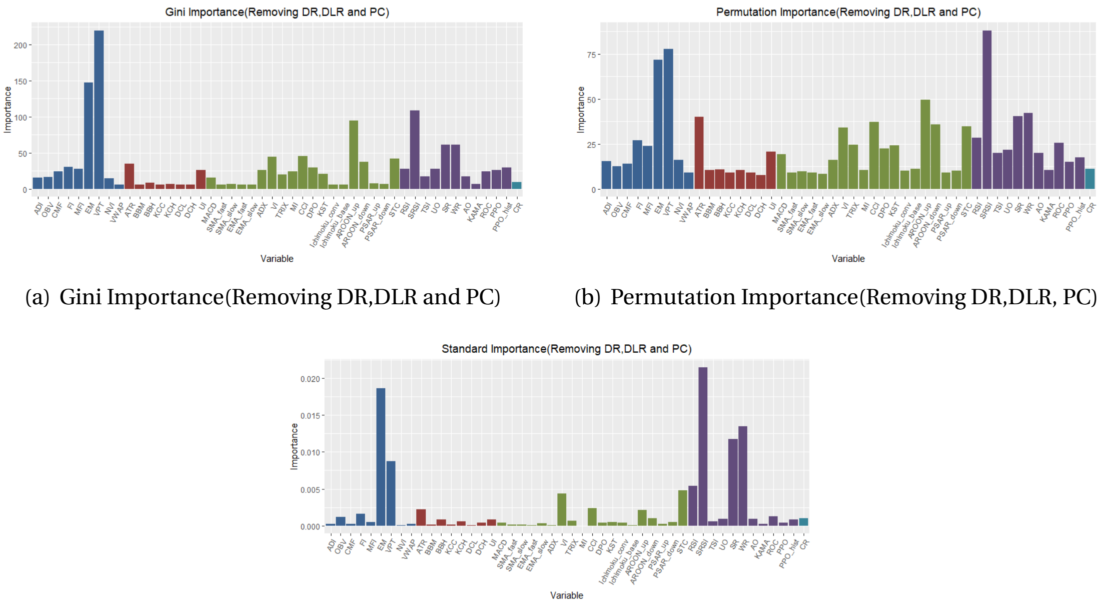
Figure 5: Experiment I: split the data into two parts and compute Permutation Importance respectively in the second random forest using **cforest**



(a) Conditional Permutation Importance(Part I)

(b) Conditional Permutation Importance(Part II)

Figure 6: Experiment I: split the data into two parts and compute Conditional Permutation Importance respectively in the second random forest using **cforest**

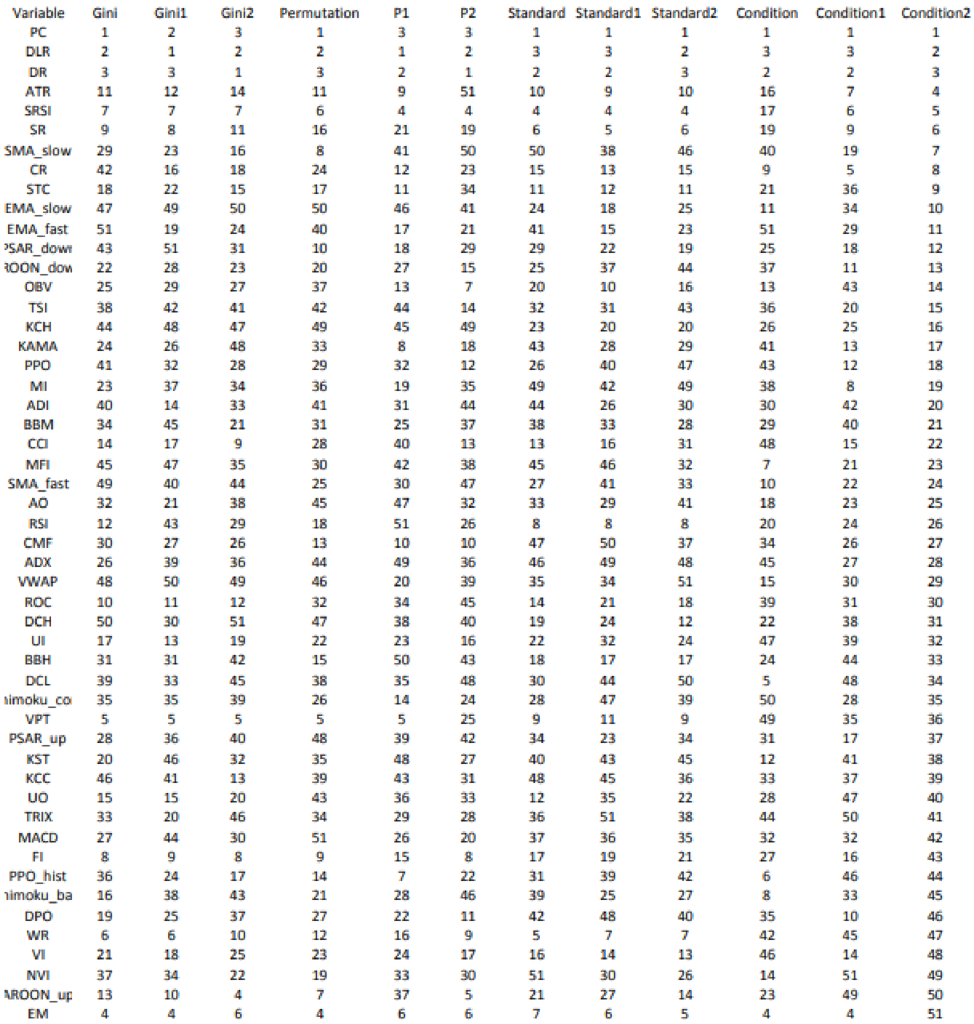


(a) Gini Importance(Removing DR,DLR and PC)

(b) Permutation Importance(Removing DR,DLR, PC)

(c) Standard Importance(Removing DR,DLR and PC)

Figure 7: Experiment II: split the data into two parts and compute Conditional Permutation Importance respectively in the second random forest using **cforest**



Variable	Gini	Gini1	Gini2	Permutation	P1	P2	Standard	Standard1	Standard2	Condition	Condition1	Condition2
PC	1	2	3	1	3	3	1	1	1	1	1	1
DLR	2	1	2	2	1	2	3	3	2	3	3	2
DR	3	3	1	3	2	1	2	2	3	2	2	3
ATR	11	12	14	11	9	51	10	9	10	16	7	4
SRSI	7	7	7	6	4	4	4	4	4	17	6	5
SR	9	8	11	16	21	19	6	5	6	19	9	6
SMA_slow	29	23	16	8	41	50	50	38	46	40	19	7
CR	42	16	18	24	12	23	15	13	15	9	5	8
STC	18	22	15	17	11	34	11	12	11	21	36	9
EMA_slow	47	49	50	50	46	41	24	18	25	11	34	10
EMA_fast	51	19	24	40	17	21	41	15	23	51	29	11
›SAR_dowr	43	51	31	10	18	29	29	22	19	25	18	12
₹OON_dow	22	28	23	20	27	15	25	37	44	37	11	13
OBV	25	29	27	37	13	7	20	10	16	13	43	14
TSI	38	42	41	42	44	14	32	31	43	36	20	15
KCH	44	48	47	49	45	49	23	20	20	26	25	16
KAMA	24	26	48	33	8	18	43	28	29	41	13	17
PPO	41	32	28	29	32	12	26	40	47	43	12	18
MI	23	37	34	36	19	35	49	42	49	38	8	19
ADI	40	14	33	41	31	44	44	26	30	30	42	20
BBM	34	45	21	31	25	37	38	33	28	29	40	21
CCI	14	17	9	28	40	13	13	16	31	48	15	22
MFI	45	47	35	30	42	38	45	46	32	7	21	23
SMA_fast	49	40	44	25	30	47	27	41	33	10	22	24
AO	32	21	38	45	47	32	33	29	41	18	23	25
RSI	12	43	29	18	51	26	8	8	8	20	24	26
CMF	30	27	26	13	10	10	47	50	37	34	26	27
ADX	26	39	36	44	49	36	46	49	48	45	27	28
VWAP	48	50	49	46	20	39	35	34	51	15	30	29
ROC	10	11	12	32	34	45	14	21	18	39	31	30
DCH	50	30	51	47	38	40	19	24	12	22	38	31
UI	17	13	19	22	23	16	22	32	24	47	39	32
BBH	31	31	42	15	50	43	18	17	17	24	44	33
DCL	39	33	45	38	35	48	30	44	50	5	48	34
ıimoku_co	35	35	39	26	14	24	28	47	39	50	28	35
VPT	5	5	5	5	5	25	9	11	9	49	35	36
PSAR_up	28	36	40	48	39	42	34	23	34	31	17	37
KST	20	46	32	35	48	27	40	43	45	12	41	38
KCC	46	41	13	39	43	31	48	45	36	33	37	39
UO	15	15	20	43	36	33	12	35	22	28	47	40
TRIX	33	20	46	34	29	28	36	51	38	44	50	41
MACD	27	44	30	51	26	20	37	36	35	32	32	42
FI	8	9	8	9	15	8	17	19	21	27	16	43
PPO_hist	36	24	17	14	7	22	31	39	42	6	46	44
ıimoku_ba	16	38	43	21	28	46	39	25	27	8	33	45
DPO	19	25	37	27	22	11	42	48	40	35	10	46
WR	6	6	10	12	16	9	5	7	7	42	45	47
VI	21	18	25	23	24	17	16	14	13	46	14	48
NVI	37	34	22	19	33	30	51	30	26	14	51	49
ʌROON_up	13	10	4	7	37	5	21	27	14	23	49	50
EM	4	4	6	4	6	6	7	6	5	4	4	51

Figure 8: Variable Importance rankings