Panel Trees Under Global Criterion: An Application to Asset Pricing *

Xin He† Lin William Cong‡ Guanhao Feng§ Jingyu He¶

September 28, 2021

Abstract

We introduce a class of tree-based models for analyzing panel data (P-Tree), with iterative global (instead of recursive local) splitting criterion, generating the stochastic discount factor (SDF) to cross-sectional returns. Distinct from the standard tree algorithm in machine learning, our multi-period tree algorithm accommodates the imbalanced panel data structure of individual asset returns, and grows using the bespoke non-arbitrage split criterion from asset pricing theory. The top-down generated leaf-basis portfolios constitute mean-variance clusters when splitting the cross-section of assets, while the generated SDF is a bottom-up output that fits cross-sectional returns. We find that P-Tree outperforms standard factor models for different pricing and prediction measures. A five-factor model constructed by boosted P-Tree cannot be explained by Fama-French models and delivers a 1.79 out-of-sample (3.50 in-sample) annualized Sharpe ratio. The out-of-bag variable importance evaluation shows only a few significant characteristics to drive cross-sectional return variation, such as volume volatility and industry-adjusted market equity. Finally, We apply P-Tree to split the panel of return data over time series and cross-section dimensions and find inflation as the most important macro predictor for regime switching.

*We thank Richard Hahn, Stefan Nagel, Nick Polson, and Dacheng Xiu for invaluable discussions. We are grateful for helpful comments from seminar and conference participants at Shanghai Jiao Tong University, Shanghai University of Finance and Economics, HKAIFT-Columbia joint seminar.

†E-mail address: xin.he@my.cityu.edu.hk.

‡E-mail address: will.cong@cornell.edu.

§E-mail address: gavin.feng@cityu.edu.hk.

¶E-mail address: jingyuhe@cityu.edu.hk.

Key Words: Cross-Sectional Returns, Firm Characteristics, Multi-Period, Regression Tree, Stochastic Discount Factor, Tree Ensembles

Contents

1 Introduction

The goal of empirical asset pricing is to understand the cross-sectional return variation: why do different stocks earn different returns? Despite the continuous success for linear factor models[1], there is a rising trend of applying nonlinear method, or machine learning, to explain the cross-section[2]. Regardless of the positive pricing or prediction performance, many economics and finance researchers still view nonlinear methods like deep learning as black boxes. This paper fills the interpretation gap for nonlinear methods in empirical asset pricing with a decision tree approach for splitting the cross-section of individual stocks. We advocate using tree models in asset pricing for their extraordinary ability to fix nonlinear functions while maintaining interpretability.

This paper develops a panel tree (P-Tree) approach that provides a unified framework to generate and estimate a latent factor model for potentially imbalanced panel data. In the empirical application, we adapt the model with a global criterion for the non-arbitrage asset pricing to individual stock returns' imbalanced panel data structure. With the graphical visualization of a decision tree, the panel tree offers an alternative top-down solution to security sorting for splitting the cross-section of stocks. In addition to understanding the stock return variation, the panel tree creates a cluster of test assets to group similar stocks into the same leaves by their past characteristics. Under the asset pricing global criterion, the panel tree generates a stochastic discount factor and estimates a time-varying beta factor model for individual stock returns. This paper is related to the literature on latent factor construction via machine learning and deep learning[3].

The famous Classification and Regression Trees (CART)[4] is introduced by Breiman et al. (1984) to refer to a specific decision tree algorithms for classification or regression. For the example of stock returns, as a nonlinear machine learning method, the decision tree allows researchers to split the cross-section and group stocks into different leaves by past firm characteristics. Noticeably, the nonlinear structure and interaction of characteristics are graphically displayed when the decision tree continues the split to further layers. This unique feature for the decision tree allows its straightforward and convenient interpretation to researchers and practitioners.

[1] Fama and French (1993),Hou et al. (2015), and Fama and French (2015)
[2] Gu et al. (2021),Chen et al. (2020), and Feng et al. (2020)
[3] Lettau and Pelger (2020), Kelly et al. (2019), Kim et al. (2020), Chen et al. (2020), Feng et al. (2020), and Gu et al. (2021)
[4] See Section 1.2 for review of CART.

Before the CART algorithm, the decision tree has numerous applications in many fields of economics and finance. Despite the prediction application, the decision tree helps split or cluster observations based on individual features. For example, investment practitioners usually split the stock universe by their market equity values (large-cap v.s. small-cap). They might consider further splits on value and growth stocks. The decision tree shares a similar implementation to security sorting used in empirical asset pricing, see Bryzgalova et al. (2020), and Creal and Kim (2021).

Specifically, the decision tree approximates the conditional sorting scheme instead of the simultaneous sorting scheme (i.e., Fama-French ME - B/M 25 portfolios). First, most security sorting procedures in empirical asset pricing do not consider the cutpoints but sort stocks into quintile or decile portfolios. Second, for conditional sorting, the order of splitting characteristics is important because of the interaction effect. The CART algorithm is developed to search the optimal cutpoints and determine the optimal order of splitting characteristics when building the decision tree. However, the original CART algorithm is designed for independent and identically distributed (i.i.d.) observations but not panel data. Its split criterion is designed to fit the average returns better using firm characteristics rather than creating a factor model. The conditional security sorting procedure can be viewed as a single-period tree building process since the CART algorithm predicts everything with a constant. When we observe cross-sectional returns for multiple periods, the economic theory requires building the same tree for every period.

The i.i.d. assumption of data is usually not valid in economics or finance. For example, stock returns are in an imbalanced panel data structure and may follow a common factor model. Stock-return observations have two dimensions: cross-section and time series. There exists strong cross-sectional dependence for stock returns in the same period and weak serial dependence for the time series of stock returns. Researchers expect the tree model that works consistently for multiple periods. Therefore, we provide the first multi-period tree to approximate conditional security sorting in empirical asset pricing for this imbalanced panel data structure.

Also, the splitting criterion in Breiman et al. (1984) is the aggregated mean squared error, yet which can be revised and adapted to different areas. Their original recursive CART algorithm optimizes at each split for the corresponding parent node and does not consider other parent nodes. This greedy strategy focuses on local optimization and usually leads to overfitting of the data.

Hence, our P-Tree addresses cross-sectional returns' underlying common factor model for the non-arbitrage condition and adopts an iterative optimization scheme as the global split criterion. Our split criterion considers the asset pricing improvement by estimating the loss from a linear factor model. For this common factor problem, ours considers all stocks in the cross-section for every split, including those not in the splitting parent node.

In the empirical study, we apply our method to individual stock returns in the U.S. equity market from 1981 to 2020. The P-Tree model shows tremendous pricing and prediction performance in both in-sample and out-of-sample analysis. We find that P-Tree outperforms a few standard factor models and different PCA latent factor models for different pricing and prediction measures. A five-factor model constructed by boosted P-Tree can not be explained by Fama-French models and delivers a 1.78% out-of-sample (3.62% in-sample) annualized Sharpe ratio. The out-of-bag variable importance evaluation shows only a few significant characteristics to drive cross-sectional return variation, such as volume volatility and industry-adjusted market equity. Finally, We apply P-Tree to split the panel of return data over time series and cross-section dimensions and find inflation as the most important macro predictor for regime switching.

In summary, there are several advantages of applying decision trees in empirical asset pricing. First of all, the decision tree provides a clear interpretation and future mapping for splitting the cross-section. Second, the interactions between characteristics are displayed graphically within a tree model by their sequential orders for splitting the cross-section. Finally, the regression tree is highly adaptive to the low signal-noise data environment and relatively short data history. Our P-Tree generates and estimates a latent factor model using a specifically designed decision tree, and the performance is comparable to recent research using different PCA or deep learning methods[5]. In addition, PCA methods lack the first two advantages, while deep learning methods do not have the first and third advantages.

1.1 Related Literature

Early attempts of the regression tree (or random forest, boosted regression trees ensembles) for asset returns include Moritz and Zimmermann (2016) and Rossi (2018). Bryzgalova et al. (2020)

[5] Lettau and Pelger (2020), Kelly et al. (2019), Kim et al. (2020), Chen et al. (2020), Feng et al. (2020), and Gu et al. (2021)

proposes pruning a existing tree as a regularized portfolio optimization problem. Their paper aims to include the elastic-net regularization for pruning the tree, while do not focus on how to grow the tree, and the split criterion is not adapted for asset pricing. Creal and Kim (2021) use the tree structure to split asset-return observations and implement the Bayesian factor model on each leaf within the Bayesian additive regression tree. They also apply an asset pricing likelihood for building the tree, but their model does not distinguish between cross-section and time series dimensions.

This paper also contributes to the growing literature in empirical asset pricing that develops latent factor models for pricing cross-sectional returns. This literature usually consists of two goals: asset pricing performance on explaining cross-sectional returns and average returns, plus the risk-adjusted investment performance. Lettau and Pelger (2020), Kelly et al. (2019) and Kim et al. (2020) show different principal component methods to produce latent factors, while Chen et al. (2020), Feng et al. (2020) and Gu et al. (2021) generate the SDF through various deep learning models. In our empirical analysis, we show several comparative advantages for our P-Tree.

This paper is also related to conditional asset pricing models and the construction of basis assets. We follow Avramov (2004), Gagliardini et al. (2016), and Feng and He (2019) to allow time-varying factor loadings for individual stock returns. While our factor is generated by the tree, the estimation for characteristics-driven factor loading is incorporated within the split criterion. Unlike the deterministic sorting in Bryzgalova et al. (2020), our data-driven cross-section split creates the leaf basis portfolios. In our empirical analysis, we show the clustering patterns and investment performance for leaf basis portfolios generated by shallow and deep tree models.

Finally, our paper belongs to the vast literature for machine learning in finance. Freyberger et al. (2019), Gu et al. (2020), Cong et al. (2021), and Avramov et al. (2021) find equity returns predictable using machine learning and deep learning methods. Bianchi et al. (2018) and Feng et al. (2020) find positive results on treasury bond returns, while Bali et al. (2020) and He et al. (2021) find positive results on corporate bond returns. For market timing and portfolio construction, Cong et al. (2020) provide a reinforcement learning approach, while DeMiguel et al. (2018) and Feng and He (2019) provide portfolio regularization methods by optimization or Bayesian modeling. The return prediction performance is one application for our P-Tree, while our method also considers the regularized estimation to recover the stochastic discount factor.

1.2 Classification and Regression Tree

A tree is a set of split rules (or cutpoints) defining a rectangular partition of the covariate space. The cutpoint essentially is a pair $c = (x_i, c_i)$ indicating the variable and corresponding value to cut. The central problems for decision trees are how to grow the tree and choose the cutpoints. Classification and regression trees (CART) (Breiman et al., 1984) is arguably the most popular tree algorithm. It grows by recursive partition, examining all cutpoint candidates by a split criterion and choosing the optimized one. Then the data is divided according to the cutpoint, and the process repeats. It is worth emphasizing when considering split a node, and the recursive algorithm only processes with data in that specific node without looking at data in other branches. This greedy approach is preferred mainly for easy coding and efficient computation reasons.

Assume $r_{i,t}$ denote the return of asset i at time period t. The CART uses a squared loss split criterion, which treats the cross-sectional data as pool data. For each cutpoint candidate,

$$\sum_{i,t\in\text{left node}} (r_{i,t} - \bar{r}_{\text{left}})^2 + \sum_{i,t\in\text{right node}} (r_{i,t} - \bar{r}_{\text{right}})^2$$

where $\bar{y}_{\text{left}} = \frac{1}{\#_{\text{left node}}} \sum_{i,t\in\text{left node}} r_{i,t}$ and $\bar{y}_{\text{right}} = \frac{1}{\#_{\text{right node}}} \sum_{i,t\in\text{right node}} r_{i,t}$. Both left and right leaves are constants over not only cross-section but also time-series, which serve as pricing kernels for corresponding individual stocks. Note that the constant does not have subscript for time nor stock, which implies a one-period model for scalar outcomes.

However, as mentioned above, the two fundamental properties of CART are not suitable for asset pricing. We want to form a *basis portfolio* at each leaf node instead of a constant. The basis portfolio should be a vector representing returns of multiple periods. Second, the aim is to create stochastic discount factors from the basis portfolios at each leaf, implying that the factor should be defined globally across all leaves rather than a local recursive one. Our P-Tree model takes a global split criterion to solve those problems, which is evaluated on all leaf portfolios. Thus, the tree has to grow iteratively rather than recursively. Furthermore, we will discuss how to form basis portfolios at leaves using the new split criterion.

2 P-Tree Factor Model for Asset Pricing

2.1 A Conditional Stochastic Discount Factor Model

Our tree model generates a conditional stochastic discount factor (SDF) model and explains cross-sectional difference for individual stock returns. The conditional SDF model follows as below,

$$E_t\left[m_{t+1}r_{i,t+1}\right]=0 \Longleftrightarrow E_t\left[r_{i,t+1}\right]=\underbrace{\frac{\mathrm{Cov}_t\left(m_{t+1},r_{i,t+1}\right)}{\mathrm{Var}_t\left(m_{t+1}\right)}}_{\beta_{i,t}}\underbrace{\left(-\frac{\mathrm{Var}_t\left(m_{t+1}\right)}{E_t\left[m_{t+1}\right]}\right)}_{\lambda_t}, \tag{1}$$

where $r_{i,t+1}$ is the excess return of individual stock i at time $t+1$. The SDF model is supposed to be true for all individual stocks. For the market expectation at time t, $\beta_{i,t}$ is the SDF exposure of stock i, and λ_t is the risk price of the SDF. A natural solution to the SDF formulation, m_{t+1}, is a portfolio of individual stock returns,

$$m_{t+1}=1-w_t^\intercal r_{t+1}, \tag{2}$$

where $r_{t+1}=(r_{1,t+1},\cdots,r_{N,t+1})$ denotes returns of all N assets at time period $t+1$, w_t is a vector of portfolio weights. Plugging Equation 2 into Equation 1 yields the SDF portfolio weights

$$w_t=E_t\left[r_{t+1}r_{t+1}^\intercal\right]^{-1}E_t\left[r_{t+1}\right]. \tag{3}$$

However, it is challenging to estimate w_t for a large number of individual stock returns mainly due to the large dimension of the covariance term.

Following the latent factor literature[6], our tree model estimates the SDF using basis portfolios instead of individual stocks to solve the estimation challenge. Unlike existing literature where basis portfolios are pre-specified, we take advantage of the splitting natural of trees — our algorithm generates trees from the data to split the cross section of asset returns into many basis portfolios. The split criterion of the tree is developed specifically for this task, rather than the standard CART criterion. In addition, the criterion is based on global information instead of local information at a specific leaf. Thus our tree grows iteratively to generate both the leaf basis portfolio and the SDF.

The tree model splits the stock universe into non-overlapping basis portfolios by cross-sectional

[6] Lettau and Pelger (2020), Kelly et al. (2019), Chen et al. (2020), and Feng et al. (2020)

quantiles of past firm characteristics. The number of basis portfolios increases one at a time when the tree splits a parent node into two child nodes. At the k-th split of the tree growing process, we denote the SDF $m_{t+1}^{(k)}$ generated by basis portfolios $R_{t+1}^{(k)}$ as

$$m_{t+1}^{(k)} = 1 - W_t^{(k)} R_{t+1}^{(k)}, \tag{4}$$

$$W_t^{(k)} = E_t\left[R_{t+1}^{(k)} R_{t+1}^{(k)\intercal}\right]^{-1} E_t\left[R_{t+1}^{(k)}\right]. \tag{5}$$

For the conditional factor model, the time-varying factor exposure $\beta_{i,t}$ is given by:

$$\beta_{i,t}^{(k)} = \frac{\mathrm{Cov}_t\left(W_t^{(k)} R_{t+1}^{(k)}, r_{i,t+1}\right)}{\mathrm{Var}_t\left(W_t^{(k)} R_{t+1}^{(k)}\right)}. \tag{6}$$

It is common to model the time-varying factor exposures driven by past firm characteristic updates.[7] Therefore, after the k-th split of the tree growing process, we adopt a reduced-form approximation to $\beta_{i,t}^{(k)}$.

$$\beta_{i,t}^{(k)} = b_0^{(k)} + b_1^{(k)\intercal} z_{i,t}, \tag{7}$$

where $z_{i,t}$ are firm characteristics such as market equities or book-to-market ratios.

The initial state corresponds to CAPM: a single leaf basis portfolio with 100% weight. When building the tree model with additional splits, we update $\{R_{t+1}^{(k)}, W_t^{(k)}, \beta_{i,t}^{(k)}\}$ through an iterative scheme. First of all, leaf basis portfolios, $R_{t+1}^{(k)}$, are expanded by the continuous asset pricing factor model improvement. Second, the SDF weights, $W_t^{(k)}$, are estimated by the expanded leaf basis portfolios. Finally, time-varying factor exposures, $\beta_{i,t}^{(k)}$, are updated by the new characteristics data and updated SDF when fitting the cross-section.

2.2 Splitting the Cross-Section

In the following section, we explain the algorithm step-by-step. Algorithm 1 summarizes the tree growing algorithm using pseudo-code. A tree is a set of cutpoints defining a rectangular partition of the variable space. Each cutpoint is a pair of variable index and value, indicating variable

[7] Please see Avramov (2004), Kelly et al. (2019), and Feng and He (2019).

to split and the value it cuts. Tree consists of multiple nodes. The final nodes are called terminal or leaf nodes, and the intermediate nodes are called internal nodes. The criterion of how to find the cutpoints is essential for a tree model. In this section, we demonstrate the crucial steps of growing an asset pricing tree.

Let $R_t^{(k)}$ denote the return of the *basis portfolios* after the k-th leaf node of the tree, which can be equal or value weighted portfolio of the assets in that specific leaf. The number of basis portfolios increase by one after each iterative split. So, there are $k+1$ basis portfolios after the k-th split, and $R_t^{(k)} = [R_{1,t}^{(k)}, R_{1,t}^{(k)}, \cdots . R_{k+1,t}^{(k)}]^\intercal$.

In the beginning, the entire cross-section of stock returns are in the top node of the tree, named the *root* node $R_t^{(0)}$. We consider several cutpoint (a pair of firm characteristics index and value to split at) candidates and pick the one that optimizes the split criterion. All firm characteristics are normalized cross-sectionally in the range from -1 to 1. Instead of searching for hundreds of cutpoints, we only consider the cross-sectional quintiles such as -0.6, -0.2, 0.2, and 0.6. This choice follows the conventional security sorting and reduces the computational time.

Each cutpoint candidate partitions the root node to left and right child node consisting a subset of assets and months of the data. Each potential leaf can form a leaf basis portfolio, denoted by $R_{1,t}^{(1)}$ and $R_{2,t}^{(1)}$. Since there are only two leaf nodes, the SDF is estimated as a mean-variance efficient portfolio[8] of the two leaf basis portfolios,

$$f_t^{(1)} = w^{(1)} R_t^{(1)}, \quad w^{(1)} = \widehat{\Sigma}_1^{-1} \widehat{\mu}_1 \tag{9}$$

where $\widehat{\Sigma}_1^{-1}$ and $\widehat{\mu}_1$ are the covariance matrix and average returns of two leaf portfolios $R_t^{(1)} = [R_{1,t}^{(1)}, R_{2,t}^{(1)}]^\intercal$. The split criterion of the cutpoint candidate c_k is defined as the loss of the SDF model

[8]On a separate note, to deal with the estimation error in the sample mean and sample covariance matrix estimation for the mean-variance efficient portfolio, we add two small regularization parameters $\lambda_\Sigma = 10^{-4}$ and $\lambda_\mu = 10^{-4}$ in Equation 5. These two shrinkage parameters help to stabilize the portfolio weight estimation and avoid over-leveraging. The similar regularized portfolio optimization problem is also addressed in Kozak et al. (2020) Equation (18) and (22), and Bryzgalova et al. (2020).

$$W_t^{(k)} = \left[E_t\left(R_{t+1}^{(k)} R_{t+1}^{(k)\intercal}\right) + \lambda_\Sigma I_{k+1}\right]^{-1} \left[E_t\left(R_{t+1}^{(k)}\right) + \lambda_\mu \mathbf{1}\right]. \tag{8}$$

as follows

$$\mathcal{L}(c_k) = \sum_{t=1}^{T} \sum_{i=1}^{N_t} (r_{i,t} - \beta(z_{i,t-1}) f_t)^2 \tag{10}$$

where $\beta(z_{i,t-1}) = b_0 + b^\intercal z_{i,t-1}$ are conditional factor loadings on the past firm characteristics. We estimate the regression coefficients by the ordinary least squares estimator, using a pooled regression model for individual stock returns on f_t and $f_t \times z_{i,t-1}$ without an intercept. The split criterion of equation (10) is evaluated at all cutpoint candidates $c_k \in \mathcal{C}$, and the one minimizes split criterion is picked as the first cutpoint.

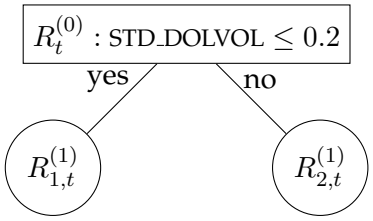
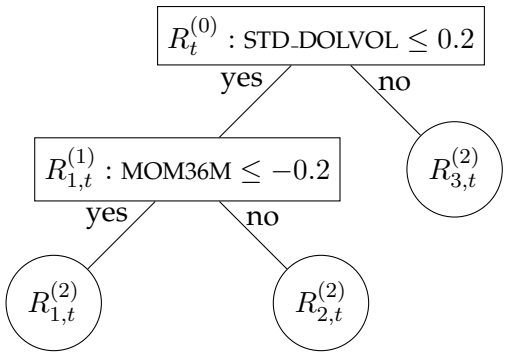
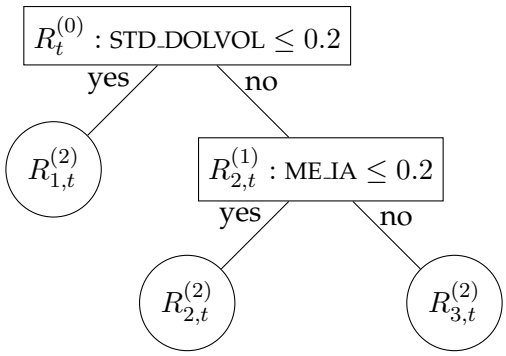


Figure 1: Demonstration of the first split.



(a) If split node $R_{1,t}^{(1)}$ at MOM36M.



(b) If split node $R_{2,t}^{(1)}$ at ME_IA.

Figure 2: Demonstration of the cutpoint candidates for the second split.

Next, we proceed to the second cutpoint. Note that there exist two leaf nodes after the first split. The second split could happen at either the left or right child node of the root. We take an iterative approach to grow the tree. In specific, the potential split at the left or right leaf node are considered simultaneously. We evaluate the split criterion for all cutpoint candidates at two nodes and pick the one optimizing the criterion. Figure 2 depicts the tree of the cutpoint candidates for the second split. In both cases, one leaf node is split, becomes an internal node, and creates two new leaf nodes. The SDF will be evaluated based on the three basis portfolios.

$$f_t^{(2)} = w^{(2)} R_t^{(2)}, \quad w^{(2)} = \widehat{\Sigma}_2^{-1} \widehat{\mu}_2 \tag{11}$$

where $R_t^{(2)} = [R_{1,t}^{(2)}, R_{2,t}^{(2)}, R_{3,t}^{(2)}]$. The graphical position of the three basis portfolios depends on the which node to split, as shown in Figure 2. Similary, $\widehat{\Sigma}_2^{-1}, \widehat{\mu}_2$ are covariance matrix and average return of the basis portfolios $R_{2,t}$.

The next split point proceeds similarly. For the k-th split, each cutpoint candidate of the existing k leaf nodes creates $k+1$ leaf basis portfolio, which creates the SDF $f_t^{(k)}$ and hereby to evaluate the split criterion of equation (10). The stopping conditions are pre-specified as the total number of iterations, max depth of the tree, or the minimum number of data observations in a node, whichever is met first. The complete algorithm is summarized in Algorithm 1.

There are several important differences between P-Tree and the standard tree models in machine learning, such as CART (Breiman et al., 1984). First, each leaf of CART is associated with a constant (leaf parameter), which predicts new data falls in that leaf. CART aims to approximate a function by step function represented by the tree. By contrast, the P-Tree has a clear economic objective. Each tree partitions the entire cross-section of stock returns to multiple leaf basis portfolios and creates the SDF. Second, the CART tree grows recursively, where each split is selected based on the data at a particular node without looking at other branches of the tree. This approach is greedy since it optimizes a local split criterion at each node and overfits the data. On the other hand, P-Tree grows iteratively: when considering each cutpoint, P-Tree looks at *all* leaf basis portfolios. The split criterion is defined globally as the pricing loss of the generated SDF for the returns of *all* asset, regardless of whether the data is in the current node or not.

Algorithm 1 The main algorithm that grows the asset pricing tree from the data.

procedure GROWTREE(y, $\mathbf{X}$, Φ, Ψ, d, T, node)
outcome Grow the tree T and find corresponding basis portfolios
    for i from 1 to num_iter **do** ▷ Loop over number of iterations
        if the stopping conditions are met **then**
            return.
        else
            Search the tree, find all leaf nodes $\mathcal{A}$
            for each leaf node A_j in $\mathcal{A}$ **do** ▷ Loop over all current leaf nodes
                for each cutpoint candidate c_k in $\mathcal{C}$ **do** ▷ Evaluate all cutpoint candidates
                    Partition data in A_j according to c_k.
                    if Either left or right child of A_j does not satisfy minimal leaf size **then**
                        Ignore this cutpoint candidate.
                    else
                        Find the SDF based on all leaf portfolios as in equation (9).
                        Calculate the split criterion in equation (10).
                    end if
                end for
            end for
            Find the leaf node to split and cutpoint c_k that minimizes the split criterion (10).
            Split the node selected at cutpoint c_k. ▷ Split one node at one iteration
        end if
    end for
    return
end procedure

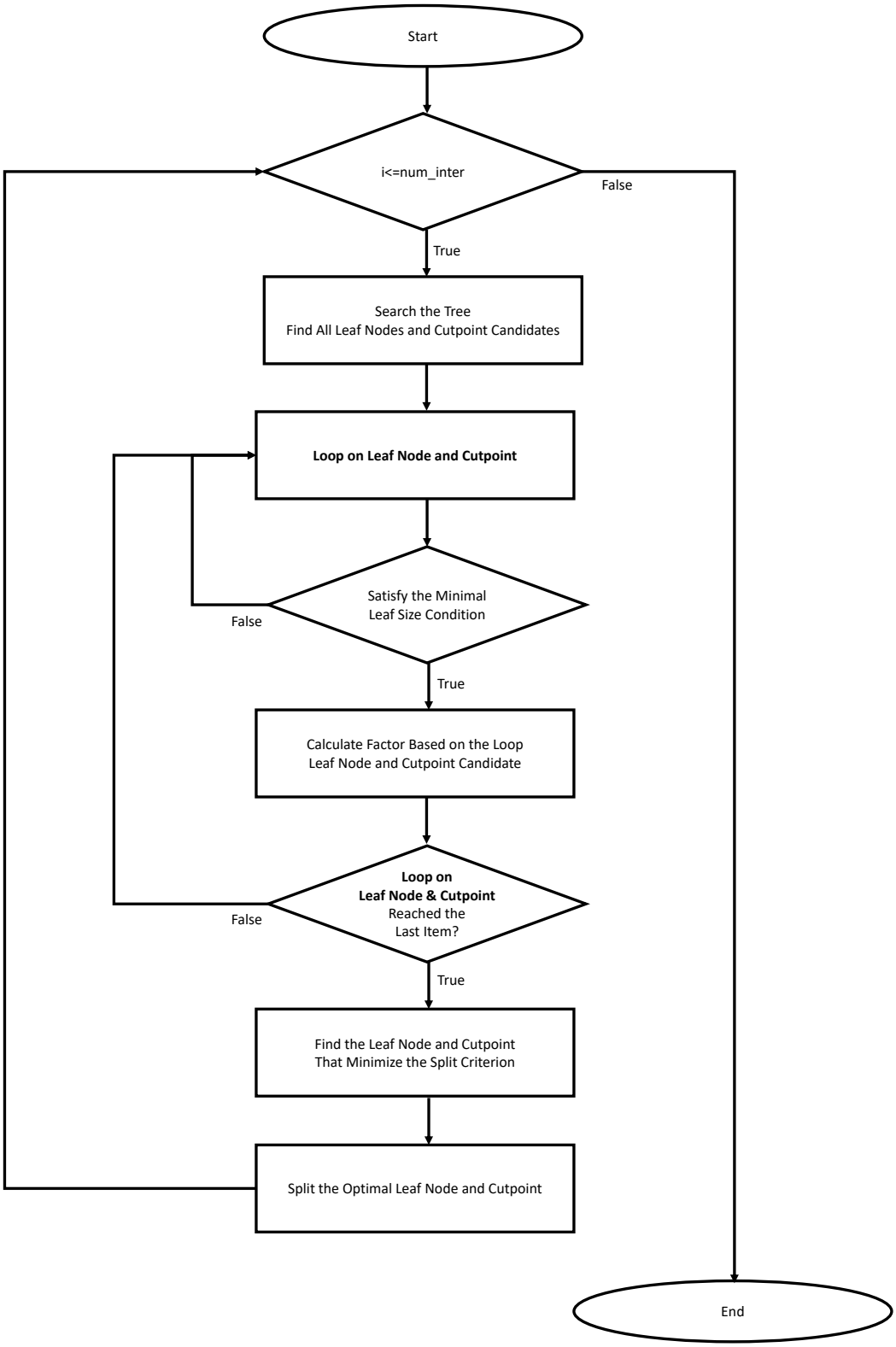


Figure 3: Flowchat of the Algorithm

2.3 Model Extensions

This section demonstrates various extensions of the plain tree model in section 2.2. The ensemble method is a common strategy to improve the performance of a single tree model in machine learning. Rather than growing a single tree, ensemble methods grow multiple trees either independently or dependently. Random forest and boosting are two popular tree ensemble methods with completely different intuition. For a standard textbook treatment, see Friedman et al. (2001). Section 2.3.1 and 2.3.2 extend the P-Tree to Random Forests and Boosting respectively.

Furthermore, P-Tree is flexible enough to split data observation cross-sectionally and on the time series. Section 2.3.3 discusses the time series split of a tree. Finally, 2.3.4 concludes the section with extension of interaction factors.

2.3.1 Variable Significance by Random Forest

The plain Random Forests (Breiman, 2001) grow a number of trees on bootstrapped training data samples, which takes random samples of the observations as well as variables from the complete training set. The forecast of the entire ensemble is the average of all "decorrelated" trees, which usually has a smaller variance than that of a single tree. We want to answer the question, "Which variable is more important to create the SDF". Therefore, we apply the similar bootstrap idea from Random Forest to P-Tree, train various "decorrelated" P-Tree trees on bootstrapped subsamples. In the empirical studies, we focus on the out-of-bag (OOB) variable importance. Next, we define two measurements to calculate the characteristics' importance.

The first measurement of importance is based on the frequency of being selected for a split. Intuitively, a characteristic variable is more often selected as tree cutpoints, and then it is more important. For each bootstrapped subsample, a subset of characteristics is randomly drawn to build the tree, and only a fraction of them are actually used to split. In specific, we create 500 trees and count the number of times a particular characteristic z_i being used in the first K splits and the total number of appearances in bootstrapped subsamples. The measurement of importance is

defined as the ratio as follows

$$\text{Selection Probability}(z_i) \quad = \quad \frac{\#(z_i \text{ is selected at first K splits})}{\#(z_i \text{ appears in the bootstrapped subsamples})} \tag{12}$$

Another interesting measurement besides the selection probability is a "treatment effect", like evaluating the characteristics. For the 500 trees grown on bootstrapped subsamples, we force half of them to use the characteristic but the other half not. Note that even if one characteristic is included in the subsample, it is not guaranteed to be selected as cutpoints when building the tree. The with- and without- sampling scheme for a particular characteristic creates the treatment effect evaluation and allows one to perform the significance test of its importance. We compute the asset pricing model fitting performance, the total R^2 (Equation 18), for this treatment effect evaluation.

$$\text{Char. Importance} \quad = \quad \left[\frac{E(\text{P-tree fit} \mid \text{with char}_i)}{E(\text{P-tree fit} \mid \text{without char}_i)} - 1 \right] \times 100 \tag{13}$$

Next, we detail the bootstrap sampling scheme for the panel data of individual stock returns. First of all, we take subsample in the time horizon, and for the selected period, the complete cross-section of the panel data is preserved. Second, we randomly draw ten characteristics out of the 61 for building each tree. The bootstrap procedure repeats 500 times to form a forest. In our empirical application, we mainly demonstrate the variable importance by the Random Forests P-Tree.

2.3.2 Multiple Factors by Boosting

Boosting (Freund and Schapire, 1997) creates tree ensembles from another perspective. It combines a group of weak learners for better fitting and prediction. It grows a list of trees iteratively where each tree fits the residual of all previous trees. They all form a strong learner, and the final prediction is the (weighted) sum of each tree outcome.

The P-Tree uses the same strategy to build the sum of the trees. Specifically, the boosting P-Tree iteratively creates additional factors to fit the unexplained pricing errors of all previous factors. Applying the boosting scheme to P-Tree establishes a sequence of factors to form the SDF, and it extends to a multi-factor model. The following of this section shows the boosting P-Tree.

1. The first factor $f_{1,t}$ is generated by the standard P-Tree on excess returns $\{r_{i,t}\}$, as discussed in section 2.2. The residual of the first factor is $\epsilon_{i,t}^{(1)} = r_{i,t} - \beta(z_{i,t-1})f_{1,t}$, where the regression coefficients are the OLS estimator.

2. The second factor generated from fitting the residual $\epsilon_{i,t}^{(1)}$ instead of the original return $r_{i,t}$. The tree growing steps are the same as section 2.2 except replacing the split criterion of equation (10) by the following one

$$\mathcal{L}(c_k) = \sum_{t=1}^{T}\sum_{i=1}^{N_t}\left(\epsilon_{i,t}^{(1)} - \beta_2(z_{i,t-1})f_{2,t}\right)^2, \tag{14}$$

   Again, the new residual is defined as $\epsilon_{i,t}^{(2)} = \epsilon_{i,t}^{(1)} - \beta_2(z_{i,t-1})f_{2,t}$.

3. The third factor $f_{3,t}$ proceeds similarly to fit the residual of the first two trees $\{\epsilon_{i,t}^{(2)}\}$, with the corresponding split criterion

$$\mathcal{L}(c_k) = \sum_{t=1}^{T}\sum_{i=1}^{N_t}\left(\epsilon_{i,t}^{(2)} - \beta_3(z_{i,t-1})f_{3,t}\right)^2, \tag{15}$$

4. Repeating the procedures above K times, each tree fits the residual of all previous trees to generate all K factors.

For example, in the section of empirical study, we set the number of factors $K = 3$. The three generated factors can be listed with decreasing importance order as $[f_{1,t}, f_{2,t}, f_{3,t}]$. Generating additional tree factors is similar to generating additional components in the principal component analysis. Below is the three-factor model with time-varying factor loadings. Any general application of this three-factor model requires re-estimating factor loadings.

$$E(r_{i,t}) = \widehat{\beta}_1(z_{i,t-1})f_{1,t} + \widehat{\beta}_2(z_{i,t-1})f_{2,t} + \widehat{\beta}_3(z_{i,t-1})f_{3,t} \tag{16}$$

Another natural application of the boosting scheme is to create an augmented factor model. One can start with a commonly used benchmark model, such as CAPM or Fama-French 3-factor model, using the boosting P-Tree to fit the benchmark model's pricing errors directly. Tree factors

developed beyond the benchmark model are supposed to explain "orthogonal" information from factor model residuals. In the section of empirical studies, we show results of two versions of multi-factor P-Tree with and without the CAPM benchmark.

2.3.3 Splitting the Time-Series and Regime Switching

The previous sections focus on the cross-section splits, i.e., all nodes in the tree split at the firm characteristics $z_{i,t}$. As the tree grows from top to bottom, the panel of stocks is split into many leaf nodes, and each leaf node maintains the entire time series from period 1 to T. Intuitively, the latent factors and/or factor loading functions may substantially differ under two macroeconomic states (i.e., high and low inflation or interest rate states). There is a long literature that discuss about the regime changes in the financial market, see Ang and Timmermann (2012).

In our P-Tree, although the factor loadings are time-varying based on the firm characteristics, the latent factor generation is not. A natural solution is to create different factor models and estimate corresponding loadings under different macroeconomic states. With the panel data of individual stock returns, it is clear that the cutpoint of some nodes (for example, the root node) can be replaced by certain macroeconomic variables instead of the firm characteristics. The P-Tree is flexible enough to incorporate the time series split in addition to the cross-section split. We only need to define a proper split criterion for macro states.

Given the short history length of individual stock returns comparing to the vast cross-section, we only implement one time-series split of the data at the root node (i.e., the first cutpoint of the tree). The P-Tree model will search over all possible cutpoint candidates of the macro variables and choose the one optimizes the factor pricing error. In specific, the split criterion of the first time-series split is defined as

$$\mathcal{L}(c_k) = \sum_{t=1}^{N_A}\sum_{i=1}^{N_t}\left(r_{i,t} - \beta_A(z_{i,t-1})f_{A,t}\right)^2 + \sum_{t=1}^{N_B}\sum_{i=1}^{N_t}\left(r_{i,t} - \beta_B(z_{i,t-1})f_{B,t}\right)^2 \tag{17}$$

where the cutpoint candidate c_k partitions the time series of data to state A and B, for example, high or low inflation. Each state has number of months N_A and N_B correspondingly. Comparing to the cross-section split criterion in equation (10), the time series split criterion is the *total* pricing loss

of two time periods, with two corresponding factors $f_{A,t}$ and $f_{B,t}$.

After searching for the optimal time series cutpoint, all following growing procedure only evaluates the cross-section cutpoints. Note that any further split on either child of the root node only depends on stock returns observations on one side. The split criterion is no longer iterative to the first split but still works for the entire time series on the one side. Our empirical analysis finds that Inflation is the most important macroeconomic variable that differentiates factor models in two different states. Details are postpones to section 3.

2.3.4 Interaction of Factors

The tree model provides an alternative solution to factor construction. For example, a two-layer tree splits the cross-section for two up and down portfolios. With the economic theory implied long-short direction, one can construct a univariate factor using these two leaf basis portfolios. In practice, researchers typically use quintile or decile sorted portfolios to create long-short factors with higher return spreads, but the sorting idea applies the same. Researchers might also apply bivariate or triple-way sort to include the interaction of the characteristics when creating long-short factors.

Our P-Tree searches the optimal characteristics for interaction for the second and probably third splits when growing the tree with the asset pricing goal. Building a tree model for three layers makes it possible to have different interactions on the long and short legs from the first split. Our empirical analysis finds these interaction factors produce a tremendous improvement over the univariate sorted factor without interactions. These are the unique empirical evidence for nonlinearity from the tree model, which other machine learning methods cannot be found.

3 Empirical Results

3.1 Data Sample

The data observations range from January 1981 to December 2020. The construction of individual stock universe filtering follows that of Fama-French factor as follows: (1) choose only stocks listed on NYSE, AMEX, or NASDAQ are included; (2) we use those observations for firms with a

CRSP share code of 10 or 11; (3) include only stocks listed for more than one year; (4) stocks with negative book equity or negative lag market equity are excluded.

We take 61 firm characteristics listed in Table A.1, covering six major categories: momentum, value, investment, profitability, frictions (or size), and intangibles. The calculation of firm characteristics follows Feng et al. (2020), only differes in that the data is updated in monthly frequency since we choose to adopt a monthly sorting scheme. The monthly characteristics of firms are standardized cross-sectionally in the range $[-1, 1]$.[9] This cross-sectional data standardization is useful when we construct regression trees that work for multiple periods.

In addition, ten macroeconomic variables are selected for the empirical extension to splitting time-series. The macro predictor list follows Feng and He (2019) and are summarized in Table A.2, which includes market timing macro predictors, bond market predictors, and aggregate characteristics for S&P 500. We standardize these macro predictor data by the historical percentile numbers for the past ten years. For example, inflation greater than 0.7 implies that the current inflation level is higher than 70% of observations during the past decade. This rolling window data standardization is useful when we compare the predictor level to detect different macroeconomic regimes.

For the train-test sample design, we use the 20-year period from 1981 to 2000 for training and the 20-year period from 2001 to 2020 for testing. The average and median monthly numbers of stocks are 4,667 and 4,450 in the train sample, and 3,953 and 3,791 in the test sample. Unlike most machine learning methods, our P-Tree is flexible to handle such an imbalanced data panel.

3.2 Multi-Period Asset Pricing Tree

We apply P-Tree to generate leaf basis portfolios and the stochastic discount factor to fit cross-sectional individual stock returns. We plot the tree in Figure 4. The tree structure shows the splitting orders S#, selected splitting characteristics, and the cross-sectional cutpoints. Before the first split, the tree grows from the root node, representing the market portfolio for all stocks (N1). The leaf basis portfolios at each layer of the tree are numbered with N#. In Figure 4, the tree first splits on

[9] For example, the market equity in 2019 December is uniformly standardized in the range of $[-1, 1]$. The firm with the lowest market equity is -1, and the firm with the highest market equity is 1, all others distributed uniformly in between. Therefore, this uniform standardization transforms the data onto $[-1, 1]$ every month. If a firm has missing values for some characteristics, the imputed values are 0, implying the firm is not important in security sorting.

the trading volume volatility (STD_DOLVOL) at 0.2 (60% quantile). After the first split, 60% of firms are moved to the left leaf (labeled N2), and 40% are moved to the right one (N3).

The second split is implemented on the market-adjusted market equity (ME_IA) at 0.2 on the right leaf (N3), while the third split is on 3-year long-term reversal (MOM36M) on the left one (N2). The split criterion allows one to evaluate and visualize the interaction of characteristics. ME_IA is useful for high STD_DOLVOL stocks, while MOM36M is useful for low STD_DOLVOL stocks. The tree stops growing[10] after 22 splits and generates 23 leaf basis portfolios. The numbers printed in the leaves are the median number of stock observations in the monthly updated leaf basis portfolios. Because our P-Tree focuses on fitting a multi-period tree model, numbers of stocks for these leaf basis portfolios are reasonable for asset pricing studies.

Compared to other machine learning methods, the tree model provides a precise mapping for generating these characteristic-managed portfolios. The corresponding partition plot is in Figure 5. We print the average return and Sharpe ratio for each leaf basis portfolio. These performance gaps show the usefulness of splitting the cross-section via the interaction of characteristics. The top right plot in Figure 5 further splits N2 by STD_DOLVOL and MOM36M. The bottom plot splits N6 by ATO. Section 3.6 shows an application for exploiting these characteristics interaction: interaction factors.

We can also see the clustering patterns for these leaf basis portfolios for further splits along the tree. The market portfolio is split into four portfolios (N4 - N7) at the tree depth three, each group of individual stocks with similar characteristic exposures. The return-risk relationship of the four portfolios as well as their mean-variance efficient portfolio (MVE3) is plotted in Figure 6. Further splitting these four portfolios, we have 23 leaf basis portfolios at the tree depth 6. We find those great-grandchild leaf basis portfolios (in a light color) labeled with (N4+ - N7+) cluster around their great grandparent nodes (in dark color). The mean-variance efficient portfolio from a 6-depth tree produces a higher Sharpe ratio than a 3-depth tree, while both are higher than the market portfolio. These performances are robust for the out-of-sample plot in Figure 7.

[10]For asset pricing studies, one natural tuning parameter to limit the tree size and avoid over-fitting is the minimum number of stocks in the leaf basis portfolio. We set 50 when growing this tree.

3.3 Asset Pricing Performance

We calculate XS-R^2 for evaluating the beta-pricing model performance, see Bryzgalova et al. (2020). We also follow Kelly et al. (2019) and Feng et al. (2020) to include Total R^2 and Predictive R^2, which are designed for measuring statistical model fitness. The first metric is generally suitable to evaluate any portfolio, while the second and third ones can be used to evaluate model performance on individual stocks conditional on firm characteristics.

1.
$$\text{Total } R^2 = 1 - \frac{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(r_{i,t} - \widehat{r}_{i,t}\right)^2}{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} r_{i,t}^2}, \tag{18}$$

where $\widehat{r}_{i,t} = \beta(z_{i,t-1})f_t$. Total R^2 represents the fraction of realized return variation explained by the model-implied contemporaneous return, aggregated over all assets and all periods.

2.
$$\text{Predictive } R^2 = 1 - \frac{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(r_{i,t} - \widehat{r}_{i,t}\right)^2}{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} r_{i,t}^2}, \tag{19}$$

where $\widehat{r}_{i,t} = \hat{\beta}(z_{i,t-1})\lambda_f$. The risk premia λ_f is the historical average of the tradable factor. Predictive R^2 summarizes the predictive performance by the model-implied return forecasts.

3.
$$\text{XS-}R^2 = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}\alpha_i^2}{\frac{1}{N}\sum_{i=1}^{N}\bar{R}_i^2}, \tag{20}$$

where α_i is from the times series regression of portfolio i's excess returns on a factor model, and $\bar{R}_i$ is the average excess return of the portfolio. It measures the relative magnitude of pricing errors in a given cross-section.

P-Tree models are competitive in terms of modeling individual stock returns. In Table 1, we summarize the performance for pricing the individual stocks. Our models are reported in panel A for Panel Tree factors (P-Tree), and panel B for market factor expanded by Panel Tree factors (MKT+P-Tree). Panel C and D are discussed later for time-series split model. Panel E is for other factor models for comparison.[11] A higher total R^2 means smaller pricing error for the individual

[11]The other models include CAPM, Fama-French three-factor model Fama and French (1993), Fama-French five-factor model Fama and French (2015), Q factors Hou et al. (2015), IPCA Kelly et al. (2019), and RPPCA Lettau and Pelger (2020).

stock. The P-Tree one-factor model out-performs market factor in terms of total R^2. And, P-Tree3 or P-Tree5 beats Fama-French three or five factor model. For the recent latent factor models, P-Tree does better than RPPCA, but no better than IPCA models.

We investigate the asset pricing performance for pricing the average returns of portfolios. The three groups of test assets are Fama-French size-value 25 portfolios, the Industry 49 portfolio, and the 23 portfolios generated by P-Tree. [12] We run a time series regression for each asset, and get its beta. To get accurate estimations for beta's, we make the sample period as long as possible, which is from 1981 to 2020. The pricing error measure is the XS-R^2 in Equation 20. We find the Fama-French three and five factor models, and Q4 factors have strong performance in explaining the average returns in three groups of test assets. The latent factor models seem to be weaker, but P-Tree and RPPCA give reasonble results. The IPCA model doesn't result in a positive XS-R^2 for the given portfolios.

P-Tree factors can be traded and the gain is huge. In Table 2, we develop two investment strategies for each factor model. The Mean-Variance-Efficient portfolio is spanned by the factors and the 1/N case is the equally weighted portfolio of the factors.[13] The table reports the average returns, Sharpe ratio, Jensen's alpha, and the t-statistic of alpha. The MVE strategy in panel B provides 4.89% monthly average return, 4.66% Jensen's alpha, and 3.62 annualized Sharpe ratio, and the strong performance is consistent for the test sample. The 1/N strategy in panel B provides 2.20% monthly average returns, 1.56% alpha, 1.72 annualized Sharpe ratio, and strong for the test sample. The investment gain is very high compared to all the other models in panel E.

We regress each factor on the prevailing factor models to distinguish between our factor models and the other factor models. The time series regression intercepts (alpha) are summarized in Table 3. We find almost all of our factors cannot be fully explained by the listed factor models statistically. And, we see big alpha numbers, which means our factors earn significant expected returns even controlling other factor models. The row names are the top two spitting characteristics of the corresponding tree structure, highlighting the interaction among characteristics. The interaction of characteristics makes our factors different from the Fama-French factors. We report the MVE

[12] The Fama-French size-value 25 portfolios and the Industry 49 portfolio returns can be downloaded from https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html. The 23 portfolios constructed by P-Tree are visualized in Figure 4.

[13] The MVE weight on each factor follows Equation 8, and then re-scaled to sum to one.

spaned by Fama-French 25 portfolio and Industry 49 portfolio for comparison in panel C.

3.4 Significant Splitting Characteristics

What are the significant characteristics that improve asset pricing performance for the regression tree? The characteristic significance is summarized in the bar plots of Figure 8. Referring to Equation 13, a negative number on the bar name means our model does better in pricing individual stock returns when we include that characteristic than ignoring it. A deep color negative bar indicated the benefit of that characteristics is significant at 5% level, via two-sample t test. The in-sample results show the significant characteristics are volume volatility (STD_DOLVOL), long-term momentum over 60 months (MOM60M), market-adjusted market equity (ME_IA), and seasonality over a 1-year horizon (SEA1A). For the out-of-sample analysis, three characteristics stand out. They are market-adjusted market equity (ME_IA), trading volume (DOLVOL), and volume volatility (STD_DOLVOL).

In Table 4 panel A, we summarize the selection frequency of characteristics in the top splits, referring to Equation 12. We find volume volatility (STD_DOLVOL) has a 73% chance of the first splitting characteristic if a sample contains it. The other important characteristics are long-term momentum over 60 months (MOM60M), trading volume (DOLVOL), and market-adjusted market equity (ME_IA). The results of two different characteristic significance measures agree on a few characteristics, which concentrate on the categories of frictions (or size) and momentum.

3.5 Splitting the Time Series

Time series split makes P-Tree more flexible. Statistically, P-Tree varies conditional on different regimes. Economically, P-Tree adapts to different macroeconomic conditions with different cross-sectional structure. The cutpoint candidates are the 40%, 50%, and 60% quantile values of the macroeconomic variable time series. After greedy searches among the ten macroeconomic variables and the cutpoints, our model cuts inflation at 50% quantile, based on a sample from 1981 to 2000. We get a high inflation period and a low inflation period. The time series split tree structure is in Figure 9. We see the left and right branches hold different tree structure, which means the tree model adapts to different economic conditions and generate different basis portfolios.

The asset pricing performance of time series split model is summarized in panel C and D of Table 1, and the investment gain is reported in Table 2. Compared to the models without time-series split in panels A and B, the five-factor model in panel C gives a higher total R^2 and investment gain. Adding time series split, the total R^2 of P-Tree5 increase from 11.63 to 12.09, and the predictive R^2 grows from 0.37 to 0.57. The Sharpe ratio of MVE on P-Tree5 increase from 3.24 to 3.59. However, we face a trade-off in time-series splitting. The time-series split model has only a half-length time index than the original model, which enlarges the estimation error in SDF weight when the model searches among characteristic and cross-sectional cutpoints. [14] So, there is no guarantee that adding time-series dimension would improve the model performance. We find the MVE Sharpe ratio of MKT+P-Tree4 decrease from 3.62 to 1.97, after adding time series split.

The time series split characteristic importance is reported in Figure A.1 The time series split variable importance by selection probability is in Table 4. We find the high and low inflation periods have different characteristic importance, but the important characteristics concentrate on the categories of frictions (or size) and momentum in both periods.

3.6 Interaction Factor

Our tree model explores interaction among characteristics. We find interaction helps increase the premium of some characteristics-based factors, compared to the commonly used univariate sorting. We collect the long-short direction of each characteristic from literature. Then we compare the following four specifications of long-short portfolios for each characteristic, and the graphical demonstration is in Figure A.2. Let's name the characteristic of interest as A.

1. Uni-Sort 4x1: We split the cross-section of stocks by quartile values of A.[15] We get four portfolios, then long 1 short 4 or long 4 short 1 based on the long-short direction of A.

2. Uni-Sort 2x1: We split the cross-section once into two halves, splitting at zero. Referring to the direction of A, we make decision to long one portfolio and short the other.

3. Interaction: Repeat specification 2. For the depth-two nodes, we split once on the long-A

[14] For time-series split, the left/right branch has a smaller sample in terms of time series dimension, so we add larger regularization on the SDF weight in Equation 8. We set $\lambda_\Sigma = 10^{-3}$ and $\lambda_\mu = 10^{-3}$.

[15] We have standardized the characteristics into range $[-1, 1]$, so the quartiles are -0.5, 0, and 0.5.

branch, then split once on the short-A branch. The second and third splits are driven by our tree model pricing kernel, searching among all the characteristics except A, thus there are interactions between A and others. Let the long-A branch split on B and the short-A branch split on C. Our final long position is long-A-long-B, and the final short position is short-A-short-C. This is a long-short strategy on four basis portfolios, and comparable to the 4x1 sorting in specification 1.

4. Market-Adjusted Interaction: Use market factor to price all the assets and get the residuals. Repeat specification 3, but the pricing kernel fits the residuals, instead of excess returns of the assets. This specification takes the idea of Boosting. Let the long-A branch split on D and the short-A branch split on E. Our final long position is long-A-long-D, and the final short position is short-A-short-E.

3.6.1 Overall Comparison of Four Specifications

The (Market-Adjusted) Interaction factors are fitted by P-Tree with data from 1981 to 2000. We summarize the number of significantly positive cases of average returns and Jensen's alpha's in Table 5 panel A and B. First, the stock market is more efficient in 2001-2020 than 1981-2000, as we observe fewer significant alphas recently for the commonly used Uni-Sort 4x1 specification. Second, in the recent 20 years, Interaction factors have more significant expected returns and alphas than Uni-Sort 4x1 factors, and Market-Adjusted Interaction factors have more significant alphas than Uni-Sort 4x1 factors. The Interaction cases do better in the average returns. For comparison, the Market-Adjusted cases do better in the market-adjusted performance, the Jensen's alphas. Overall, interactions among characteristics strengthen the Uni-Sort 4x1 factors.

In Table 5 panel C, we report the quantile numbers of the average returns and Jensen's alpha's. We find the Uni-Sort 4x1 factors give very high average returns and Jensen's alpha's from 1981 to 2000. However, the scale of the factors decreases for 2001-2020. The Interaction factors have much higher average returns, and Jensen's alpha's in the recent 20 years than the previous 20 years. The Market-Adjusted factors have persistent alpha's both in- and out-of-sample. In summary, (Market-Adjusted) Interaction factors are more robust in the recent 20 year than Uni-sort 4x1 factors.

3.6.2 Examples of Interaction Factors

We find the (Market-Adjusted) Interaction specification is exceptional for some characteristics. In Table 6 and Figure A.3, six examples are reported. For a Uni-sort 4x1 factor, which is profitable, we find the interaction increases its scale. The Standardized Unexpected Earnings (SUE) Uni-sort 4x1 factor is persistently strong. Our model doubles the SUE premium by the interaction with R&D to Market and Market Equity. Interaction can make a non-profitable Uni-Sort 4x1 factor become profitable. We find the Uni-sort 4x1 of Profit Margin has almost zero premium. Interacting with R&D to Sales and Dollar Trading Volume, the Profit Margin factor earns 47 basis point monthly and negatively hedges with the market factor, thus has a 51 basis point Jensen's alpha.

Another interesting finding is that some characteristics which fail alone can be profitable when they interact. In panels, B, D, and E, we find Profit Margin, Dollar Trading Volume, and R&D to Sales don't have significantly positive returns. However, when the three characteristics are put together and interact, they earn positive returns and alpha's, as shown in Table 6 panel B.

4 Summary

We provide a Panel Tree (P-Tree) model framework, which complements the traditional CART model with i.i.d. assumption. Considering the panel structure of economic data, P-Tree partitions the cross-section and generates a time-series latent factor that could explain the cross-section. Also, our model is flexible to split in the time-series dimension in the first step, which adapts our P-Tree under different economic conditions. More extensions are discussed, including boosting, random forest, and feature interaction. Finally, we apply P-Tree to explain the cross-sectional variation of U.S. stock returns and find competitive asset pricing performance and investment gain.

On the application side, the P-Tree framework is flexible. Researchers can customize the split criterion with an economic objective. We welcome more applications of P-Tree in economic studies. On the model side, P-Tree have many potential directions to be developed. The current model only consider time series split in the first step, future works can integrate the time series split and cross-sectional split in split criterion for every split. Also, a regularized version of P-Tree and Bayesian P-Tree are promising extensions.

References

Ang, A. and A. Timmermann (2012). Regime changes and financial markets. *Annu. Rev. Financ. Econ. 4*(1), 313–337.

Avramov, D. (2004). Stock return predictability and asset pricing models. *The Review of Financial Studies 17*(3), 699–738.

Avramov, D., S. Cheng, and L. Metzker (2021). Machine learning versus economic restrictions: Evidence from stock return predictability. Technical report, Hebrew University of Jerusalem.

Bali, T. G., D. Huang, F. Jiang, and Q. Wen (2020). The cross-sectional pricing of corporate bond using big data and machine learning. Technical report, Georgetown University.

Bianchi, D., M. Büchner, and A. Tamoni (2018). Bond risk premia with machine learning. Technical report, University of Warwick.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and regression trees*. Routledge.

Bryzgalova, S., M. Pelger, and J. Zhu (2020). Forest through the trees: Building cross-sections of stock returns. Technical report, London Business School.

Chen, L., M. Pelger, and J. Zhu (2020). Deep learning in asset pricing. Technical report, Stanford University.

Cong, L. W., K. Tang, J. Wang, and Y. Zhang (2020). Alphaportfolio for investment and economically interpretable ai. Technical report, Cornell University.

Cong, L. W., K. Tang, J. Wang, and Y. Zhang (2021). Deep sequence modeling: Development and applications in asset pricing. *The Journal of Financial Data Science 3*(1), 28–42.

Creal, D. and J. Kim (2021). Empirical asset pricing with bayesian regression trees. Technical report, University of Oklahoma.

DeMiguel, V., A. Martin-Utrera, F. J. Nogales, and R. Uppal (2018). A portfolio perspective on the multitude of firm characteristics. Technical report, London Business School.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics 33*(1), 3–56.

Fama, E. F. and K. R. French (2015). A five-factor asset pricing model. *Journal of Financial Economics 116*(1), 1–22.

Feng, G., A. Fulap, and J. Li (2020). Real-time macro information and bond return predictability: Does deep learning help? Technical report, City University of Hong Kong.

Feng, G. and J. He (2019). Factor investing: Hierarchical ensemble learning. Technical report, City University of Hong Kong.

Feng, G., N. Polson, and J. Xu (2020). Deep learning of characteristics-sorted factor models. Technical report, City University of Hong Kong.

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences 55*(1), 119–139.

Freyberger, J., A. Neuhierl, and M. Weber (2019). Dissecting characteristics nonparametrically. *The Review of Financial Studies, Forthcoming*.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics New York.

Gagliardini, P., E. Ossola, and O. Scaillet (2016). Time-varying risk premium in large cross-sectional equity data sets. *Econometrica 84*(3), 985–1046.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Gu, S., B. Kelly, and D. Xiu (2021). Autoencoder asset pricing models. *Journal of Econometrics 222*(1), 429–450.

He, X., G. Feng, J. Wang, and C. Wu (2021). Predicting individual corporate bond returns. Technical report, City University of Hong Kong.

Hou, K., C. Xue, and L. Zhang (2015). Digesting anomalies: An investment approach. *The Review of Financial Studies 28*(3), 650–705.

Kelly, B. T., S. Pruitt, and Y. Su (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics 134*(3), 501–524.

Kim, S., R. A. Korajczyk, and A. Neuhierl (2020). Arbitrage portfolios. *Review of Financial Studies, Forthcoming*.

Kozak, S., S. Nagel, and S. Santosh (2020). Shrinking the cross-section. *Journal of Financial Economics 135*(2), 271–292.

Lettau, M. and M. Pelger (2020). Estimating latent asset-pricing factors. *Journal of Econometrics 218*(1), 1–31.

Moritz, B. and T. Zimmermann (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Technical report, Ludwig Maximilian University Munich.

Rossi, A. G. (2018). Predicting stock market returns with machine learning. Technical report, Georgetown University.

Figure 4: Panel Tree for the period from 1981 to 2000

The main tree structure trained from the period from 1981 to 2000 is displayed in this figure. We show splitting characteristics and cutpoint values for each parent nodes. The node numbers (N#) and splitting order numbers (S#) are also printed on each parent node. We have included the median monthly number of assets in the final leaves basis portfolios. The description of characteristics are listed in Table A.1.

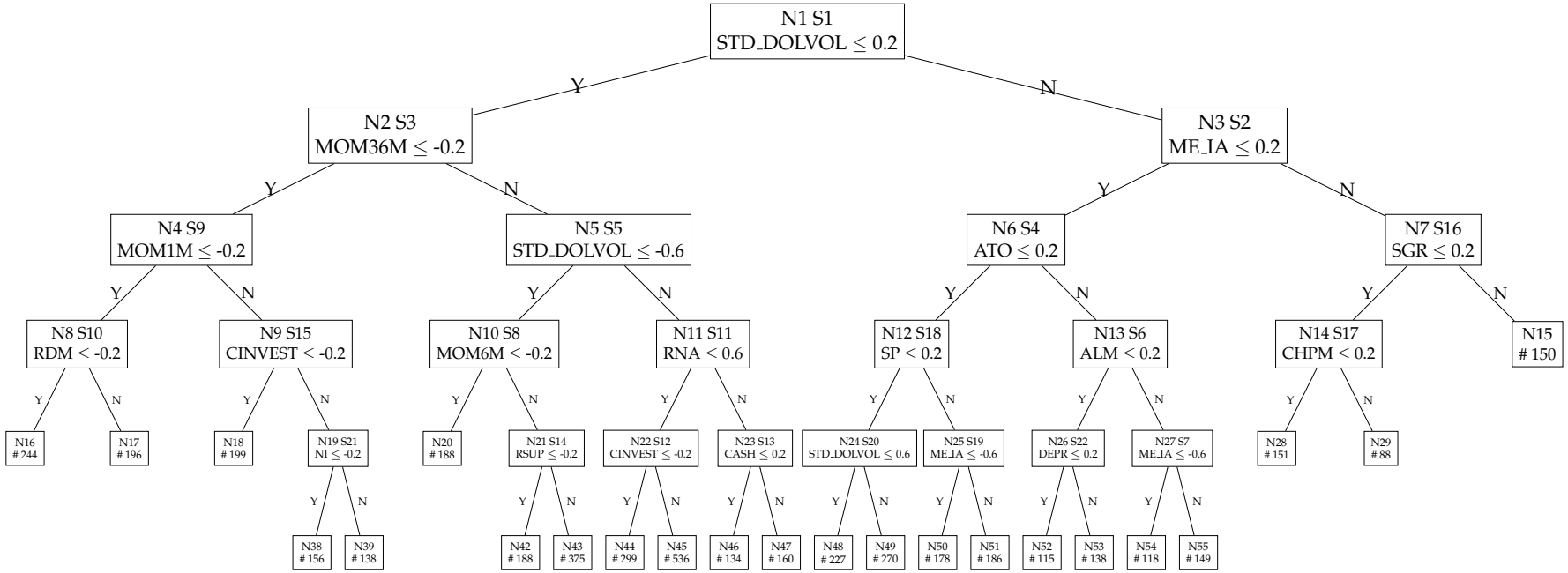


Figure 5: Partition Plots for Figure 4

This diagram visualizes the partition for the first five splits of the tree structure in Figure 4. For example, the first split (S1) is implemented with STD_DOLVOL on the entire stock universe, and the second split is implemented with ME_IA on the high STD_DOLVOL portfolios. The space area for each partition represents the corresponding proportion of the stock universe. We also provide the monthly average return and annualized Sharpe ratio for each leaf. The overlayed arrows represent the next split is implemented on the partitioned area from the previous partition.

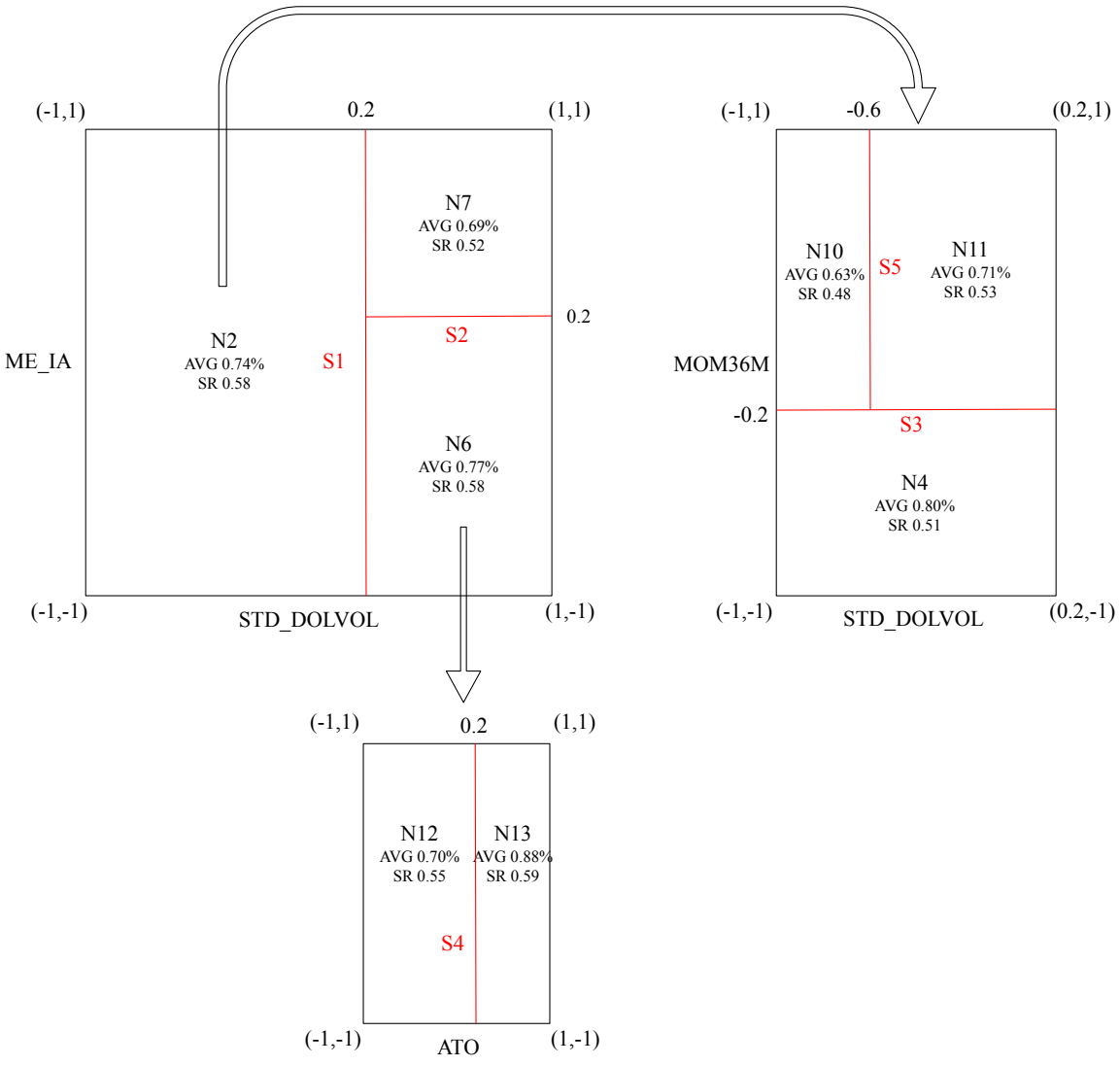


Figure 6: Return-Risk Clustering for Leaf Basis Portfolios

Following the tree structure in Figure 4, this figure plots the monthly average return and standard deviation for leaf basis portfolios. Labels N4, N5, N6, and N7 represent those four leaf basis portfolios at depth 3. Labels N4+, N5+, N6+, and N7+ represent corresponding leaf basis portfolios at depth 6. We also show our generated SDF portfolios at depth 3 and depth 6 labeled with MVE-3 and MVE-6. For the reference lines, we have included the annualized Sharpe ratio 0.3, 0.6, and 0.9.

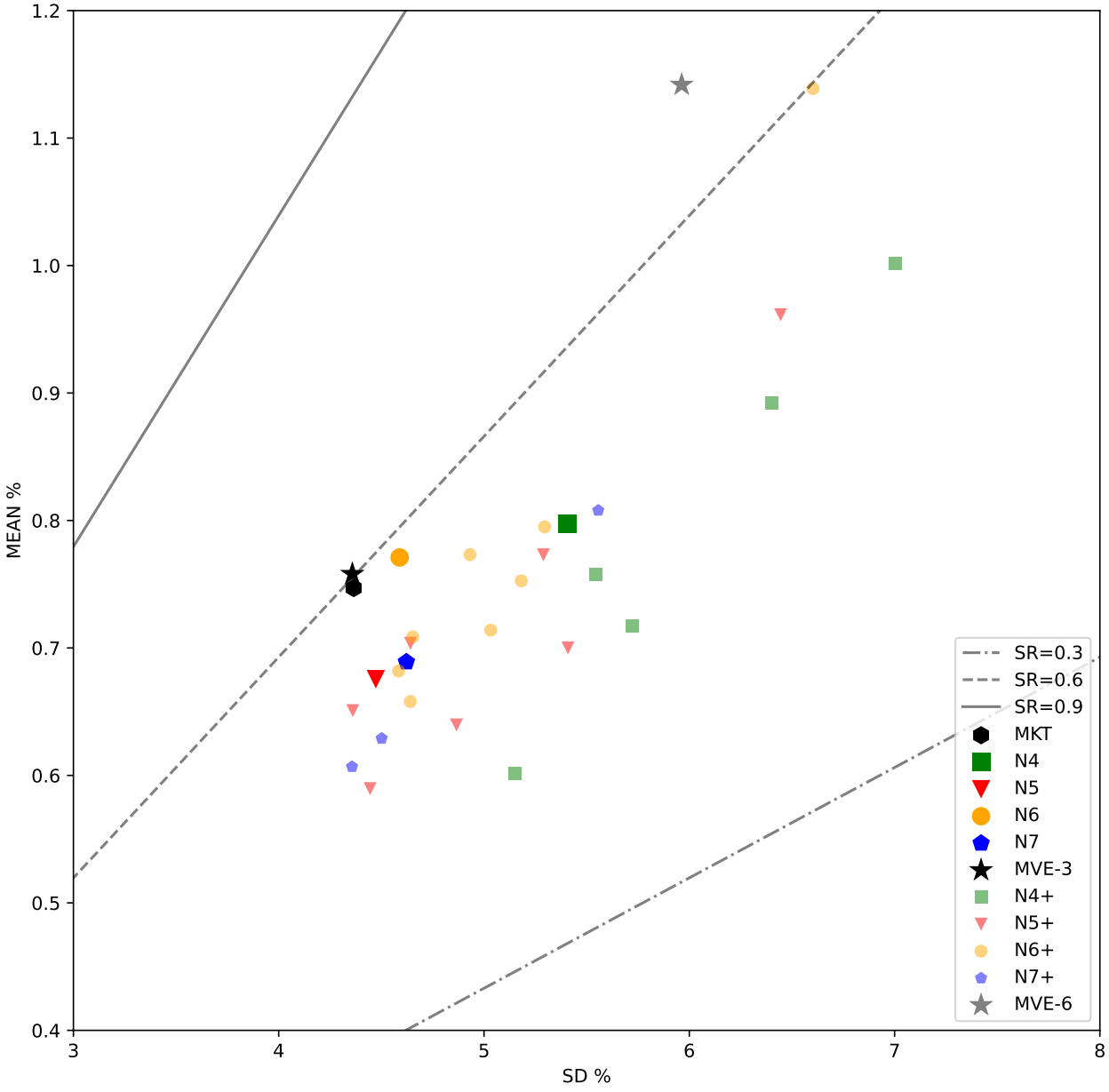


Figure 7: Out-of-Sample: Return-Risk Clustering for Leaf Basis Portfolios

This figure shows the out-of-sample performance for all those portfolios plotted in Figure 6. The figure format follows Figure 6

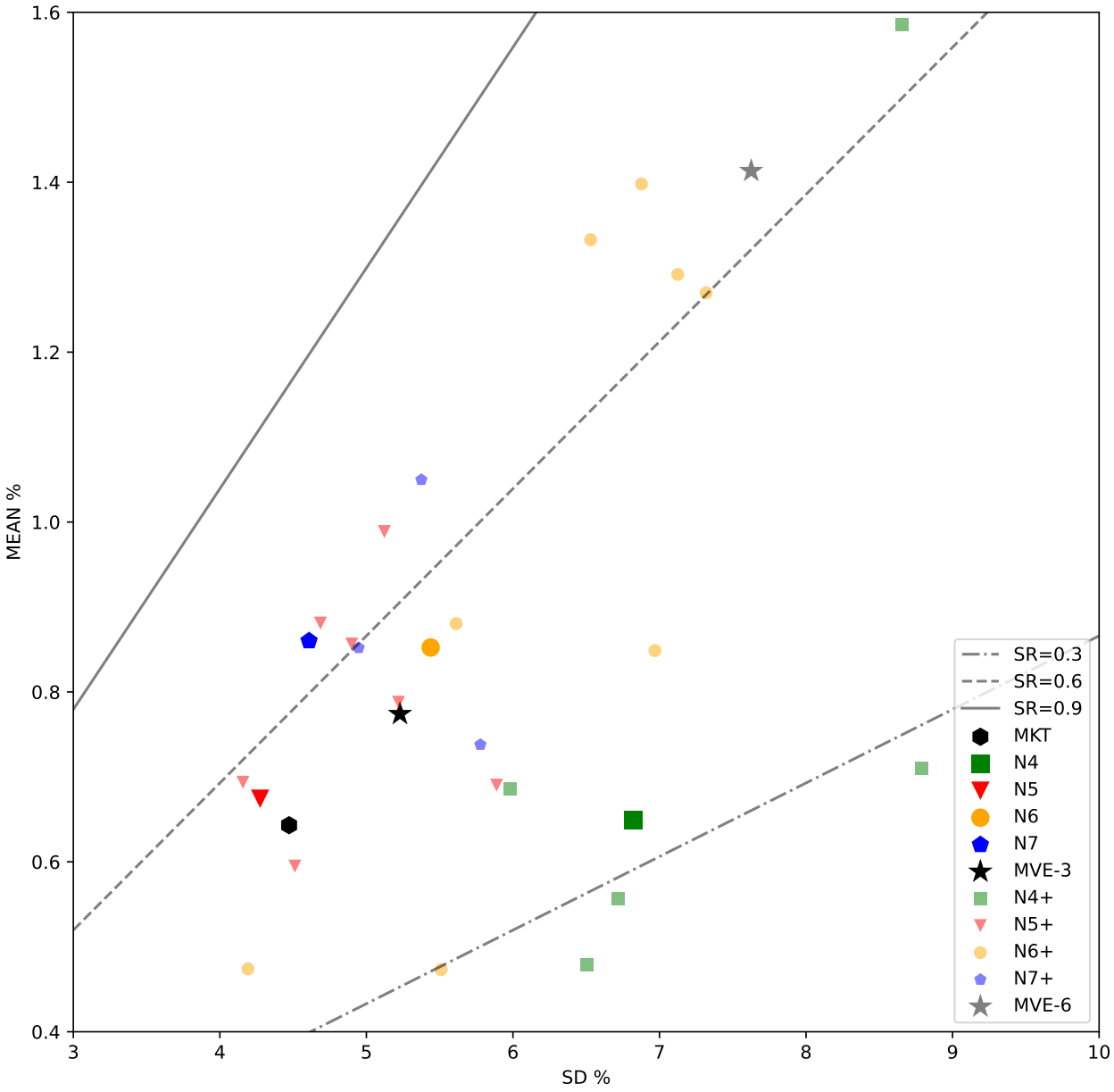
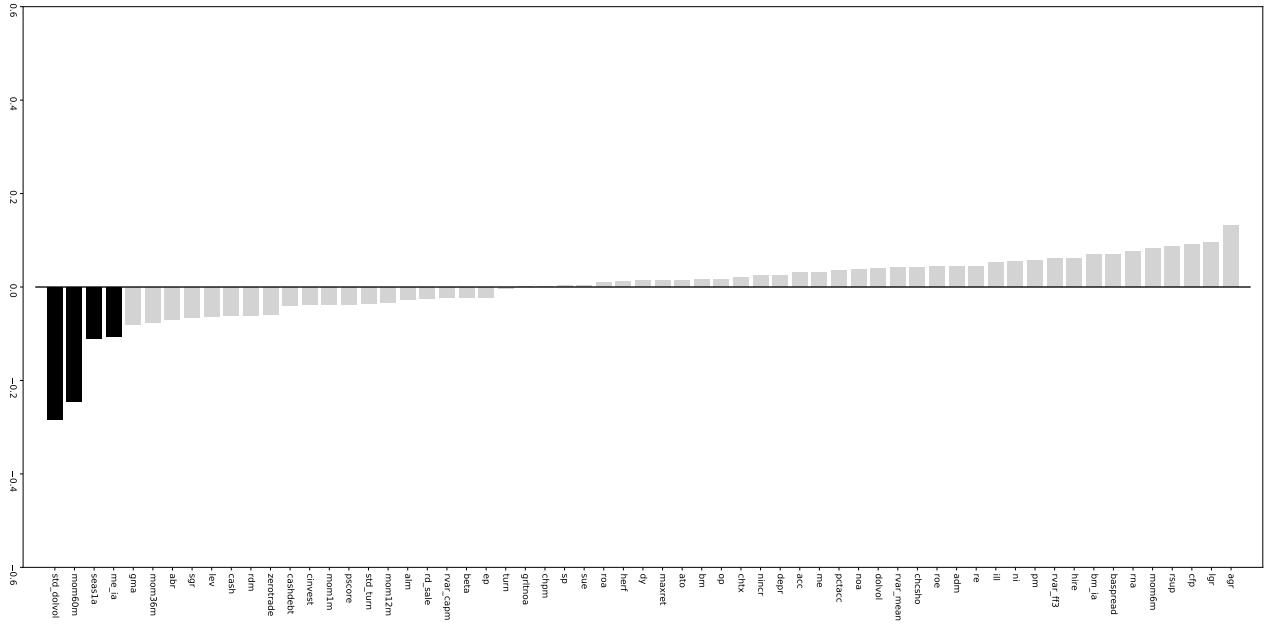
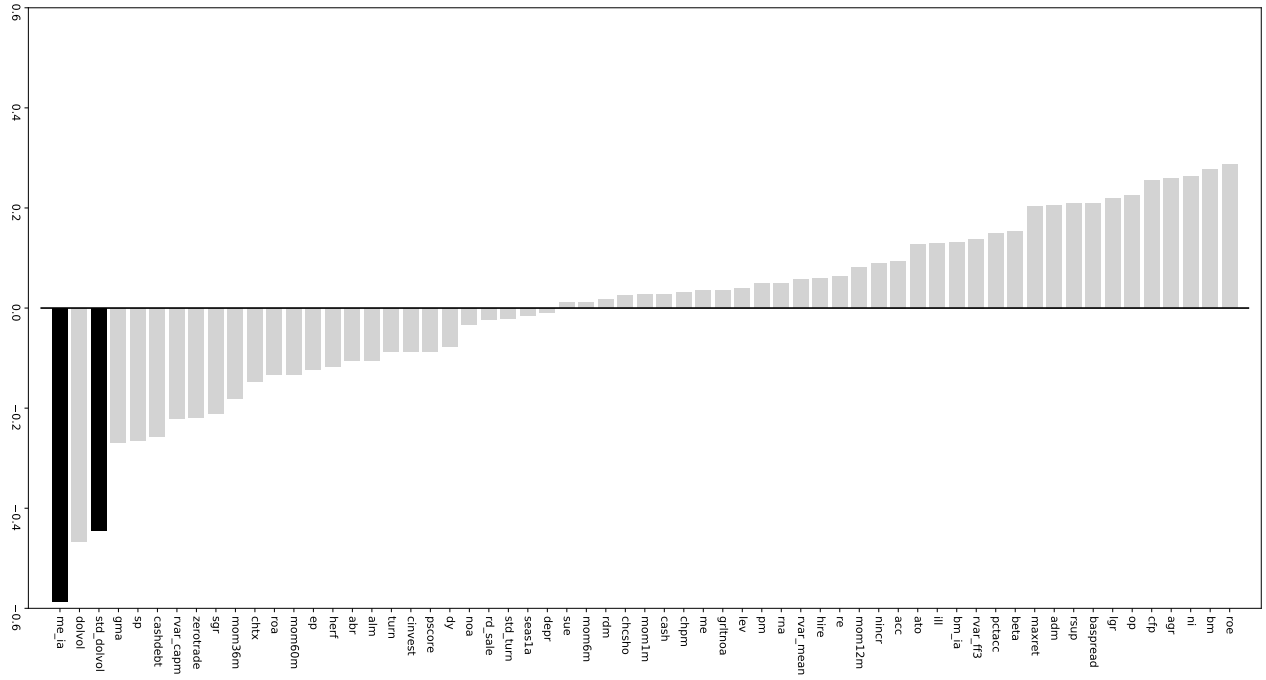


Figure 8: Out-of-Bag Characteristics Significance

This figure reports the characteristic significance by the out-of-bag ensembles from the random forest of 500 trees. The train sample period is 1981-2000, and the test sample period is 2001-2020. The variable importance measure is defined as the average increase percentage of loss function by including a characteristic in a tree model. A negative value implies that including this characteristic reduces loss and is useful. The dark color bars on the left are significant characteristics at the 5% level by the two-sample t-test. The description of characteristics are listed in Table A.1.



1981-2000



2001-2020

Figure 9: Panel Tree for the period from 1981 to 2000: High/Low Inflation

This figure shows the Panel Tree structure by considering both time-series and cross-sectional variation. The most important macro predictor is Inflation, and the first split is implemented when the current inflation level is lower than the median of the past decade. For high and low inflation periods, two different tree models are provided as two child leaves. The figure format follows Figure 4.

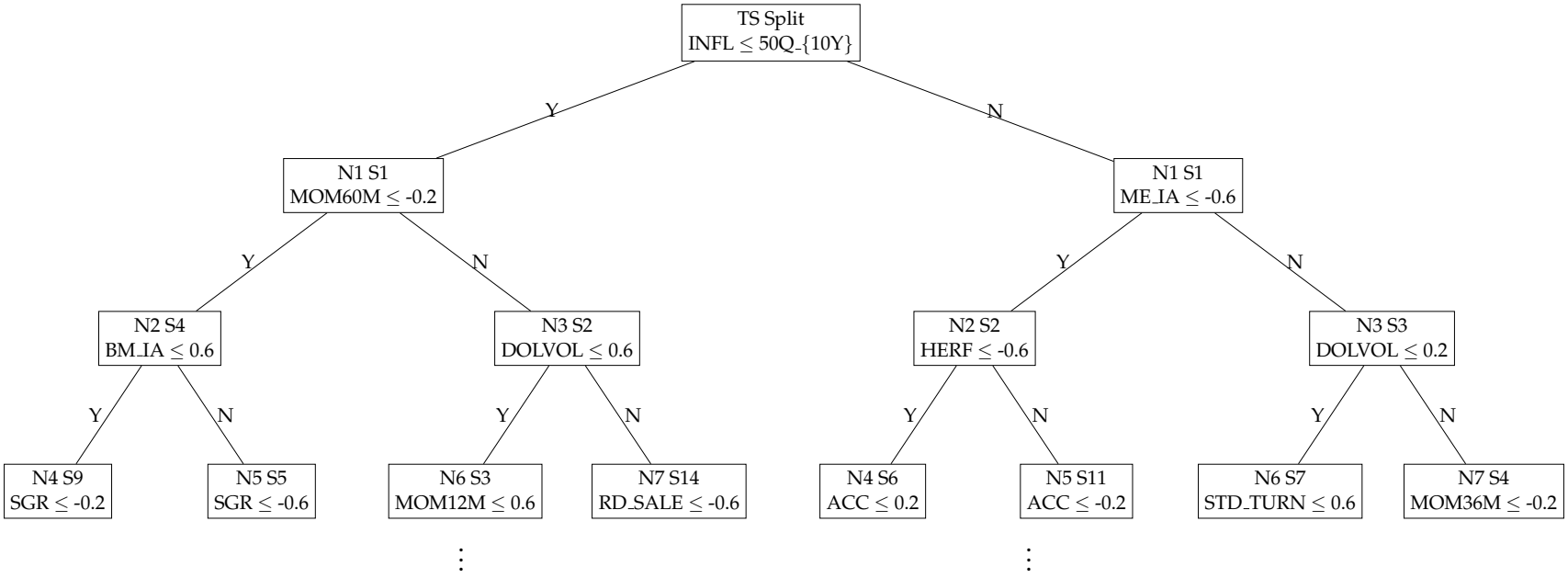


Table 1: Asset Pricing Performance

This table reports the performances of asset pricing models. The 'Tot' (total R^2 %) in Equation 18 and 'Pred' (predictive R^2 %) in Equation 19 are goodness of fitness measures for individual stock returns. The in-sample period is from 1981 to 2000, and the out-of-sample period is from 2001 to 2020. We also report the XS-R^2 % in Equation 20, using the factor models in the rows to price the test asset portfolios in the columns. Though the P-Tree factors are generated in the training sample, XS-R^2 values are estimated using the entire sample from 1981 to 2020.

	Individual Stocks				Portfolios		
	In-Sample		Out-of-Sample		Entire Sample		
	Tot	Pred	Tot	Pred	FF25	Ind49	P-Tree23
Panel A: Panel Tree Factors							
P-Tree1	8.46	-0.04	12.38	0.29	88.0	82.2	97.1
P-Tree3	10.98	0.11	14.89	0.32	50.2	-0.4	62.1
P-Tree5	11.63	0.37	15.78	0.38	68.1	15.5	77.0
Panel B: MKT Adj. Panel Tree Factors							
MKT	6.86	0.05	11.15	0.28	91.2	87.7	93.5
MKT+P-Tree2	11.22	0.25	15.09	0.17	71.8	55.7	74.0
MKT+P-Tree4	11.86	0.38	16.33	0.29	76.8	82.0	85.7
Panel C: Time Series Split - Panel Tree Factors							
TS-P-Tree1	9.16	0.00	13.22	0.28	83.6	75.3	86.1
TS-P-Tree3	11.40	0.32	15.50	0.09	56.8	12.8	70.5
TS-P-Tree5	12.09	0.57	16.21	0.09	76.1	35.2	81.1
Panel D: Time Series Split - MKT Adj. Panel Tree Factors							
TS-MKT	6.89	0.10	10.99	0.24	87.7	82.0	81.2
TS-MKT+P-Tree3	11.37	0.43	14.99	0.03	68.3	68.2	67.9
TS-MKT+P-Tree5	12.12	0.49	16.53	0.02	86.8	58.4	83.1
Panel E: Others							
CAPM	6.86	0.05	11.18	0.29	91.2	87.7	93.5
FF3	9.92	0.19	14.01	0.38	94.8	84.9	92.8
FF5	10.34	0.24	14.57	0.38	96.0	77.9	91.6
Q4	10.07	0.30	14.49	0.34	95.7	81.7	88.6
RPPCA	10.31	0.03	14.44	0.36	77.8	57.6	84.7
IPCA	12.54	1.23	16.46	0.87	-11.7	-69.1	-28.9

Table 2: Investment Performance

This table reports the investment performance of the factor models. The table format is similar to Table 1. We report the average returns, Sharpe ratio, and monthly Jensen's alpha % (t-stat) of MVE and 1/N portfolios.

	In-Sample (1981-2000)						Out-of-Sample (2001-2020)					
	MVE			1/N			MVE			1/N		
	AVG	SR	α	AVG	SR	α	AVG	SR	α	AVG	SR	α
Panel A: Panel Tree Factors												
P-Tree1	1.14	0.66	0.34	1.14	0.66	0.34	1.41	0.64	0.48	1.41	0.64	0.48
P-Tree3	2.08	1.52	1.53***	1.94	1.49	1.34***	1.65	1.07	1.01***	1.60	1.02	0.92***
P-Tree5	4.44	3.24	4.22***	2.42	1.98	1.84***	2.18	1.12	2.36***	1.44	1.25	1.00***
Panel B: MKT Adj. Panel Tree Factors												
MKT	0.73	0.57	0.00	0.73	0.57	0.00	0.64	0.49	0.00	0.64	0.49	0.00
MKT+P-Tree2	3.90	2.43	3.63***	2.25	1.70	1.66***	2.26	1.39	1.87***	1.14	0.98	0.69***
MKT+P-Tree4	4.89	3.62	4.66***	2.20	1.72	1.56***	2.61	1.78	2.52***	1.05	1.03	0.67***
Panel C: Time Series Split - Panel Tree Factors												
TS-P-Tree1	1.11	0.71	0.36*	1.11	0.71	0.36*	1.20	0.63	0.37*	1.20	0.63	0.37*
TS-P-Tree3	4.12	1.83	3.41***	3.76	1.72	2.99***	3.33	1.50	3.14***	2.75	1.28	2.55***
TS-P-Tree5	7.60	3.59	7.22***	4.18	2.43	3.53***	4.25	2.02	4.15***	2.78	1.78	2.50***
Panel D: Time Series Split - MKT Adj. Panel Tree Factors												
TS-MKT	0.73	0.57	0.00	0.73	0.57	0.00	0.64	0.49	0.00	0.64	0.49	0.00
TS-MKT+P-Tree2	2.77	1.63	2.32***	2.69	1.33	2.09***	1.62	1.21	1.28***	1.30	0.90	0.98***
TS-MKT+P-Tree4	2.96	1.97	2.47***	2.67	1.38	1.98***	1.58	1.24	1.35***	1.15	0.77	0.78***
Panel E: Others												
FF3	0.55	1.17	0.41***	0.38	0.85	0.20***	0.21	0.29	-0.06	0.28	0.40	0.00
FF5	0.46	1.47	0.39***	0.38	1.34	0.33***	0.26	0.62	0.12*	0.25	0.59	0.12
Q4	0.58	1.95	0.52***	0.52	1.40	0.36***	0.22	0.69	0.18***	0.29	0.77	0.16***
RP-PCA	2.09	1.64	1.72***	0.48	0.21	-0.63	0.93	0.47	0.21	0.91	0.51	0.16
IPCA	2.84	4.94	2.80***	1.85	1.96	1.40***	1.78	2.81	1.67***	1.40	1.22	0.91***

Table 3: Factor Spanning Alpha Test

This table reports the monthly alphas in basis point and their significance for the factor spanning test. We regress each of the factors in the rows against a factor model in the columns. For the P-Tree factor, we name the factors by the first two splitting characteristics in the tree structure, which can also be viewed as interaction factors. We also show the mean-variance efficient (MVE) and 1/N portfolios for five factors and different test assets. For t-statistics *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

	In-Sample (1981-2000)				Out-of-Sample (2001-2020)			
	CAPM	FF3	FF5	Q4	CAPM	FF3	FF5	Q4
Panel A: Panel Tree Factors								
STD_DOLVOL-ME_IA	34	45***	54***	65***	48*	39*	61***	65***
OP-ME	245***	170***	134***	61	116***	137***	69**	55
BM_IA-DOLVOL	123***	44*	34	61**	112***	116***	115***	124***
MOM12M-ME	488***	470***	450***	392***	234***	261***	177***	149***
ME-STD_DOLVOL	28	36*	33*	9	-8	-7	-10	-21
MVE (5-factor)	422***	370***	349***	317***	236***	257***	192***	179***
1/N (5-factor)	184***	153***	141***	118***	100***	109***	83***	74***
Panel B: MKT Adj. Panel Tree Factors								
MKT	-	-	-	-	-	-	-	-
RVAR_FF3-STD_TURN	397***	347***	316***	243***	111*	146***	52	50
BM_IA-ME	102***	57***	51***	71***	95***	88***	106***	112***
MOM12M-ME_IA	230***	276***	270***	200***	149***	154***	63	58
ME-CASH	52**	102***	118***	130***	-20	-21	21	17
MVE (5-factor)	466***	427***	408***	375***	252***	260***	215***	221***
1/N (5-factor)	156***	156***	151***	129***	67***	73***	48***	47***
Panel C: Other Test Assets								
MVE-FF25	340***	281***	270***	253***	111***	129***	93***	68*
MVE-IND49	332***	329***	335***	323***	40	47	63	77
MVE-P-Tree23	34	45***	54***	65***	48*	39*	61***	65***

Table 4: Characteristics Importance by Top Splits

This table reports the most frequently selected characteristics from the random forest of 500 trees. The "Top 1" rows only count the first split for 500 trees. The "Top 2" or "Top 3" rows only count the first two or three splits. The numbers reported are the selection frequency for these top characteristics selected out of the 500 ensembles. Panel A uses the entire train sample, and the other panels report the mostly characteristics chosen for high and low inflation periods from 1981 to 2000. The description of characteristics are listed in Table A.1.

Panel A: Entire Training Sample (1981-2000)

	1	2	3	4	5
Top1	std_dolvol	mom60m	dolvol	me_ia	lgr
	0.73	0.64	0.61	0.33	0.29
Top2	std_dolvol	dolvol	mom60m	me_ia	beta
	0.78	0.73	0.72	0.61	0.44
Top3	std_dolvol	mom60m	dolvol	me_ia	ato
	0.78	0.76	0.73	0.64	0.52

Panel B: High Inflation

	1	2	3
Top 1	mom60m	dolvol	me_ia
	0.53	0.44	0.42
Top 2	mom60m	dolvol	me_ia
	0.59	0.56	0.53
Top 3	mom60m	dolvol	me_ia
	0.60	0.56	0.53

Panel C: Low Inflation

	1	2	3
Top 1	me_ia	dolvol	me
	0.56	0.42	0.41
Top 2	me_ia	me	dolvol
	0.68	0.55	0.49
Top 3	me_ia	mom60m	me
	0.63	0.50	0.47

Table 5: Uni-Sort Factors v.s. Interaction Factors

This table summarizes the significant counts for long-short factors for average returns and Jensen's alphas. We count the number of significant average returns and alphas at the 5 % and 10% level in panels A and B. We report the cross-sectional quantile values of average returns and alphas in panel C among the 61 characteristics. The four specifications are (1) 4x1 long-short portfolio, (2) 2x1 long-short portfolio, (3) interaction factors, and (4) market-adjusted interaction factors. Our panel tree creates specifications (3) and (4), which follow the same models in Table 1.

Panel A: # of Significant Cases with 5% level

	Uni-Sort 4x1		Uni-Sort 2x1		Interaction		Mkt-Adj Interaction	
	# Mean	# Alpha	# Mean	# Alpha	# Mean	# Alpha	# Mean	# Alpha
81-00	17	30	6	22	11	27	10	30
01-20	4	22	5	11	24	28	11	21
81-20	22	33	12	28	32	31	16	42

Panel B: # of Significant Cases with 10% level

	Uni-Sort 4x1		Uni-Sort 2x1		Interaction		Mkt-Adj Interaction	
	# Mean	# Alpha	# Mean	# Alpha	# Mean	# Alpha	# Mean	# Alpha
81-00	26	34	10	31	18	30	19	39
01-20	8	27	7	18	29	30	14	33
81-20	27	36	18	34	36	41	21	48

Panel C: Cross-Sectional Quantiles for Average and Alpha

		Uni-Sort 4x1		Uni-Sort 2x1		Interaction		Mkt-Adj Interaction	
	q	Avg	Alpha	Avg	Alpha	Avg	Alpha	Avg	Alpha
81-00	25	0.12	0.06	0.07	0.06	0.09	0.09	0.1	0.22
	50	0.33	0.46	0.18	0.21	0.22	0.28	0.19	0.29
	75	0.61	0.69	0.23	0.31	0.33	0.43	0.27	0.41
01-20	25	0.02	0.07	-0.02	0.05	0.06	0.08	-0.02	0.13
	50	0.18	0.29	0.06	0.13	0.41	0.36	0.09	0.25
	75	0.36	0.58	0.2	0.31	0.56	0.65	0.25	0.41
81-20	25	0.13	0.14	0.04	0.08	0.08	0.12	0.05	0.17
	50	0.28	0.34	0.11	0.16	0.30	0.29	0.14	0.28
	75	0.41	0.61	0.19	0.26	0.43	0.48	0.25	0.39

Table 6: Examples for Interaction Factors

This table follows the 6 examples in Figure A.3. We report the monthly average returns (%) and Jensen's alpha (%) of the 4x1 long-short factors and the interaction factors. The interaction factors are created with the train sample period from 1981 to 2000, and we have provided the corresponding interaction characteristics. For t-statistics *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively. The description of characteristics are listed in Table A.1.

Panel A: Stand. Unexp. Earnings

	Uni-Sort 4x1		+RDM+ME	
	Mean	Alpha	Mean	Alpha
81-00	0.68***	0.63***	0.95***	0.91***
01-20	0.53***	0.67***	1.43***	1.36***
81-20	0.61***	0.66***	1.19***	1.13***

Panel B: Profit Margin

	Uni-Sort 4x1		+RDS+DOLVOL	
	Mean	Alpha	Mean	Alpha
81-00	0.10	0.06	0.39***	0.46***
01-20	0.18	0.28*	0.55***	0.56***
81-20	0.14	0.17	0.47***	0.51***

Panel C: Cash Holdings

	Uni-Sort 4x1		+NOA+BAS	
	Mean	Alpha	Mean	Alpha
81-00	0.39	0.05	1.12***	1.10***
01-20	0.43*	0.21	-0.04	0.02
81-20	0.41**	0.14	0.54***	0.56***

Panel D: Dollar Trading Volume

	Uni-Sort 4x1		+RVAR_FF3+ME	
	Mean	Alpha	Mean	Alpha
81-00	-0.20	0.02	0.11	0.33**
01-20	0.33*	0.42**	0.34*	0.45***
81-20	0.06	0.22	0.23*	0.39***

Panel E: Seasonality

	Uni-Sort 4x1		+RDM+DOLVOL	
	Mean	Alpha	Mean	Alpha
81-00	0.54**	0.37*	0.13	0.21
01-20	0.01	0.05	0.47**	0.46**
81-20	0.27	0.22	0.30*	0.33**

Panel F: R&D to Sales

	Uni-Sort 4x1		+RVAR_FF3+EP	
	Mean	Alpha	Mean	Alpha
81-00	0.43	0.08	1.11***	1.29***
01-20	-0.06	-0.28	0.04	0.49*
81-20	0.19	-0.10	0.58***	0.90***

Appendices

Table A.1: Equity Characteristics (61 in total

No.	Characteristics	Description
1	abr	Abnormal returns around earnings announcement
2	acc	Operating Accruals
3	adm	Advertising Expense-to-market
4	agr	Asset growth
5	alm	Quarterly Asset Liquidity
6	ato	Asset Turnover
7	baspread	Bid-ask spread (3 months)
8	beta	Beta (3 months)
9	bm	Book-to-market equity
10	bm_ia	Industry-adjusted book to market
11	cash	Cash holdings
12	cashdebt	Cash to debt
13	cfp	Cashflow-to-price
14	chcsho	Change in shares outstanding
15	chpm	Industry-adjusted change in profit margin
16	chtx	Change in tax expense
17	cinvest	Corporate investment
18	depr	Depreciation / PP&E
19	dolvol	Dollar trading volume
20	dy	Dividend yield
21	ep	Earnings-to-price
22	gma	Gross profitability
23	grltnoa	Growth in long-term net operating assets
24	herf	Industry sales concentration
25	hire	Employee growth rate
26	ill	Illiquidity rolling 3m
27	lev	Leverage
28	lgr	Growth in long-term debt
29	maxret	Maximum daily returns rolling 3m
30	me	Market equity
31	me_ia	Industry-adjusted size
32	mom12m	Cumulative Returns in the past (2-12) months
33	mom1m	Previous month return
34	mom36m	Cumulative Returns in the past (13-35) months
35	mom60m	Cumulative Returns in the past (13-60) months
36	mom6m	Cumulative Returns in the past (2-6) months
37	ni	Net Equity Issue

Continue: Equity Characteristics (61 in total)

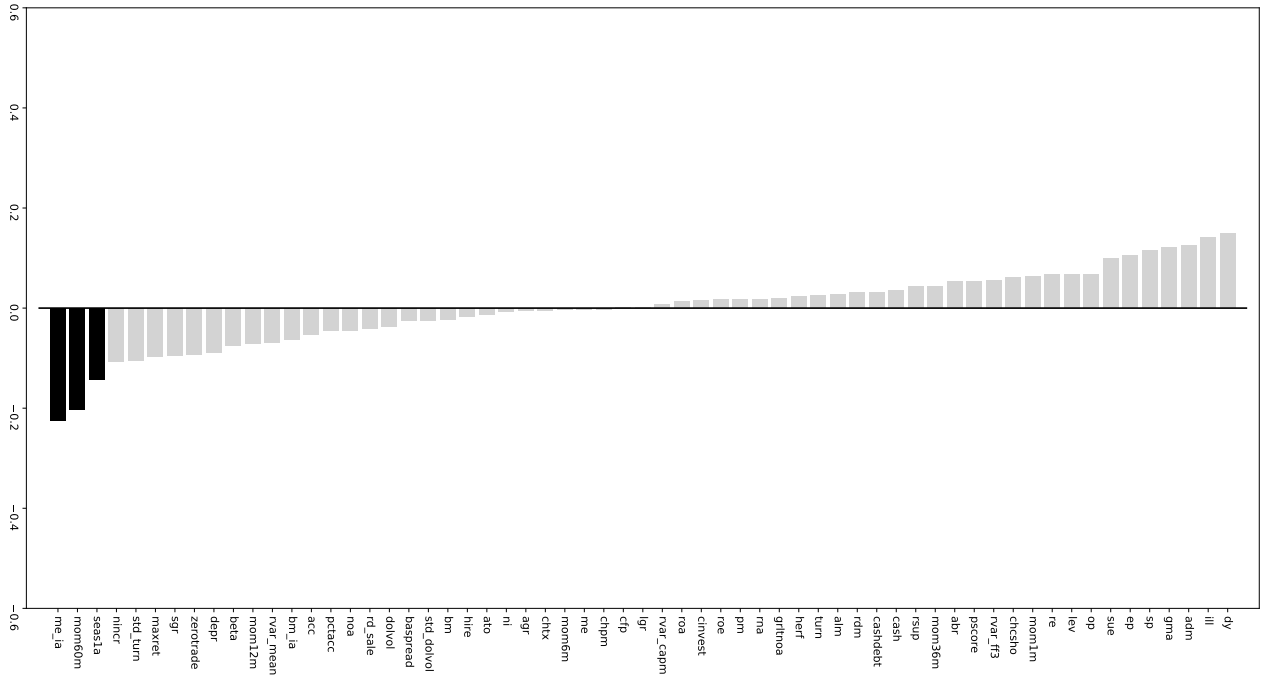
No.	Characteristics	Description
38	nincr	Number of earnings increases
39	noa	Net Operating Assets
40	op	Operating profitability
41	pctacc	Percent operating accruals
42	pm	profit margin
43	ps	Performance Score
44	rd_sale	R&D to sales
45	rdm	R&D Expense-to-market
46	re	Revisions in analysts' earnings forecasts
47	rna	Return on Net Operating Assets
48	roa	Return on Assets
49	roe	Return on Equity
50	rsup	Revenue surprise
51	rvar_capm	Residual variance - CAPM (3 months)
52	rvar_ff3	Residual variance - ff3 (3 months)
53	rvar_mean	Return variance (3 months)
54	seas1a	1-Year Seasonality
55	sgr	Sales growth
56	sp	Sales-to-price
57	std_dolvol	Std of dollar trading volume (3 months)
58	std_turn	Std. of Share turnover (3 months)
59	sue	Unexpected quarterly earnings
60	turn	Shares turnover
61	zerotrade	Number of zero-trading days (3 months)

Table A.2: Macro Predictors for Market Timing

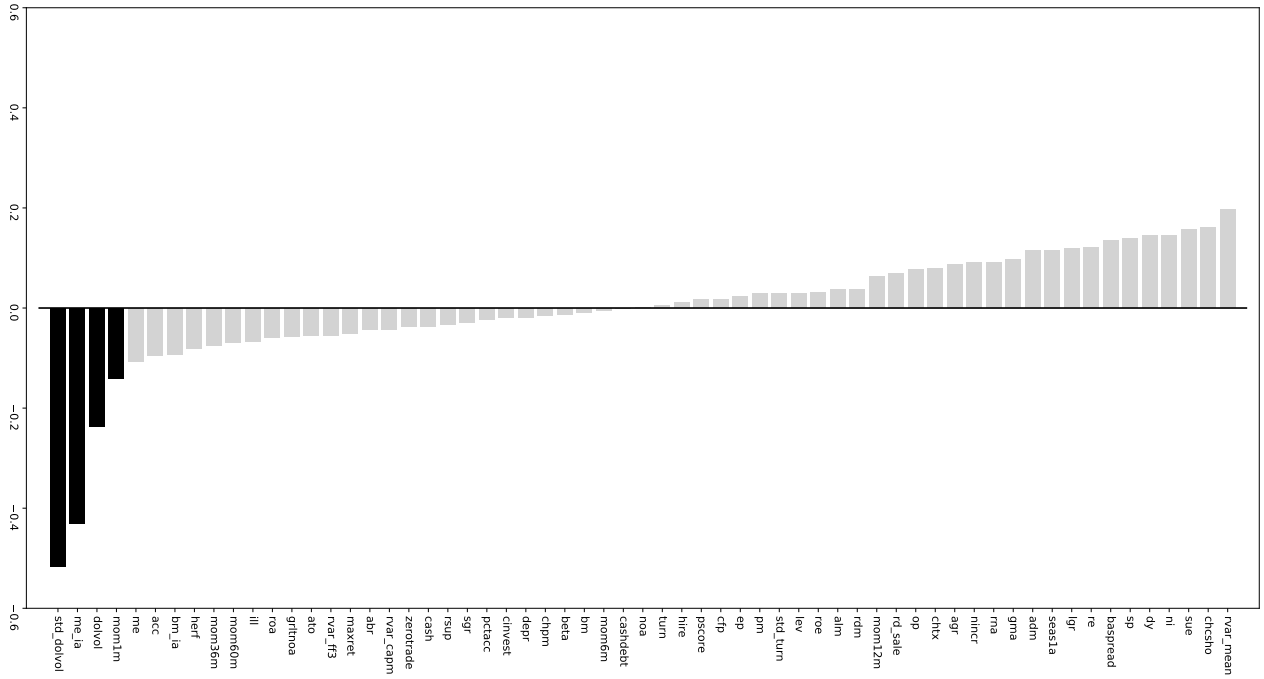
No.	Variable Name	Description
1	ep	Earnings-to-price of S&P 500
2	dy	Dividend yield of S&P 500
3	lev	Leverage of S&P 500
4	ni	Net equity issuance of S&P 500
5	svar	Stock Variance of S&P 500
6	ill	Pastor-Stambaugh illiquidity
7	infl	Inflation
8	tbl	Three-month treasure bill rate
9	dfy	Default yield
10	tms	Term spread

Figure A.1: Out-of-Bag Characteristics Significance: High/Low Inflation

This figure reports the characteristic significance by the out-of-bag ensembles from the random forest of 500 trees. We report results for high and low inflation periods in the train sample 1981-2000. This figure details follow Figure 8. The left dark columns indicate significant characteristics.



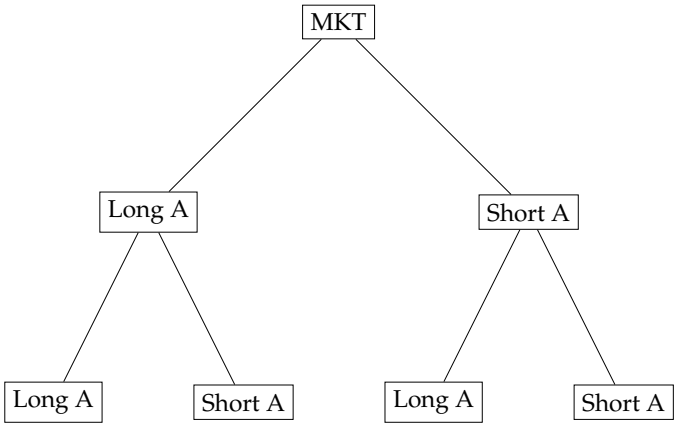
High INFL



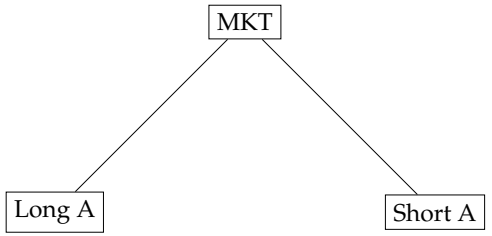
Low INFL

Figure A.2: Four Specifications of Characteristic-sorted Factors

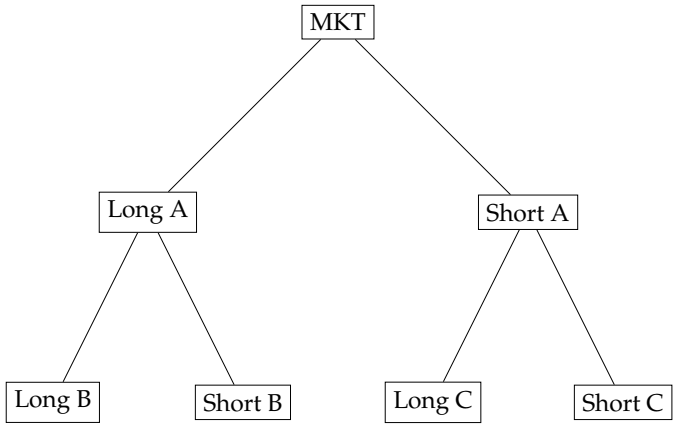
We show four different specifications for creating long-short factors by the tree model. Specification (a) is the classic sorting with a single characteristic and creates four 4x1 sorted portfolios. Specification (b) is similar and creates two 2x1 sorted portfolios. Specification (c) is trained by the Panel Tree model with interactions among characteristics. To each characteristic, we fit the Panel tree model to search optimal characteristics for its interaction by fixing the first split. Specification (d) is similar to (c) but fits with market-adjusted returns.



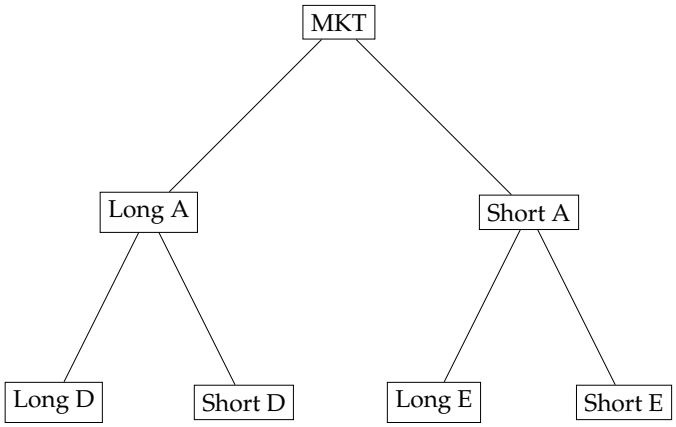
(a) Uni-Sort 4x1



(b) Uni-Sort 2x1



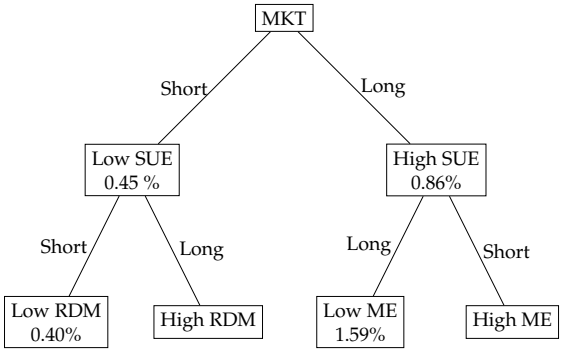
(c) Interaction



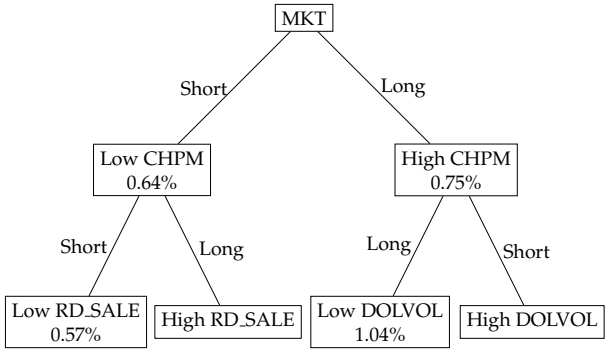
(d) Market Adjusted Interaction

Figure A.3: Examples for Interaction Factors

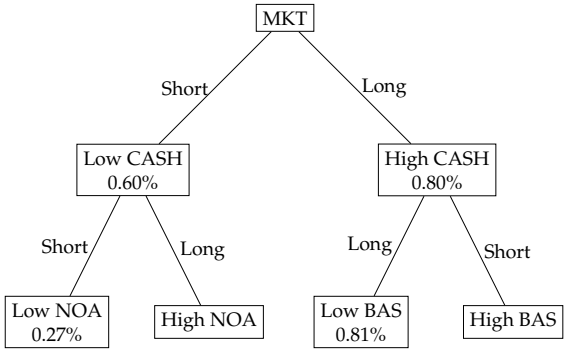
This figure shows six examples of the interaction factors. More details are reported in Table 6. In the second layer, the numbers report the average returns of the long portfolio and short portfolio. In the third layer, the numbers are the average returns of the long-long portfolio and short-short portfolio. One can see further splitting helps to create a higher return spread.



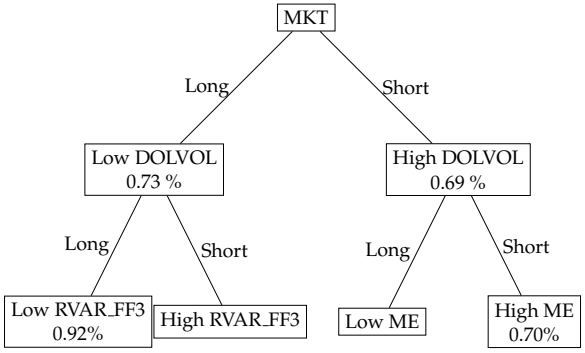
(a) Standardized Unexpected Earning



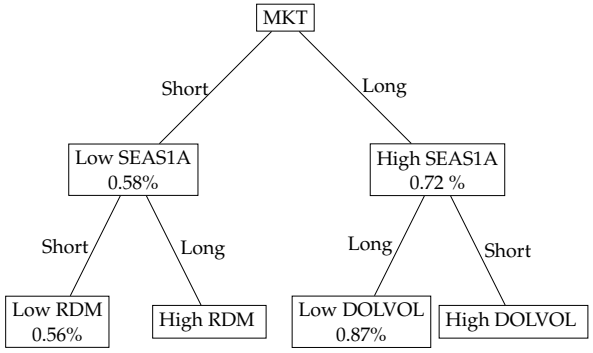
(b) Profit Margin



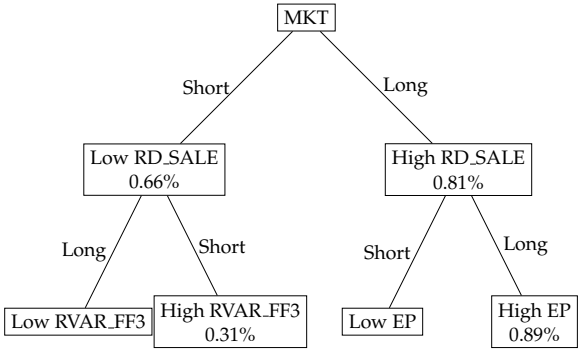
(c) Cash Holdings



(d) Dollar Trading Volume



(e) Seasonality



(f) R&D-to-Sales