0. Introduction

In the 06/28/2021 Managed Account Rating System (MARS) validation report, Model Risk Oversight (MRO) issued a plan of action due 4/30/2022 that included the following language:

> MRO requests the model owner's team conduct thorough analyses related to model development and revisit its modeling choices or provide evidence that model enhancements are not possible; this should include choices going into [redacted] regressions, the number of [redacted] used in the final modeling step, the use of new input factors, revisitation of the value of using all the currently utilized factors, and an overview of the model's overall structure.

To these broad ends, we first conduct a battery of tests to set a quantitative baseline for current MARS independent variables (components). Next, we investigate entirely new components and modified components (including the various calculations of alpha) using the same tests. Finally, we evaluate the current weight selection process with existing and new component sets and consider methods that may enhance this process.

In both univariate and multivariate tests, we find existing MARS components are not redundant, they each contain useful information, and there does not exist evidence to drop them from the current model. Although we do find the [redacted] component to be a consistent laggard. Through those same tests, there is evidence to include new and modified components, including a different computation of alpha and its factor selections. We show that tidy linear regression outputs generated during the alpha component construction stage do not imply [redacted] component predictive properties. Using a test set of new components, we find successful weight deployments are heavily dependent on existing quantitative research. Using this research, we show new and modified components with predictive properties do generate compelling model results, but their inclusion involves tradeoffs, such as performance volatility. There exists evidence that modifications to the nonlinear solvers that generate model weights could improve performance metrics. These modifications also reduce performance volatility. Lastly, we make the case that all modeling enhancements must fit the model's business needs and scale across other models, and that the evidence that supports these enhancements must be shown not to be spurious.

1. Existing Components

We evaluate MARS existing components—[redacted]—to establish their predictive power and inclusion in the model. Theory and logic and mathematical formulations underpinning these components are detailed exhaustively in the MARS whitepaper.

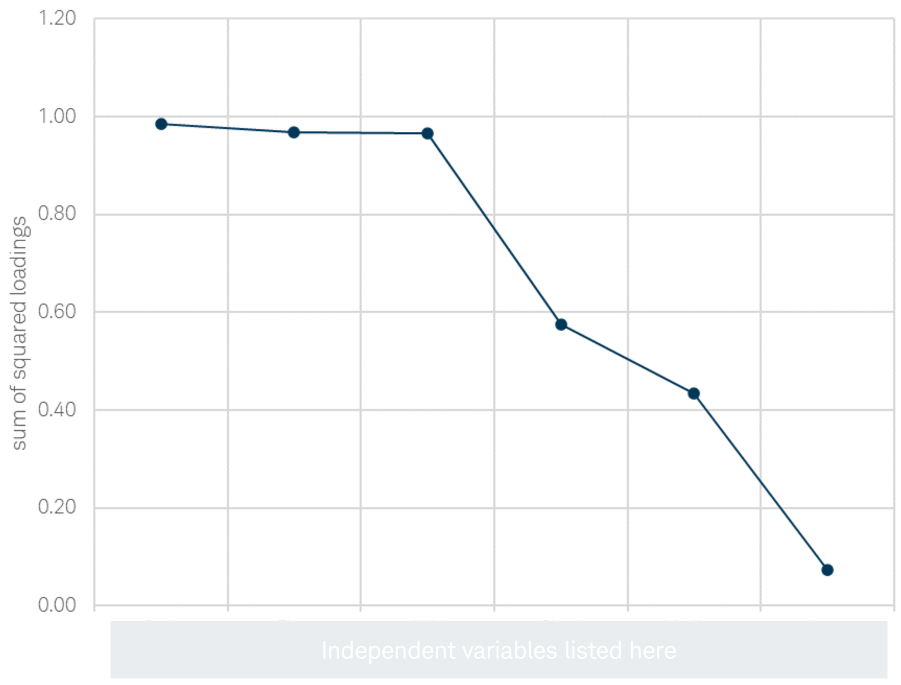
Principal component analysis (PCA). Here we aim to prime our intuition on the structure of the components currently used in the MARS model. Using PCA, we can do things like assess the dispersion of influence within the broader dataset by making elbow plots of sorted variance and observe kinks that may imply good partitions to drop noise. We can locate the existence and location of possible redundant information because we know redundancies will load heavily on the first few principal components. And we can get a broad sense of component importance by evaluating their loadings and thus influence on broader dataset. In general, we can get early sense of dominant components and the possible exclusion of weaker components. Please see appendix for addition details on PCA.

We consider the universe-level dataset of n ~300,000 rows where m-columns are the percentile ranks of components across all separately managed accounts (SMA) and periods. We compute the m x m covariance matrix then change its basis to m x m orthogonal eigenspace. In this space we have eigenvectors (coordinates) and eigenvalues (span). In application, the m x m eigenvector matrix columns are the principal components and rows are the component loadings (generally called "factor loadings" but we avoid that terminology here for clarity). As shown in Table 1 below, as defined by the corresponding 1 x m eigenvalue vector, we systematically clip the PCs to retain ~80% cumulative variance or "influence." This drops PC5 and PC6. We then ignore sign and consider total loadings on the remaining rows to generate the elbow plot in Figure 1 below.

Table 1: PC loadings and explained influence
i.e., eigenvectors and eigenvalues

	PC1	PC2	PC3	PC4	PC5	PC6
	-0.1519	0.3296	0.6652	0.0250	-0.6489	-0.0639
	-0.2776	0.1978	0.5610	-0.0478	0.7475	-0.0894
	-0.0695	0.0280	0.0788	-0.2483	0.0069	0.9625
Independent variables listed here	-0.0373	0.6697	-0.2737	0.6633	0.0771	0.1708
	0.0864	-0.6038	0.3241	0.7007	0.0147	0.1779
	0.9414	0.1955	0.2379	-0.0664	0.1180	0.0248
	0.1721	0.0945	0.0938	0.0718	0.0650	0.0508
	31%	49%	66%	79%	91%	100%

Figure 1: Sum of squared row loadings of retained PC



Observations include

1. Table 1 shows [redacted] has an outsized loading on PC1. This is highly material (for this exercise) and draws our attention towards the influence of this component.

2. When we sum, Figure 1 shows three tiers of influence, the most dominant of which does not include the [redacted]. Note this exercise does not include the dependent variable, so there is nothing to be observed regarding predictability. However, the structure of the data appears to be characterized by material influence [redacted] components.
3. [redacted] loadings suggest varied and material influence and a lack of redundancy. [redacted] low loadings across retained PC draws our attention to the possibility of low influence.
4. Loadings on first and second PC account for nearly half the influence. And the range of magnitude in the eigenvalues themselves suggests the universe-level data contains a signal and is worth further evaluation.

Cluster analysis is an attempt to further understand the structure of the data. Here we add the dependent variable to the test (three year ranked HPR) and ask, “what characteristics gather together?” We haven’t told the algorithm the “answer” but we will interpret results through that lens. A strong relationship might exist if high or low percentile ranks of a component populate a cluster. A weak or nonexistent relationship could exist if only ~0.50 values are consistent across a cluster. The purpose is very similar to interquartile analysis, only we use different partition criteria and are trying to be less deterministic with the dependent variable. Please see appendix for addition details on cluster analysis.

Below we deploy the k-means algorithm which uses a Euclidian distance measure. In general, we pick k clusters, assign each observation to the nearest cluster center or “centroid,” compute a new centroid, then iterate until centroid values converge. We compute inertia, which is the sum of the squared distances from each observation to its assigned centroid. We do this for k = 1:10 and evaluate another elbow plot shown in Figure 2 below. Inspection does not indicate obvious kinks. Inertia appears to flatten around three or four clusters. Further inspection tells us k=3 is where the first derivative will have its largest delta, and this could be the optimal k. However, when we evaluate the dependent variable at k=3 we only see middling values. At 10 clusters the dependent variable begins to partition, and we show these values in Table 2.

Figure 2: Cluster inertia for k=1:10 clusters

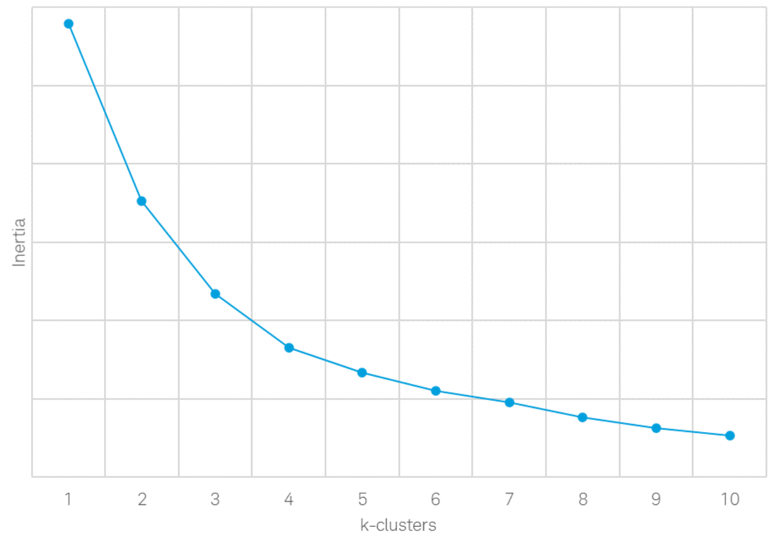


Table 2: Average intra-cluster value and count

	Independent variables listed here								
cluster8	0.49	0.79	0.47	0.46	0.68	0.04	0.23	35,655	15%
cluster4	0.23	0.24	0.46	0.30	0.53	0.06	0.24	18,063	7%
cluster9	0.38	0.45	0.54	0.80	0.23	0.06	0.31	22,701	9%
cluster2	0.78	0.24	0.47	0.49	0.66	0.06	0.42	19,163	8%
cluster5	0.23	0.30	0.44	0.47	0.55	0.91	0.47	36,896	15%
cluster6	0.59	0.62	0.59	0.23	0.21	0.05	0.51	22,795	9%
cluster1	0.70	0.52	0.50	0.54	0.50	0.89	0.59	24,121	10%
cluster3	0.67	0.65	0.46	0.81	0.27	0.04	0.68	22,294	9%
cluster10	0.75	0.77	0.55	0.47	0.74	0.04	0.78	23,930	10%
cluster7	0.24	0.41	0.48	0.52	0.66	0.06	0.78	18,867	8%

Observations include

1. We selected k=10 to observe extreme values of the dependent variables and we see them in the first three rows and the last three rows. Note the table is sorted on return rank. We can see there are mixed components results for extreme dependent variable values. For example, clusters 3 and 10 show high [redacted] but cluster 7 does not, yet all three are associated with similar return rank values.
2. [redacted] appears to be the only component with consistent middling values. There are not enough extreme values in any cluster to compute an extreme mean. This draws our attention to the possibility [redacted] has a relatively weaker relationship to the dependent variable.
3. These mixed results are consistent with common practitioner knowledge that high and low return ranks can be associated with mixed component information, especially [redacted]. We know that component predictive strength can change over time, and this appears to be reflected in the large universe-level dataset.

Bootstrap information coefficient (IC) helps us understand the stability of the samples that compute IC. IC is exhaustively detailed in both the Mutual Fund Rating System (MFRS) and MARS whitepapers and can be broadly understood as the Pearson correlation between the ranked future returns and ranked component vectors. Here, we bootstrap at both the universe and category levels. As a baseline, we first compute the total IC for each component at the universe level, we resample with replacement 10,000 times, computing IC for each sample, and from the sample distribution form the 90% confidence interval. We do the same at the category level where sample sizes can be lower. We count how many category IC lower 90% confidence intervals are greater than zero and how many category IC are greater than zero. We show these results in Table 3 below.

Table 3: Universe and category IC for existing components

	Independent variables listed here					
Total IC across dataset	0.083	0.065	0.022	0.055	0.068	0.041
Bootstrap upper 90 IC CI	0.086	0.069	0.025	0.059	0.071	0.044
Bootstrap lower 90 IC CI	0.079	0.062	0.019	0.053	0.065	0.038
Categories w positive IC lower 90 CI	72%	64%	42%	75%	50%	50%
Categories w positive IC	82%	79%	51%	82%	61%	72%

Observations include

1. Component IC and standard error at the universe level are as expected. For example, [redacted] have the strongest IC but widest confidence intervals. Note also the lower confidence intervals of each component are above zero.

2. We expect higher variation in results at the category level. The strength of signal as measured by IC can vary by category and through time. All components are associated with greater than 50% positive IC.

Chi-square test for independence. Here the null hypothesis is that a selected component and the dependent variable are independent. We test for the absence of a relationship between components and returns ranks where independence implies the absence of predictive properties. Note the code library used for this test outputs p-value as -log(p) so for example -log(0.10) = 1 -log(0.01) = 2. So, for 90% significance we reject the null for all p-value expressions less than one; for 99% significance we reject the null for all p-value expressions less than two. This test can be highly significant where computed p-values are quite low. We retain this library's p-value expression for readability and comparison and results are shown in Table 4 below.

Table 4: Universe and category tests for independence

	Independent variables listed here					
-log(p-value) chi-square test for indy	13.5	11.1	44.5	7.9	11.7	43.2
Categories significant p-value (90)	96%	99%	92%	99%	100%	89%
Mean category -log(p-value)	9.7	11.4	10.4	29.0	25.6	10.9
Mean category -log(p-value) rank	4.5	4.1	3.4	2.7	2.9	3.4

Observations include

1. At the universe level, we reject the null hypotheses for all components, and there exists a relationship between component and return ranks. In this test, that relationship appears stronger with [redacted].
2. At the category level we find overwhelming evidence that the model components and future returns are not independent. But we also see different relative strength at the category level. For example, [redacted] show more dependence than at the universe level, while [redacted] show less.

Lasso regression coefficients. Here we conduct the first multivariate test that evaluates components together. The purpose here still resides in the realm of component importance; we will not evaluate multivariate prediction results until later sections. For lasso regression (L1 regularization), we add to the standard OLS equation a regularization term that includes a hyperparameter which penalizes OLS coefficients by absolute value. This formulation may drop the coefficients of redundant components to zero. See appendix for more on Lasso regression. Please see appendix for addition details on lasso.

The code library used here inputs a range of hyperparameter values, conducts k=10 k-fold cross validation, then selects the coefficient set with lowest mean squared error (MSE). Code can also systematically select a sparse solution (as defined by standard errors) which may include more zero coefficients and reveal any sensitivities. In Table 5 below we collect coefficients and note the occurrence of zero or "dropped" coefficients.

Table 5: Universe and category lasso coefficients

	Independent variables listed here					
Lasso Coefficients	0.07	0.06	0.03	0.07	0.08	0.04
Lasso Coefficients (sparse)	0.06	0.04	0.01	0.05	0.06	0.03
Catgeories w nonzero weights	86%	96%	89%	93%	85%	83%
Catgeories w nonzero weights (sparse)	50%	47%	44%	64%	53%	29%

Observations include

1. At the universe level no coefficients were dropped to zero, including the sparse solution. This could of course reflect the dilution of information/signals at such a high n. Note here the difference between near zero and zero; the model does not compute a zero coefficient at the universe level which means it is not suggesting an optimal subset of components. However, at the universe level the [redacted] coefficient is relatively low and very near zero and suggests lower importance in a multivariate setting.
2. At the category level the model rarely suggests an optimal subset of components. This is obviously increases when we force sparsity. On the sparsity row, we note [redacted] is dropped most frequently while [redacted] is dropped the least.

2. Candidate Components

MARS 4.0 was not intended to break new ground from a research or modeling perspective. Rather, 4.0 was intended to bring the technology and methodology used in MARS to align with MFRS so that going forward, research into any changes in methodology or the adaptation of new components could be investigated in parallel given the relative similarity of the underlying investments. Concurrent with this MARS research, the quantitative manager research (QMR) team has been developing the Alternative Investment Ratings System (AIRS), and four components from AIRS show promise in the managed account space. These are described below then tested in the same fashion as above.

Independent variables listed here

returns is relatively easier when the dispersion within a peer group is high. In this sense, PWER is another way of measuring risk-adjusted skill.

Table 6: Universe and category IC

	Bias Ratio	PWER	Track Record	Liquidity Risk
Total IC across dataset	0.014	0.142	-0.017	0.168
Bootstrap upper 90 IC CI	0.017	0.146	-0.014	0.171
Bootstrap lower 90 IC CI	0.011	0.139	-0.020	0.165
Categories w positive IC lower 90 CI	49%	72%	42%	63%
Categories w positive IC	61%	82%	51%	72%

Observations include

1. At the universe level, three components have positive IC values and do not include zero in the lower confidence interval. PWER and liquidity risk show strong universe-level IC figures.
2. We expect higher variation in results at the category level. The strength of signal as measured by IC can vary by category and through time. All components are associated with greater than 50% positive but only PWER and liquidity risk show as much consistency as existing components.

Table 7: Universe and category tests for independence

	Bias Ratio	PWER	Track Record	Liquidity Risk
-log(p-value) chi-square test for indy	6.79	15.82	29.22	44.90
Categories significant p-value (90)	97%	99%	92%	100%
Mean category -log(p-value)	8.8	18.1	10.4	30.8
Mean category -log(p-value) rank	3.5	2.4	2.3	1.8

Observations include

1. At the universe level, liquidity risk appears to show the least independence to the dependent variable. But instead of PWER it is accompanied by track record, which scored quite low in the IC tests. PWER does show more dependence than any existing component.
2. At the category level, track record falls off in strength of dependence while PWER and liquidity are as consistent as their universe-level metrics.

Table 8: Universe and category lasso coefficients

	Bias Ratio	PWER	Track Record	Liquidity Risk
Lasso Coefficients	0.03	0.13	-0.02	0.15
Lasso Coefficients (sparse)	0	0.10	0	0.12
Catgeories w nonzero weights	81%	92%	78%	99%
Catgeories w nonzero weights (sparse)	44%	64%	33%	74%

Observations include

1. At both the universe and category level lasso coefficients seem to suggest the same pattern as above: PWER and liquidity risk are the most important.
2. At the category level we do see similar parsimony without sparsity. But when we force sparsity, PWER and liquidity risk again show the most consistency.

3. Modified (Alpha) Components

MRO specifically requested an investigation into choices going into alpha factors. More generally, MRO appeared to have serious questions regarding poor alpha regression fits. Here we attempt to provide entirely new alpha components to test against the existing alpha component. As with existing MFRS and MARS exposition, note the following terminology: the independent variables in the alpha calculations are "factors;" they are regressed onto 36-month (short and medium alpha) and 37-120-month returns (long alpha), this step is conducted entirely in return space. Short, medium, and long alphas become components when ranked and subsequently included alongside the other model components.

As shown in Equation 1 below, the existing method computes short (12-month lookback) and medium (13-36-month lookback) alpha in the same ordinary least squares (OLS) regression formulation where $r_{i,t}$ is the return on SMA*i* at time *t*, $r_{f,,t}$ is the return on a risk-free asset, DMY_i is dummy that is set to 1 if the observation is from the most recent 12 months and zero otherwise, *Factor*1 through *FactorN* are the returns for category specific factors. This formulation returns the SMAshort alpha ($\alpha_{23i} + D_i$) and medium alpha ($\boldsymbol{\alpha}_{i,2-3}$). Long alpha is computed unto itself using standard asset pricing OLS formulation only we overlay a procedure to account for managers with limited data. These calculations are detailed exhaustively in the MFRS and MARS whitepapers.

Equation 1: Short and medium alpha formulation

$$r_{\mathrm{i,t}} - r_{\mathrm{f,t}} = \alpha_{i,2-3} + D_i DMY_t + \beta_{1i}(Factor1)_t + \beta_{2i}(Factor2)_t + ... + \beta_{ni}(FactorN)_t + e_t$$

Though we retain the same equations, we can select different betas (factors). Both MFRS and MARS use prior research intended to tailor factor selection to asset classes and categories. For example, a U.S. large cap strategy and a global commodities strategy simply do not have the same risk factors. There may be a large intersection, but the risk factors do not always have the same direction and magnitude. We test the following iterations of alpha components:

1. The current regression specification of the model.
2. A seven-factor model to compute alphas across all categories. This approach is similar to CSIA's Capital Market Expectations (CME) Model for alternatives that consisted of a domestic equity market excess return (S&P 500 minus T-bills), an equity size premium (Russell 2000 minus S&P 500), an equity value premium (Russell 3000 Value minus Russell 3000 Growth), an international equity premium (MSCI World minus T-Bills), an interest rate term premium (10-year Treasury minus T-Bills), a corporate credit premium (Corporate BBB minus T-Bills), and a commodities premium (GSCI minus T-Bills). Given the significant variations across categories, it is not clear that this approach is conceptually sound.
3. A single-factor model that simply uses the category median as independent variable in the spirit of a peers-based comparison. This is a deterministic approach but jibes with the broader purpose of relative ratings model, e.g., hit rates are counted as portfolios per grade category that outperformed the category median.
4. We consider a methodology change and calculate a single 36-month alpha rather than separate the lookbacks into distinct sub-periods as is done for short and medium alpha (shown in Equation 1 above)

5. At this point we have three factor models and four different alpha outputs, all of which are computed in standard return space. We also consider dimension reduction where we perform the same regression calculations as described above, only we affect the multivariate factor return matrices with PCA dimension reduction. This applies only to the three- and seven-factor models since there can be no reduction of the single-factor case. As described in Section 1, we systematically clip the bottom 20% low-variance principal components, then come back to return space via matrix multiplication and compute our four alphas.

So, we have five alpha outputs: existing, alpha median, alpha 7F, alpha existing/PCA, and alpha 7F/PCA. While we have computed the same universe- and category-level metrics as shown in sections 2 and 3, for Table 9 we select what we perceive to the be the most salient metric of the three tests.

Table 9: Universe and category alpha metrics

		Short Alpha	Medium Alpha	Long Alpha	36-Month Alpha		Short Alpha	Medium Alpha	Long Alpha	36-Month Alpha
Total IC across dataset	Alpha Existing	**0.08**	**0.07**	**0.02**	**0.09**	Catgeories w positive IC	**82%**	**79%**	**32%**	**86%**
	Alpha 7F	0.07	0.05	0.03	0.06		78%	72%	43%	76%
	Alpha Median	0.02	0.02	-0.06	0.03		63%	60%	38%	61%
	Alpha Existing/PCA	0.08	0.08	0.03	0.08		72%	71%	35%	82%
	Alpha 7F/PCA	**0.13**	**0.12**	**0.07**	**0.06**		**90%**	**85%**	**44%**	**69%**
-log(p-value) chi-square test	Alpha Existing	**14.5**	**11.0**	**44.3**	**14.5**	Mean category -log(p-value)	**9.7**	**11.4**	**22.1**	**13.9**
	Alpha 7F	11.8	14.3	38.9	16.1		9.1	11.2	21.1	12.2
	Alpha Median	11.7	13.2	44.5	14.7		8.4	14.2	28.7	15.2
	Alpha Existing/PCA	15.0	15.9	38.2	12.7		10.8	14.2	21.5	13.4
	Alpha 7F/PCA	**21.0**	**23.5**	**44.7**	**15.3**		**13.2**	**15.5**	**12.2**	**14.0**
Lasso coefficients (sparse)	Alpha Existing	**0.03**	**0**	**0**	**0.04**	Cats nonzero coeff (sparse)	**38%**	**29%**	**65%**	**26%**
	Alpha 7F	0.03	0.00	0.00	0.02		31%	18%	57%	32%
	Alpha Median	0	0	-0.04	0.01		25%	28%	65%	28%
	Alpha Existing/PCA	0.05	0.05	0	0		35%	43%	68%	40%
	Alpha 7F/PCA	**0.14**	**0.14**	**0.07**	**-0.10**		**58%**	**57%**	**60%**	**58%**

Observations include

1. For nearly all metrics the one-factor model is inferior and is thus discarded.
2. The 36-month calculation is either duplicative or inferior to existing medium calculations, is almost never superior to short alpha, and is thus discarded.
3. For lasso models, some test components seem to show duplication. Note this is the case with the existing factor selection. All but the seven-factor-related alpha components drop a coefficient to zero.

4. For dimension reduction, the existing factor model appears to stay flat or only slightly gain predictive properties. This could be because these tailored factor models are well specified.
5. For dimension reduction, the seven factor-model (7F/PCA) appears to gain predictive properties when we reduce dimension. This could be because the seven-factor model has both more information and noise, and when we strip out the low-variance principal components we reduce the latter.
6. Of all the newly computed alphas, the 7F/PCA components show the most promise. So, we compare SMA-weighted category IC distributions across the existing alpha components and 7F/PCA components in Figure 3 and Table 10 below. All three 7F/PCA alpha components show higher IC metrics, their distributions appear centered over higher IC while also showing more area above positive IC, and they show larger positive skew.

Figure 3: SMA-weighted distributions of alpha IC

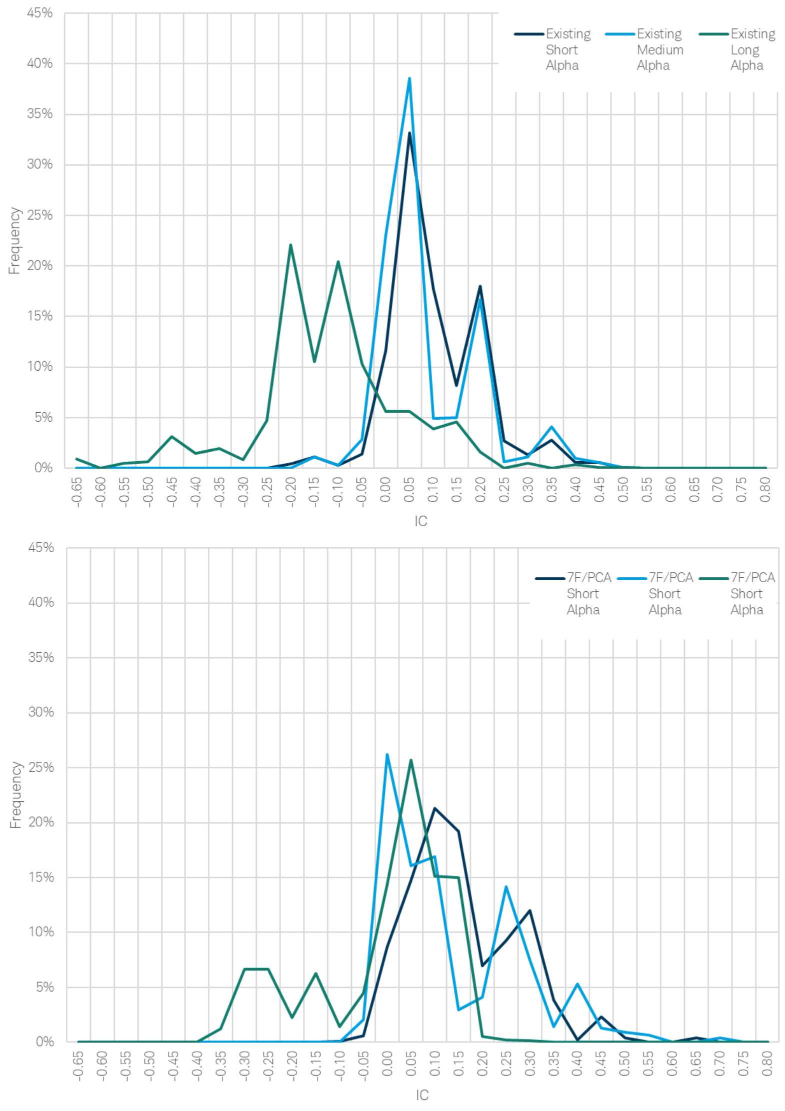


Table 10: SMA-weighted distributions of alpha IC

	Existing Short Alpha	Existing Medium Alpha	Existing Long Alpha	7F/PCA Short Alpha	7F/PCA Medium Alpha	7F/PCA Long Alpha
Min	-0.56	-0.17	-0.66	-0.56	-0.09	-0.39
Max	0.43	0.79	0.51	0.63	0.67	0.33
Mean	0.08	0.06	-0.14	0.13	0.11	-0.03
Median	0.05	0.02	-0.11	0.10	0.06	0.01
Sigma	0.09	0.11	0.16	0.12	0.14	0.14

These differences in alpha component predictive properties provide an opportunity to examine alpha regression fits. We collect regression output at the SMA-level per period for short and medium alpha for both the existing and 7F/PCA factor approaches.

R-squared or coefficient of determination. We collect the median r-squared per category per period. This is rolled up to the category level by SMA-weighted average.

Gibbons Ross Shanken (GRS) tests the hypothesis that all alphas are jointly zero. GRS is intended to determine the significance of asset pricing models, which is by definition what our factor models are. For if an asset pricing model for a basket of manager returns results in all alphas equal zero then the asset pricing model is perfectly specified and/or the existence of active manger skill is in question. We compute a GRS-specific test statistic for a chi-squared test and collect the p-value. Again, we compute per category per period and roll up to the category level by SMA-weighted average.

One-way ANOVA is an f-test to check for common means between data groups. We do this between short and medium alphas where the null hypothesis states that short and medium alpha are from the same population. This complements the above information as it relates to assessing the number of alphas going into the final modeling step. Note we do this at the category level.

Results are shown in Figure 4 and 5 below. We show category GRS-test and f-test p-values for both short and medium alphas on the y-axis and category r-squared on the x-axis.

Figure 4: Alpha regression output (existing factors)

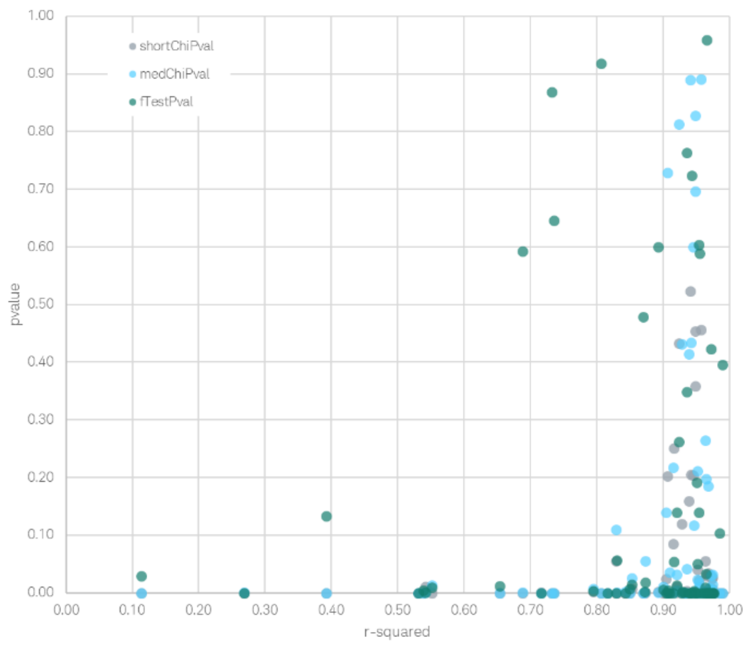
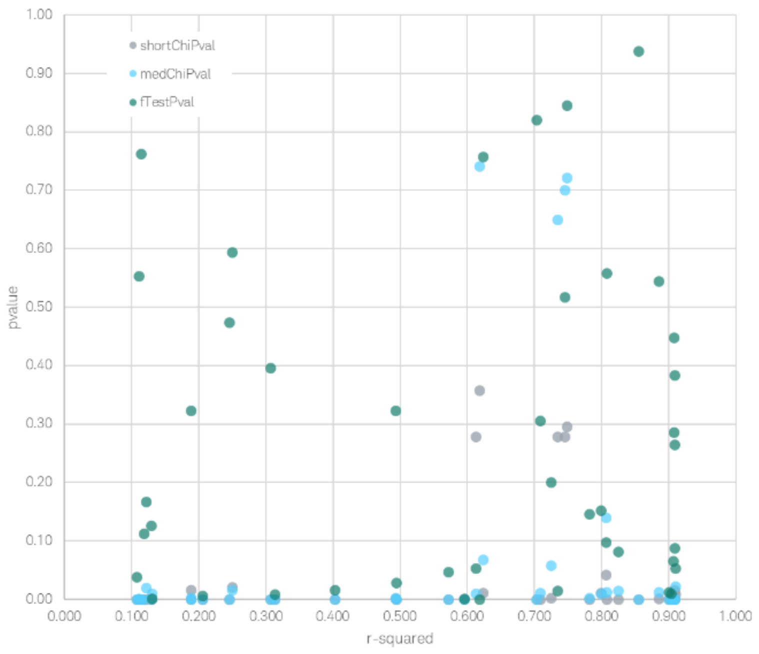


Figure 5: Alpha regression output (7F/PCA)



Observations include

1. We showed in Figure 3 and Table 10 above that 7F/PCA alphas have better prediction metrics than existing methods. But we can see in the regression-level scatter charts in Figures 4 and 5:
    a. R-squared fits are much lower for 7F/PCA. Which is to be expected after transforming the factor (or beta) return space.
    b. We reject the null that short alphas are jointly zero for 49/72 categories in Figure 4 and 63/72 categories in Figure 5.

Independent variables listed here

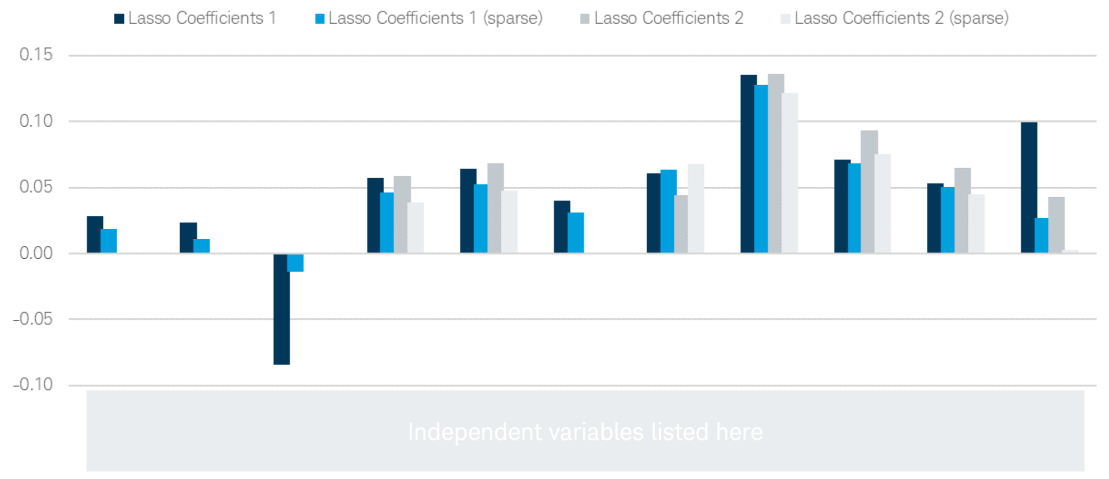
4. Derive Component Test Set

As shown in sections 2 and 3, several additional components show promise for inclusion in MARS. From AIRS, [redacted] show superior metrics across IC, chi-square, and lasso tests. [redacted] components generated with the 7F/PCA model also show improved predictive properties. And so, we include these in a broader component test where we include the six existing and five promising components.

As above, we run lasso regressions and collect coefficients with the lowest MSE across k=10 k-fold cross validation. Again, we show the coefficient with and without forced sparse solutions. In addition to the coefficient figures, here we also confirm our component selections with a forward and backward sequential feature selection algorithm where the objective function is minimizing MSE. Sequential search algorithms add (forward) or remove (backward) components from possible subsets. While a helpful check, these algos are non-exhaustive, i.e., there exists some combination of components the algo has not tested.

Figure 6 below shows two lasso runs. In the first run (blue), we include all 11 components. Neither forward nor backward sequential search algorithms dropped a component. Nor did a coefficient go to zero which suggests we do not have redundant information. Here we can see the 7F/PCA alpha components have larger coefficients than the existing alpha components. And the existing long alpha component actually has a large negative coefficient. PWER, liquidity risk, AUM, and firm also have larger coefficients. We collect these seven components for a second lasso run (gray). Again, the sequential search algorithms did not drop a component. The lowest MSE in each lasso runs was 0.0781 with 11 components and 0.0785 with seven components. This suggests this lasso model does best with the most information. But also, that the subset is efficient and a good candidate for further testing in the next section.

Figure 6: Lasso coefficients for two component sets



While this second set of components shows promise, it is primarily a test set computed at the universe level—which has a very high n. Practitioners of both MFRS and MARS are aware that component relationships to the dependent variable strengthen and weaken through time and that it is more effective to take a diversification approach to predictor variables. This is illustrated in Figures 7-9 below where we show SMA-weighted IC across time for components. We break these out across the alphas and the remaining components to observe periods of complementary diversification. Metrics for these time series are shown in Table 11 below.

Figure 7: Existing [redacted] component IC

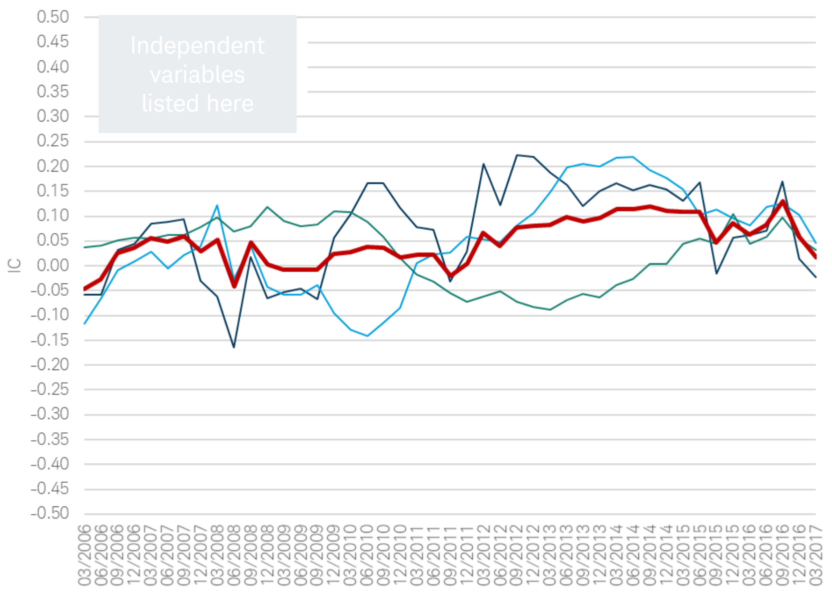


Figure 8: 7F/PCA [redacted] component IC

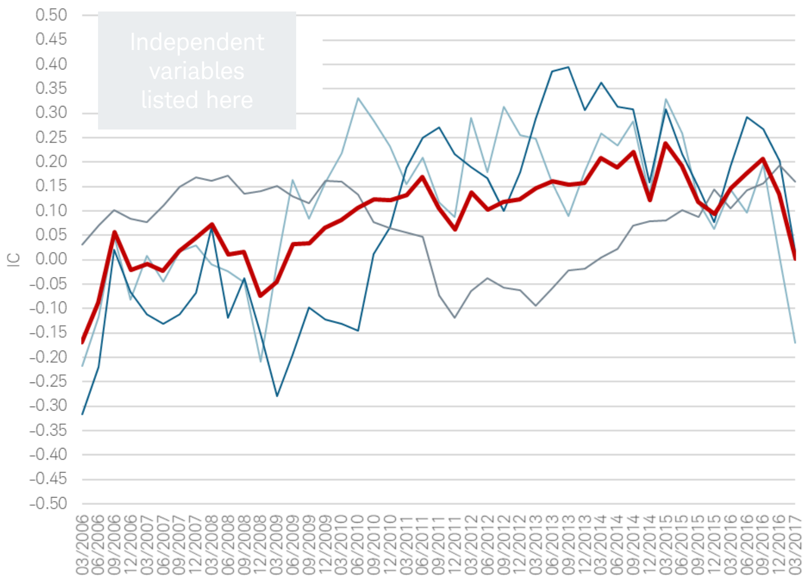


Figure 9: Additional component IC

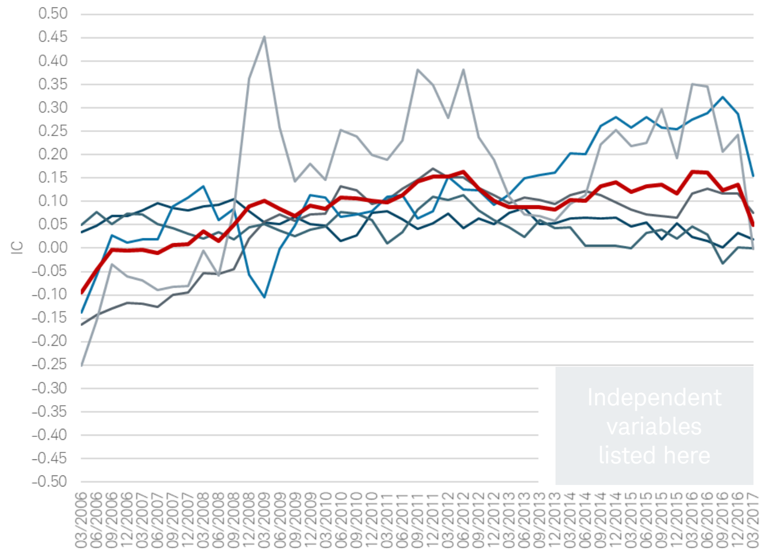


Table 12: Component IC time series

	Independent variables listed here										
Min	-0.17	-0.14	-0.09	0.00	-0.16	-0.03	-0.14	-0.25	-0.22	-0.32	-0.11
Max	0.23	0.23	0.12	0.11	0.17	0.11	0.32	0.45	0.33	0.40	0.20
Mean	0.07	0.05	0.03	0.06	0.05	0.04	0.12	0.15	0.11	0.08	0.07
Median	0.08	0.05	0.05	0.05	0.09	0.04	0.11	0.19	0.14	0.10	0.09
Sigma	0.09	0.10	0.06	0.02	0.09	0.03	0.11	0.16	0.14	0.19	0.08

We also show intra-component correlations in Table 13. Note we are showing the existing component set and the test set. We can see the [redacted] components are somewhat correlated but moving forward we do not evaluate these in the same sets. Of note, [redacted] does appear to show elevated correlation to the alphas and perhaps the other AIRS component liquidity risk. Note however the mean correlation within the existing component set is -0.01 and 0.08 for the test set.

Table 13: Component correlation matrix

Independent variables listed here

Independent variables listed here

	Exi ha	Exi Alpha	Exi a	AU	Fir	Ca	PW	Liq	7F a	7F pha	7F a
	1	0.20	0.05	0.04	-0.02	-0.07	0.23	-0.12	0.75	0.12	0.03
	0.20	1	0.07	0.02	-0.02	-0.20	0.32	-0.12	0.11	0.76	0.05
	0.05	0.07	1	-0.05	-0.06	-0.07	0.62	-0.02	0.02	0.03	0.81
	0.04	0.02	-0.05	1	-0.14	-0.03	-0.02	0.02	0.03	0.03	-0.02
	-0.02	-0.02	-0.06	-0.14	1	0.07	-0.01	0.06	0.01	0.02	-0.05
	-0.07	-0.20	-0.07	-0.03	0.07	1	-0.13	0.06	-0.05	-0.18	-0.05
	0.23	0.32	0.62	-0.02	-0.01	-0.13	1	0.23	0.28	0.39	0.70
	-0.12	-0.12	-0.02	0.02	0.06	0.06	0.23	1	0.09	0.13	0.07
7F	0.75	0.11	0.02	0.03	0.01	-0.05	0.28	0.09	1	0.17	0.02
7F/I	0.12	0.76	0.03	0.03	0.02	-0.18	0.39	0.13	0.17	1	0.02
7F	0.03	0.05	0.81	-0.02	-0.05	-0.05	0.70	0.07	0.02	0.02	1

5. Evaluate Test Set

As an additional exercise, we compare the existing component set ([redacted]) to the test set derived above ([redacted]). We use an abbreviated version of the procedure used in the production of actual rankings, which is simply the dot product of weight and rank vectors. We have spent our time thus far collecting rank vectors and we require a weight vector.

The current MARS weighting scheme was derived using a nonlinear solver without constraints on full in-sample datasets at the category level where we minimize the objective function of one minus IC. Note here this is not a linear model but a numerical method i.e. a solver. Categories with similar holdings and weight outputs were then partitioned into 18 model cohorts, evaluated for further modifications including constraints, and these 18 tuned cohorts are the current MARS weighting scheme. Note here the processes thus far have been almost entirely quantitative; where expert judgment has been deployed only in the form of constraints (if else the solver will do things like short and add leverage) and collecting categories of similar holdings. Building these cohorts and the ultimate modeling scheme represents hundreds of hours of research, the duplication of which is beyond the scope of this writeup.

But we still wish to evaluate the component test set and so we take the existing cohorts, nonlinear solver, constraints, and objective function and compute an in-sample weighting scheme for the test components.

And we emphasize this is an in-sample test, the general outcomes of which are by definition overly remarkable. We compute prediction vectors for the existing components/weights and the test components/weights. For simplified comparison metrics we compute the root mean squared error (RMSE) and IC for the actual and predicted vector pairs.

RMSEs (note these are expressed in IC units) are shown in Figure 10 below, and we see the test set predictions are closer to actuals for all model cohorts. In Figure 11 below, we compare SMA-weighted distributions of category IC and their distribution metrics are in Table 14. The test set shows improved IC performance. We have slightly clipped the area of the left tail, dramatically increased the area of the right tail, and centered over a higher IC—like the alpha component distribution charts above. As shown in the standard deviation row in Table 14, somewhat expected after the preceding component tests, the inclusion of more predictive but volatile components appears to result in a more volatile prediction distribution. Though entirely in-sample, these results suggest the inclusion of new model components could have a positive impact on performance. All in-sample backtests are imbued with the duality of being compelling and possibly spurious. We required multiple hurdles for a component's inclusion in the test set but this backtest is broad and likely requires an additional amount of research proportionate to the amount that preceded it.

Figure 10: Cohort RMSE differences (existing minus test)

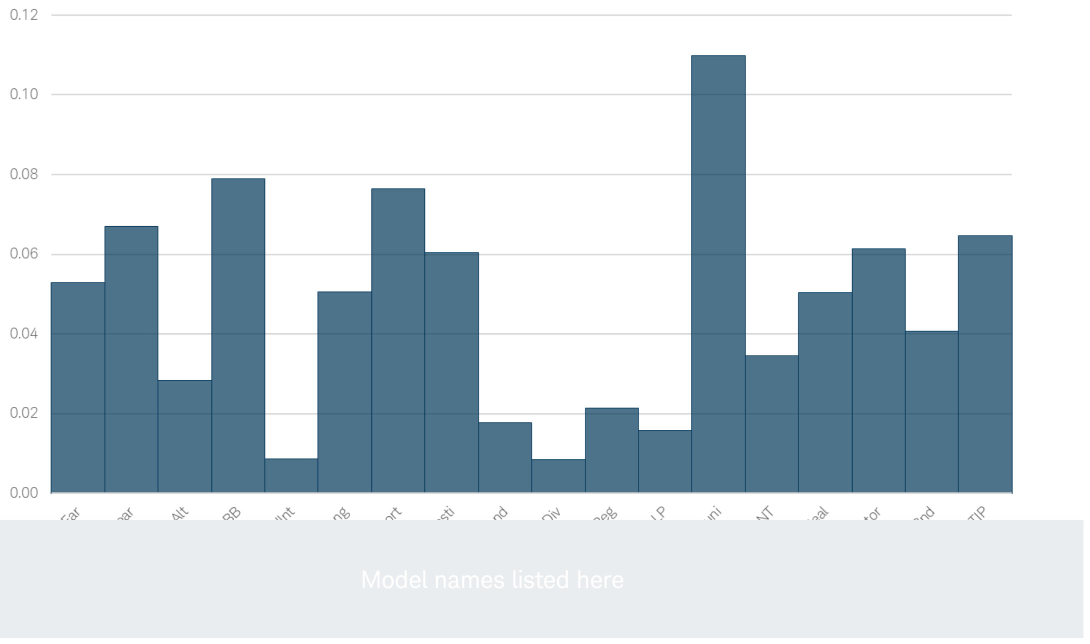


Figure 11: SMA-weighted IC distributions

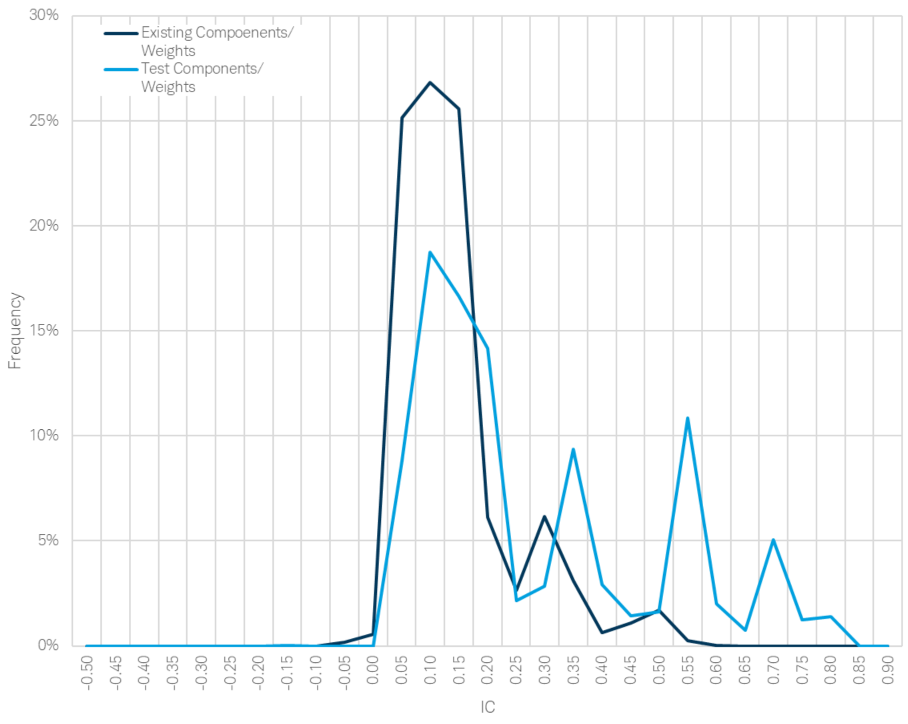


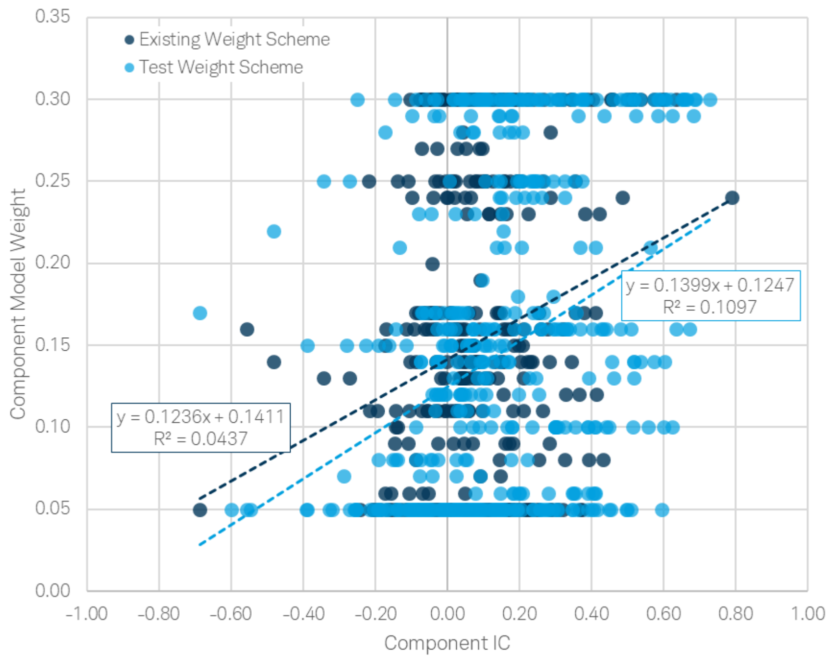
Table 14: SMA-weighted IC distributions

	Existing Compoenents/ Weights	Test Components/ Weights
Min	-0.05	-0.17
Max	0.58	0.80
Mean	0.11	0.26
Median	0.10	0.16
Sigma	0.10	0.21

6. Considering Weight Selection

The choice of model weights should reflect the relative predictability of a model component within a category or cohort of categories. In MARS 4.0 we solved for maximizing the IC then assigned categories to cohorts of similar economic rationale and optimization weights. We then recalculated optimal weights for those cohorts to reduce the chance of overfitting the in-sample data. In Figure 12 below we show the existing component weighting scheme and the test set weighting scheme to illustrate our general aim. The general optimization approach is well aligned where more predictive component IC values have higher weights and less predictive component IC values have lower weights.

Figure 12: Optimized weights per component IC



While developing the AIRS model, we identified a potential concern with our current optimization methodology which focuses on maximizing the IC of the sample. We found the current optimization leads to unwanted volatility in outcomes when considering other performance criteria such as hit rates and letter grade relative performance. A better objective function for the optimization is to consider both the magnitude of IC as well as its volatility through time using a [redacted] The objective function in Equation 2 provides a means of calibrating the model to possibly provide the right level of consistency in ratings performance.

Equation 2: Candidate objective function

[redacted]

As an example of how this new optimization may be superior to the existing approach, we share a single example where we took sample data for large cap growth from 2006-2017 and calculated the optimal weights of our existing model components using the current methodology (top chart) versus the updated utility function using a lambda of one (bottom chart). We found that the in-sample performance of the new methodology has more consistent results. In this example not only is the overall IC higher, but it is more consistent as measured by the mean IC divided by the standard deviation of IC ratio. Similarly, the hit rate is higher in this example and more stable as well. Significantly more testing is required to determine the potential benefits of using this optimization approach more broadly.

Independent variables listed here

7. Conclusion

We revisited model choices by testing existing, new, and modified components for redundancy and predictive properties. During this step we explained the alpha regressions that build alpha components and we showed and tested alternative approaches. We used these results to generate and compare in-sample performance metrics between the existing component set and a test set, both of which were also evaluated for redundancy. And we showed a possible modification to the nonlinear model that generates MARS weights. We have taken the reader through alpha regression, component construction, weights generation, and ultimately broad performance metrics. We have also shown that all processes are first informed by sound quantitative methods while secondarily attempting to strike a balance between black-box and discretionary decisions. This journey should be considered a full-blown overview of the model's structure because no other processes exist in the MARS model.

We have, of course, shown that model enhancements are possible. But we have not shown this evidence is not spurious. Even the best faith analysis of this type will certainly uncover different and possibly better solutions. Most importantly, any improvements must ultimately fit the model's business needs. While it is easy to simulate improved performance, if things like turnover become problematic or changes in alpha ranks are not intuitive or interpretable it is likely impossible to incorporate these enhancements. In order to move to a formal model proposal, these results need to be at a minimum cross-validated in MFRS, which would likely also need to be updated in parallel. Additionally, these results need to be socialized and vetted with relevant users and oversight bodies to ensure they meet additional business needs for this model; the details of which involve the inclusion and feedback from numerous user groups and stakeholders.

8. Appendix

8.1 Selected Component Category IC Tables

Table 15: Existing MARS component IC by category

Table 16: Candidate MARS component IC by category

8.2 More on Principal Component Analysis (PCA)

PCA is a change of basis where we project n x m data structures onto to lower dimensional subspace. Two- and three-dimensional examples are easy to represent graphically. When we get to higher m-dimensions we are dealing with high levels of abstraction. The curse of dimensionality notwithstanding, two-dimensional PCA notions hold for any m-dimensional data.

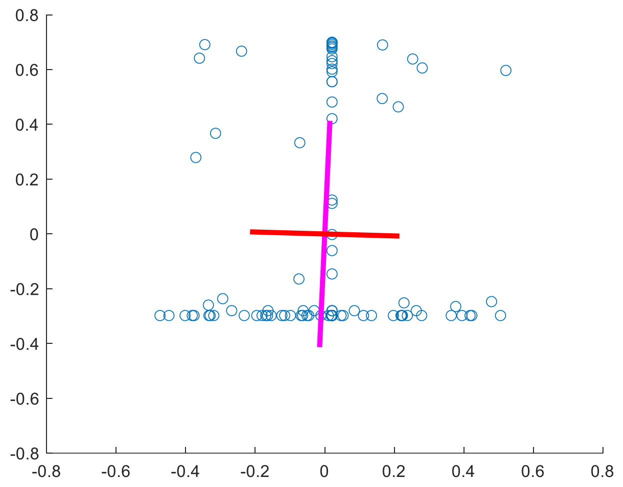
The change of basis is an expression of variance. And we seek to concentrate this collection of variance in as few dimensions as possible. Figure 14 below is scatter of long alpha and cash inflow components. We select 1,000 rows from the large blend category and show demeaned percentile ranks. In 2D space, we seek to find the axis that projects the data across the widest span while minimizing the perpendicular distance from each datapoint to that axis. This is given by the eigenvalues and eigenvectors of the covariance matrix. Principal components (PC) are the m x m eigenvectors of the covariance matrix. Here we are in an m-orthogonal subspace where the eigenvectors are the PC coordinates, and their corresponding eigenvalues are span. We can project rank-space data onto the PC axis with the following coordinates.

Table 17: Project rank-space on PC1 axis

	Endpoint	Endpoint
PC1 axis (magenta)	[-eigenvector (1,1) *sqrt(eigenvalue(1)), eigenvector (1,1) *sqrt(eigenvalue(1))]	[-eigenvector (2,1) *sqrt(eigenvalue(1)), eigenvector (2,1) *sqrt(eigenvalue(1))]
PC2 axis (red)	[-eigenvector (1,2)	[-eigenvector (2,2)

	*sqrt(eigenvalue(2)), eigenvector (1,2) *sqrt(eigenvalue(2))]	*sqrt(eigenvalue(2)), eigenvector (2,2) *sqrt(eigenvalue(2))]

Figure 14: rank-space and PC1 (mag) PC2 (red) axis



We have shown the connection between rank-space and eigenspace. In Table 18 below, we can see that PC1's eigenvalue accounts for about 80% of the influence within the data structure. Additionally, we see that cash inflow has a huge loading in PC1, which suggests it dominates this PC. This is the kind of EDA we did in section 1. To test new alphas in section 3 we reduced the dimension of datasets by discarding the bottom 20% of low variance PCs then coming back to return-space. Per Table 18, we could easily discard the low variance PC2 and come back to rank-space to perhaps conduct additional testing. We can do this with the following matrix algebra procedure:

- retainedPC = number of PCs being retained, generally 80% of cumulative eigenvalues
- Z = X*eigenvectors where X is the demeaned rank-space matrix
- Xreduced = Z(:,1:retainedPC)*eignvectors(:,1:retainedPC)
- Add back means to columns

Note that if we did not drop any PC this procedure simply brings us back to the demeaned rank-space X, we add means to columns, and see our original data.

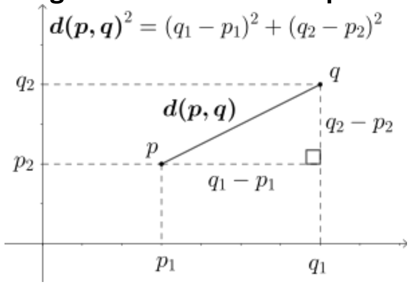
Table 18: PCA output

	Eigenvalues	
	0.1710	0.0462
	Eigenvectors	
	PC1	PC2
[redacted]	0.0358	0.9994
[redacted]	0.9994	-0.0358

8.3 More on k-Means Cluster Analysis

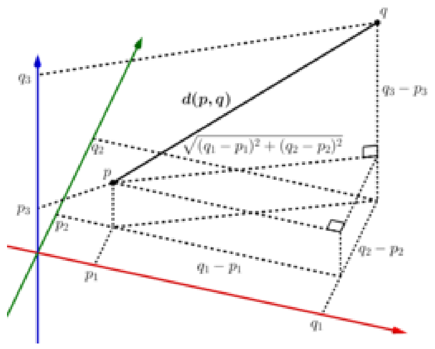
First we define Euclidian distances. Although this distance measure is not unique to this procedure, the deployed code library defaults to Euclidian distances[1]. Recall the two-dimensional measure:

Figure 15: 2D Euclidian space



This can be extended for n-dimensions i.e. for n independent variables:

Figure 16: ND Euclidian space



For points given by Cartesian coordinates in n-dimensional Euclidean space, the distance is:

Equation 3: ND Euclidian distance

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}$$

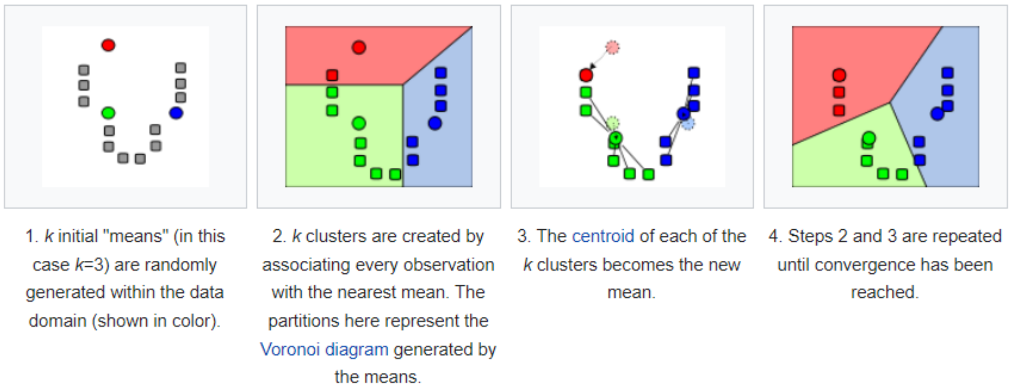
For k-mean cluster analysis we partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. We deploy Lloyd's algorithm which proceeds as follows[2]:

1. Choose k initial cluster centers (centroid).
2. Compute point-to-cluster-centroid distances of all observations to each centroid.
3. Individually assign observations to a different centroid if the reassignment decreases the sum of the within-cluster, sum-of-squares point-to-cluster-centroid distances.
4. Compute the average of the observations in each cluster to obtain k new centroid locations.
5. Repeat steps 2 through 4 until cluster assignments do not change, or the maximum number of iterations is reached.

[1] "Euclidian Distance," Wikipedia, Wikimedia Foundation Inc., 9 Apr 2022, https://en.wikipedia.org/wiki/Euclidean_distance.

[2] "K-Means Clustering," Wikipedia, Wikimedia Foundation Inc., 25 Apr 2022, https://en.wikipedia.org/wiki/K-means_clustering.

Figure 17: Demonstration of Lloyd's algorithm



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3. The centroid of each of the *k* clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached.

8.4 More on Lasso Regression

Regularization is the process of adding information to solve an ill-posed problem or to prevent overfitting. Regularization can be applied to objective functions in ill-posed optimization problems. The regularization term, or penalty, imposes a cost on the optimization function to make the optimal solution unique.

Lasso (least absolute shrinkage and selection operator) is a regression method that penalizes the absolute size of the regression coefficients. By penalizing or equivalently constraining the sum of the absolute values of the estimates some of the parameter estimates may be exactly zero. The larger the penalty the further estimates are shrunk towards zero. This is convenient if we are systematically selecting independent variables or when OLS coefficients have "blown up" due to multicollinearity. Recall for OLS the goal is to minimize the residual sum of squares (RSS) to estimate the coefficients[3]:

Equation 4: OLS

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

In the case of lasso regression we estimate the coefficients with an added L1 regularization term:

Equation 5: OLS + L1

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

The red L1 term is a sum of the absolute coefficient values penalized by λ (the hyperparameter). If λ=0 we get the same coefficients as OLS. If λ=1, the red L1 penalty constrains the size of the coefficients so that the coefficients can only increase by the same amount of decrease in RSS. Thus, the higher we set λ the more penalty is applied to the coefficients and some might be zero.

In Figure 18 below we show the highly disseminated graphic of lasso and ridge regularization constraints. We see the coefficient and RSS interplay in the red and teal shapes. For Equation 4 above OLS can be rewritten to express the red ellipses centered around the maximum likelihood estimator. L1 is the equation for the teal diamond. The optimized solution to Equation 4 is the intersection of these geometric

[3] User247233, "What is the Lasso in Regression Analysis," Cross Validated, Stack Overflow, 18 Jun 2018, https://stats.stackexchange.com/questions/348308/graphical-interpretation-of-lasso.

shapes and as λ approaches infinity the solution converges at [0,0]. We can see that the diamond constraint and in particular its vertices are more likely to induce a zero coefficient[4].

Figure 18: Linear model selection and regularization

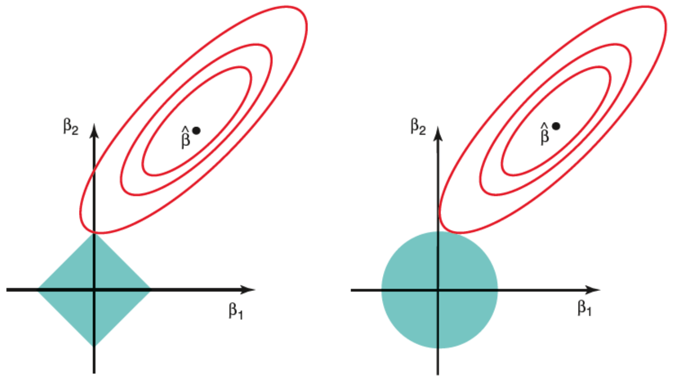
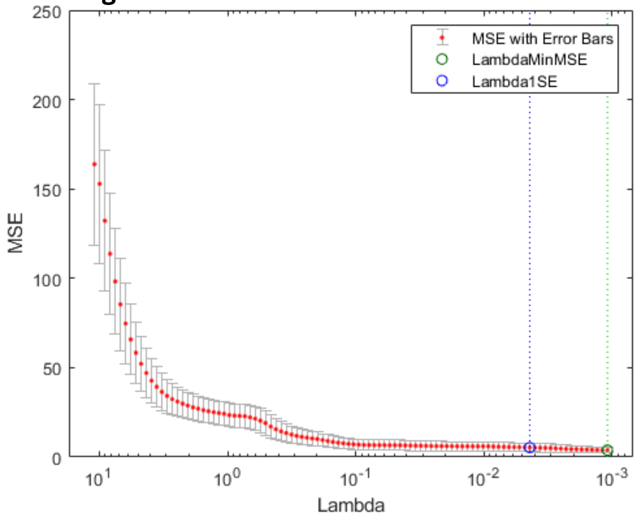


FIGURE 6.7. *Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.*

In the paper we systematically selected a sparse solution from an array of cross-validation MSE. Figure 19 below[5] shows MSE results per hyperparameter λ. The green marker shows the lowest MSE fit and the blue marker shows the sparsest solution within one standard error of the blue. This approach is entirely native to the code library.

Figure 19: Cross-validated MSE of lasso fit



[4] Xavier Bourret Sicotte, "Graphical Interpretation of Lasso," Cross Validated, Stack Overflow, 18 Jun 2018, https://stats.stackexchange.com/questions/348308/graphical-interpretation-of-lasso.
[5] "Lasso," MathWorks, accessed 25 April 2022, https://www.mathworks.com/help/stats/lasso.html.