Assignment #6

Nate Bitting

Introduction

The purpose of this assignment is to explore the use of principal component analysis (PCA) to alleviate multicollinearity that may exist among the predictor variables when building a multiple regression model. The dataset we will be working with is daily stock prices for twenty stocks (predictor variable candidates) included in a fund managed by Vanguard (VV). The first step in our analysis will be to perform a log transformation on the daily returns (current – previous stock prices) for all of the individual stocks in the fund as well as the market index (VV), which we will use as our response variable. The second step in our analysis, after performing the transformations on the predictor and response variables, is to plot a Pearson correlation matrix for all of the predictor variables against the response variable, which we'll call response_VV. This analysis will identify any correlations between the predictor and response variables and identify any multicollinearity among the variables.

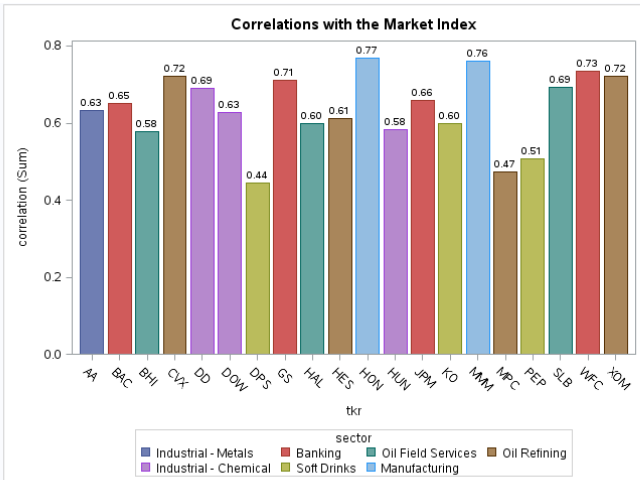
We will then perform PCA in order to deal with the multicollinearity in the data. Next, we will create two separate regression models; the first will include all of the daily log return variables against the log return of the fund (response_VV) and the second will leverage eight components we will get from our PCA analysis. We will then leverage a test data set to assess the predictive accuracy of each model. Lastly, we will assess the goodness-of-fit for both models and compare several metrics (adjusted R-squared, MAE, MSE) on both the training and test datasets for each model. The final goal is to determine if PCA improves the predictive accuracy of the model by removing the multicollinearity that may or may not exist among the predictor variables.

Results

Correlation Analysis

The first step in our analysis is to perform the log transformation on the twenty stocks and the Vanguard market index fund. Next, we will plot a Pearson correlation matrix for each stock against the market index (see Figure 1). As you can see in Figure 1, many of the variables have a positive correlation with the market index. You can see several stocks in the same sector have similar correlations with the response variable (eg. CVX and XOM, HON and MMM).

Figure 1 – Correlation for each stock against the Market Index (VV) by Sector



Principal Component Analysis (PCA)

The next phase in this assignment is to perform Principal Component Analysis (PCA) on the predictor variables. After running the PRINCOMP procedure in SAS, you can see the scree plot in Figure 3. The Kaiser Rule tells us to drop all components with eigenvalues below 1.0, which in this case would suggest we should retain only three components.

Figure 3 – Scree Plot

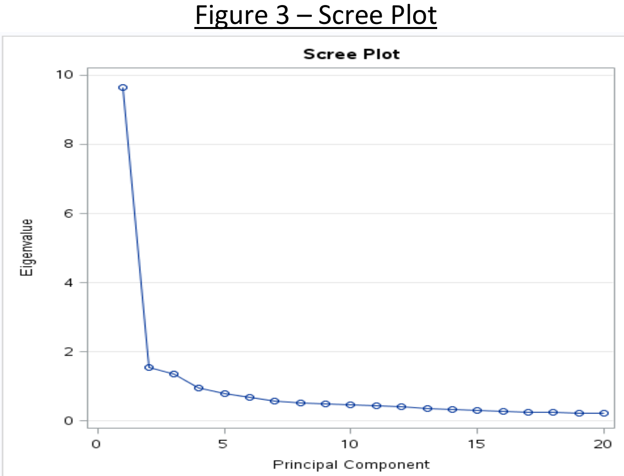
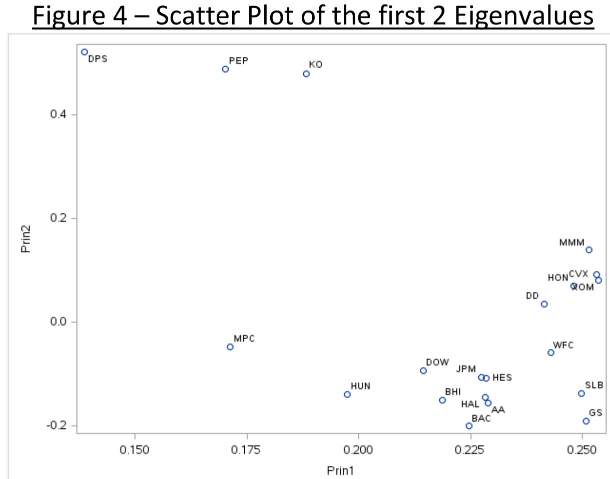


Figure 4 – Scatter Plot of the first 2 Eigenvalues



If you look at the chart in Figure 4, you can see that groupings have formed. DPS, PEP, and KO all appear to fall into component 2, which makes perfect sense since they are all in the in the same sector, Soft Drinks. You can see other similar groupings of stocks from the same industry such as CVX and XOM, both from the Oil Refining sector.

Regression Models Leveraging Principal Component Analysis

In this section, we will build two regression models in order to assess if performing PCA improves the predictive accuracy of the model.

Model 1 – All Log-Returns for each Stock against the Log-Return of the Market Index - VV

The first model we will assess include all log-return values for each stock as the predictor variables and the log-return of the Vanguard market index (response_VV). From a goodness-of-fit perspective, the residual plots all show constant variance for all observations as shown in Figure 5. Another indication that the model has a good fit is the straight line in the QQ plot for Model 1 (Figure 6). Lastly, the adjusted R-squared for Model one is 0.8919, which indicates a very good fit to the data. As can be seen in Figure 7, the variance inflation factors (VIFs) for the predictor variables are all below 5. Thus we can conclude that no multicollinearity exists among the predictor variables.

Figure 5 – Residual Plot for Model 1

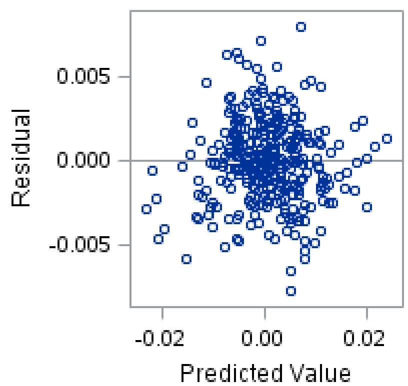


Figure 6 – QQ Plot for Model 1

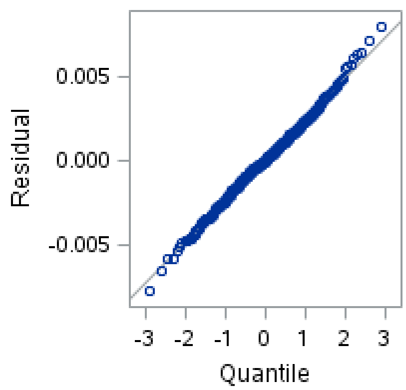


Figure 7 – VIFs for Model 1

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	0.00008640	0.00014092	0.61	0.5403	0
return_AA	1	0.01769	0.01317	1.34	0.1802	2.11490
return_BAC	1	0.03198	0.01165	2.75	0.0064	3.10927
return_BHI	1	-0.00111	0.01323	-0.08	0.9333	2.62997
return_CVX	1	0.04907	0.02536	1.93	0.0539	3.07524
return_DD	1	0.04674	0.02037	2.29	0.0224	2.51406
return_DOW	1	0.03642	0.01162	3.14	0.0019	1.88893
return_DPS	1	0.03670	0.01679	2.19	0.0295	1.54768
return_GS	1	0.04849	0.01555	3.12	0.0020	3.10450
return_HAL	1	0.00948	0.01466	0.65	0.5184	3.08758
return_HES	1	0.00359	0.01092	0.33	0.7425	2.10199
return_HON	1	0.12213	0.01924	6.35	<.0001	2.73505
return_HUN	1	0.02712	0.00836	3.24	0.0013	1.79852
return_JPM	1	0.00902	0.01708	0.53	0.5979	3.36439
return_KO	1	0.07903	0.02226	3.55	0.0004	1.93633
return_MMM	1	0.09796	0.02646	3.70	0.0003	2.98277
return_MPC	1	0.01673	0.00809	2.07	0.0394	1.32999
return_PEP	1	0.02911	0.02231	1.30	0.1929	1.68825
return_SLB	1	0.03776	0.01709	2.21	0.0279	3.13690
return_WFC	1	0.07587	0.01848	4.10	<.0001	2.59492
return_XOM	1	0.05467	0.02697	2.03	0.0435	2.98393

Model 2 – 8 components from PCA against the Log-Return of the Market Index - VV

In the second model, we have gone down the path of performing principal component analysis. As can be seen in Figure 8, the residuals appear to have constant variance in the model, similar to Model 1. The QQ plot for Model 2 (Figure 9) has a straight line, which is an indication of good model fit. Lastly, the adjusted R-squared for Model 2 is 0.8886, which is also a good indication of model fit.

Figure 8 – Residual Plot for Model 2

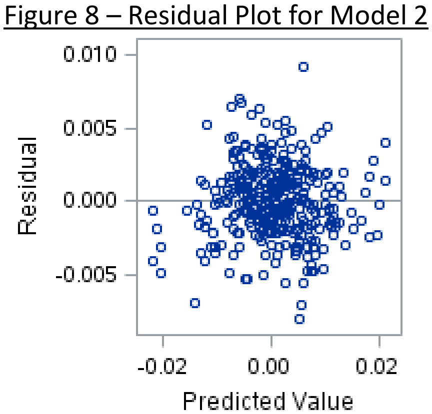


Figure 9 – QQ Plot for Model 2

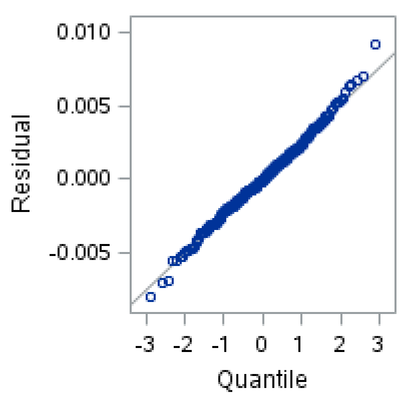


Figure 10 - VIFs for Model 2

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	0.00075978	0.00014045	5.41	<.0001	0
Prin1	1	0.00231	0.00004519	51.05	<.0001	1.00527
Prin2	1	0.00032245	0.00011425	2.82	0.0051	1.00868
Prin3	1	0.00070635	0.00012322	5.73	<.0001	1.00861
Prin4	1	0.00030481	0.00014536	2.10	0.0368	1.00636
Prin5	1	-0.00017356	0.00015516	-1.12	0.2641	1.00297
Prin6	1	0.00000315	0.00017108	0.02	0.9853	1.00766
Prin7	1	-0.00010331	0.00018604	-0.56	0.5791	1.02315
Prin8	1	-0.00040760	0.00020293	-2.01	0.0454	1.02271

As you can see in Figure 10, the VIFs for Model 2 are all very low and do not exceed 5, indicating no multicollinearity exists among the components used in the model.

Comparison of Model 1 and Model 2

After running the regression models on the training data, you can clearly see that both models have a good fit to the data. However, if we review the MSE and MAE for both the training and test datasets for both models, we can see some slight differences in model performance.

Table 1 – Model Comparison from Training and Test samples

		Model 1 – w/o PCA	Model 2 – w/ PCA
Training Sample	Predictor(s) Selected	All stock log-returns	8 components
	Adjusted R2	0.8919	0.8886
	MSE	0.00000639	0.00000659
	MAE	0.0019020	0.0019752
Test Sample	MSE	0.00000931	0.00000968
	MAE	0.0021449	0.0021792

As you can see in Table 1, the adjusted R-squared, MSE, and MAE are slightly better in Model 1 than in Model 2 for both the training and test data sets. By comparing the residual and QQ plots from both models, they almost appear to be identical. Given we did not see any indications that multicollinearity was present among the predictor variables, this would make sense why we did not see any performance improvement from leveraging PCA.

Predictive Accuracy & Final Model Selection

Now that we have assessed the performance of the models in the statistical sense, we will now compare the models based on their predictive accuracy. By comparing the predicted values vs the actual values for both Model 1 and Model 2, we can have an objective basis for final model selection.

Table 2 - Model 1 Predictive Performance

Prediction_Grade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01: Grade 1 within 10%	20	12.20	20	12.20
02: Grade 2 within 10-20%	30	18.29	50	30.49
03: Grade 3 within 20-30%	18	10.98	68	41.46
04: Grade 4 within 30-40%	14	8.54	82	50.00
05: Grade 5 above 40%	82	50.00	164	100.00

Table 3 - Model 2 Predictive Performance

Prediction_Grade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01: Grade 1 within 10%	23	14.02	23	14.02
02: Grade 2 within 10-20%	23	14.02	46	28.05
03: Grade 3 within 20-30%	19	11.59	65	39.63
04: Grade 4 within 30-40%	15	9.15	80	48.78
05: Grade 5 above 40%	84	51.22	164	100.00

As we can see by comparing the frequency tables for the results of the predicted values against the actual values, there is not much of a difference between both models, with model 1 having a couple more predicted values within 40% of the actual values. Given we did not see any multicollinearity in the data, the logical model to use would be Model 1.

Conclusion

The final takeaway from this assignment is that there really is no benefit in performing PCA when multicollinearity does not exist among the predictor variables, at least from a predictive performance perspective. While PCA is good for dealing with multicollinearity, I can also see it being a very useful tool for identifying groupings of multiple predictors and for dimensionality reduction. The main thing I struggle with is that both of these models' predictive accuracy is far too low for me to have the confidence to recommend using either model. When we only see 50% of the predicted values coming within 40% of the actual values, I can't trust that these models are very accurate using linear regression. The next step would be to explore other machine learning techniques, such as support vector machines or naive Bayes, to see if there is any increase in predictive accuracy with the dataset we are working with in this assignment.

SAS Code Output

```
*-------------------------------------------------------------------------------------------
*  Nate Bitting
*  Assignment 6
*-------------------------------------------------------------------------------------------;

* Code used to get the data into my library;
ods graphics on;
libname mydata '/courses/d6fc9ae5ba27fe300/c_3505/SAS_Data/' access=readonly;
proc datasets library=mydata; run; quit;

data temp;
set mydata.stock_portfolio_data;
run;

proc sort data=temp; by date; run; quit;

data return_data;
set temp;

return_AA = log(AA/lag1(AA));
return_BAC = log(BAC/lag1(BAC));
return_BHI = log(BHI/lag1(BHI));
return_CVX = log(CVX/lag1(CVX));
return_DD = log(DD/lag1(DD));
return_DOW = log(DOW/lag1(DOW));
return_DPS = log(DPS/lag1(DPS));
return_GS = log(GS/lag1(GS));
return_HAL = log(HAL/lag1(HAL));
return_HES = log(HES/lag1(HES));
return_HON = log(HON/lag1(HON));
return_HUN = log(HUN/lag1(HUN));
return_JPM = log(JPM/lag1(JPM));
return_KO = log(KO/lag1(KO));
return_MMM = log(MMM/lag1(MMM));
return_MPC = log(MPC/lag1(MPC));
return_PEP = log(PEP/lag1(PEP));
return_SLB = log(SLB/lag1(SLB));
return_WFC = log(WFC/lag1(WFC));
return_XOM = log(XOM/lag1(XOM));
response_VV = log(VV/lag1(VV));
run;

*create a list of all the predictor variables;
%let xlist =return_AA return_BAC return_BHI return_CVX return_DD return_DOW return_DPS return_GS
                    return_HAL return_HES return_HON return_HUN return_JPM return_KO
                    return_MMM return_MPC return_PEP return_SLB return_WFC return_XOM;

proc print data=return_data(obs=10); run; quit;

ods output PearsonCorr=portfolio_correlations;
proc corr data=return_data;
var return_:;
with response_VV;
run; quit;

proc print data=portfolio_correlations; run; quit;

data wide_correlations;
set portfolio_correlations (keep=return_:);
run;

proc transpose data=wide_correlations out=long_correlations;
run; quit;

data long_correlations;
set long_correlations;
tkr = substr(_NAME_,8,3);
drop _NAME_;
rename COL1=correlation;
run;

proc print data=long_correlations; run; quit;

*print a scatter plot of a few variables with high correlation with the response;
proc sgscatter data=return_data;
title 'Scatter Plots of a few Predictors';
plot return_HON*return_MMM return_GS*return_XOM;
run;
```

```
data sector;
input tkr $ 1-3 sector $ 4-35;
datalines;
AA Industrial - Metals
BAC Banking
BHI Oil Field Services
CVX Oil Refining
DD Industrial - Chemical
DOW Industrial - Chemical
DPS Soft Drinks
GS Banking
HAL Oil Field Services
HES Oil Refining
HON Manufacturing
HUN Industrial - Chemical
JPM Banking
KO Soft Drinks
MMM Manufacturing
MPC Oil Refining
PEP Soft Drinks
SLB Oil Field Services
WFC Banking
XOM Oil Refining
VV Market Index
;
run;

proc print data=sector; run; quit;

proc sort data=sector; by tkr; run;

proc sort data=long_correlations; by tkr; run;

data long_correlations;
merge long_correlations (in=a) sector (in=b);
by tkr;
if (a=1) and (b=1);
run;

proc print data=long_correlations; run; quit;

ods graphics on;
title 'Correlations with the Market Index';
proc sgplot data=long_correlations;
format correlation 3.2;
vbar tkr / response=correlation group=sector groupdisplay=cluster
datalabel;
run; quit;
ods graphics off;


* create a single dataset that only includes the log return values for the predictor and repsonse variable;
data return_data_only;
set return_data;
drop AA;
drop BAC;
drop BHI;
drop CVX;
drop DD;
drop DOW;
drop DPS;
drop GS;
drop HAL;
drop HES;
drop HON
drop HUN;
drop JPM;
drop KO;
drop MMM;
drop MPC;
drop PEP;
drop SLB;
drop WFC;
drop XOM;
drop VV;
run;
```

```
ods graphics on;
proc princomp
data=return_data_only
out=pca_output
outstat=eigenvectors
plots=scree(unpackpanel);
var &xlist;
run; quit;
ods graphics off;

proc print data=pca_output(obs=10); run;

proc print data=eigenvectors(where=(_TYPE_='SCORE')); run;

data pca2;
set eigenvectors(where=(_NAME_ in ('Prin1','Prin2')));
drop _TYPE_ ;
run;

proc print data=pca2; run;

proc transpose data=pca2 out=long_pca; run; quit;
proc print data=long_pca; run;

data long_pca;
set long_pca;
format tkr $3.;
tkr = substr(_NAME_,8,3);
drop _NAME_;
run;

proc print data=long_pca; run;

* Plot the first two principal components;
ods graphics on;
proc sgplot data=long_pca;
scatter x=Prin1 y=Prin2 / datalabel=tkr;
run; quit;
ods graphics off;

*************************************************************************;
* Create a training data set and a testing data set from the
PCA output;
* Note that we will use a SAS shortcut to keep both of these
'datasets' in one data set that we will call cv_data (cross-validation
data). ;
*************************************************************************;
data cv_data;
merge pca_output return_data_only(keep=response_VV);
* No BY statement needed here. We are going to append a column in
its current order;
* generate a uniform(0,1) random variable with seed set to
123; u = uniform(123);
if (u < 0.70) then train = 1;
else train = 0;
if (train=1) then train_response=response_VV;
else train_response=.;
run;
proc print data=cv_data(obs=10); run;


*regression model 1 without PCA;
proc reg data=cv_data outest=RegOut;
model train_response = &xlist / vif mse;
*output out=residuals_final (keep = resid_final) r=resid_final;
run;

* calc the MAE for Model 1;
data abs_resid;
    set residuals_final;
    abs_resid = abs(resid_final);
run;

proc means data=abs_resid mean;
    var abs_resid;
run;

* calcualte the estimated values using the test dataset;
```

```
proc score data=cv_data score=RegOut out=RScoreP type=parms;
      var &xlist;
run;

proc score data=cv_data score=RegOut out=RScoreR type=parms;
      var train_response &xlist;
run;

* Output the scores using the model built with the training dataset against the test dataset;
proc score data=cv_data score=RegOut out=NewPred_M1 type=parms
              nostd predict;
      var train_response &xlist;
run;

* Smart data formats for predictions;
proc format;
value pred_acc_sfmt
    0 -< .1 = '01: Grade 1 within 10%'
    .1 -< .2 = '02: Grade 2 within 10-20%'
    .2 -< .3 = '03: Grade 3 within 20-30%'
    .3 -< .4 = '04: Grade 4 within 30-40%'
    other = '05: Grade 5 above 40%'
    ;
run;

*create a new dataset that contains the test set data and the predictive scores for Model 1;
data prediction_Data_M1;
    set NewPred_M1;
    if (train_response = null);
    pred_score = abs(response_VV / MODEL1 - 1);
    abs_resid = abs(response_VV - MODEL1);
    error_term = response_VV - MODEL1;
    sq_error = error_term**2;
    Prediction_Grade = put(pred_score,pred_acc_sfmt.);
run;

* calculate the MAE and MSE from the test sample;
proc means data=prediction_Data_M1 mean;
    var abs_resid sq_error;
run;

* create a frequency table to show the operational accuracy of model 1;
proc freq data=prediction_Data_M1;
    TABLES Prediction_Grade;
run;


*regression model 2 with 8 components;
proc reg data=cv_data outest=RegOutPCA1;
model train_response = Prin1 Prin2 Prin3 Prin4 Prin5 Prin6 Prin7 Prin8 / vif mse;
*output out=residuals_PCA1 (keep = resid_PCA1) r=resid_PCA1;
run;

* calc the MAE for Model 2;
data abs_resid_PCA1;
    set residuals_PCA1 ;
    abs_resid_PCA1 = abs(resid_PCA1);
run;

proc means data=abs_resid_PCA1 mean;
    var abs_resid_PCA1;
run;

* calcualte the estimated values using the test dataset;
proc score data=cv_data score=RegOutPCA1 out=RScoreP_M2 type=parms;
      var Prin1 Prin2 Prin3 Prin4 Prin5 Prin6 Prin7 Prin8;
run;

proc score data=cv_data score=RegOutPCA1 out=RScoreR_M2 type=parms;
      var train_response Prin1 Prin2 Prin3 Prin4 Prin5 Prin6 Prin7 Prin8;
run;

* Output the scores using the model built with the training dataset against the test dataset;
proc score data=cv_data score=RegOutPCA1 out=NewPred_M2 type=parms
              nostd predict;
      var train_response Prin1 Prin2 Prin3 Prin4 Prin5 Prin6 Prin7 Prin8;
run;
```

```
*create a new dataset that contains the test set data and the predictive scores for Model 2;
data prediction_Data_M2;
    set NewPred_M2;
    if (train_response = null);
    pred_score = abs(response_VV / MODEL1 - 1);
    abs_resid = abs(response_VV - MODEL1);
    error_term = response_VV - MODEL1;
    sq_error = error_term**2;
    Prediction_Grade = put(pred_score,pred_acc_sfmt.);
run;

* calculate the MAE and MSE from the test sample;
proc means data=prediction_Data_M2 mean;
    var abs_resid sq_error;
run;

* create a frequency table to show the operational accuracy of model 2;
proc freq data=prediction_Data_M2;
    TABLES Prediction_Grade;
run;

*regression model 3 with 3 components;
proc reg data=cv_data outest=RegOutPCA2;
model train_response = Prin1 Prin2 Prin3 / vif mse;
output out=residuals_PCA2 (keep = resid_PCA2) r=resid_PCA2;
run;

* calc the MAE for Model 3;
data abs_resid_PCA2;
    set residuals_PCA2 ;
    abs_resid_PCA2 = abs(resid_PCA2);
run;

proc means data=abs_resid_PCA2 mean;
    var abs_resid_PCA2;
run;

* calcualte the estimated values using the test dataset;
proc score data=cv_data score=RegOutPCA2 out=RScoreP_M3 type=parms;
      var Prin1 Prin2 Prin3;
run;

proc score data=cv_data score=RegOutPCA2 out=RScoreR_M3 type=parms;
      var train_response Prin1 Prin2 Prin3;
run;

* Output the scores using the model built with the training dataset against the test dataset;
proc score data=cv_data score=RegOutPCA2 out=NewPred_M3 type=parms
              nostd predict;
      var train_response Prin1 Prin2 Prin3;
run;


*create a new dataset that contains the test set data and the predictive scores for Model 3;
data prediction_Data_M3;
    set NewPred_M3;
    if (train_response = null);
    pred_score = abs(response_VV / MODEL1 - 1);
    abs_resid = abs(response_VV - MODEL1);
    error_term = response_VV - MODEL1;
    sq_error = error_term**2;
    Prediction_Grade = put(pred_score,pred_acc_sfmt.);
run;

* calculate the MAE and MSE from the test sample;
proc means data=prediction_Data_M3 mean;
    var abs_resid sq_error;
run;

* create a frequency table to show the operational accuracy of model 3;
proc freq data=prediction_Data_M3;
    TABLES Prediction_Grade;
run;
```