Abstract:

This project presents OLS, Lasso, CV Lassoand CV Elastic Net algorithms built on Historical values of the factor Variables to understand the pattern of their volatility with respect to time and attempts to interpret the source of the asset return of asset classes and attribute it to the factor returns. The data is obtained from Federal Reserve Economic Data. FRED is an online database consisting of hundreds of thousands of economic data time series from scores of national, international, public, and private sources.

Chapter 1: Introduction

The fundamental objective of any Asset allocation management is to minimize the risk as minimally as possible and maximize the expected returns to as high values as possible. We will need to design and formulate many separate portfolios and compare the portfolios among themselves to choose among them only those having optimal returns and minimum risks.(Wikipedia, Portfolio finance)

Again, we should clearly be able to differentiate between Long-term investments and short-term investments. We would usually like to retain long term investments for more than a year or two and in doing so our expectations is that our investment should increase in value substantially so that we can sell it later for sizeable profits.

We purchase short term investments to mitigate the risks so that they provide greater degree of protection for our long term investments. This is because volatility poses enormous threats and fluctuations in the market can cause our investments to lose values beyond recognition. Hence, thusfinancial specialists will need to judiciously study the various motivating forces inside the business which would impact making optimal choices in asset allocations.

Citing the example of Yes Bank shares, the stock price that crossed Rs.350 towards the end of July 2017 is valued at around Rs.10 currently. This has led to financial advisors to believe that effective asset allocation is more important to investment returns than individual stock selection. This would help minimizing investor losses especially when the market is bearish and it is more important to investment returns than individual stock selection.

A majority of investors believe that they can create a proper Asset allocation plan. But they held on to an investment too long and have fallen prey to Certainty Bias phenomena. Very few investors rebalance as often as they should. They tend to stick on to traditionally top performers which cause them to lose focus.

Factor investing is one amongst the most innovative investment approach which enables us to invest in asset classes based on factor variables that helps to understand clearly about differences in stock returns with or without factor investing. It helps in adjusting and reallocating our portfolios based on specific factors which would result in higher investment returns on long-term basis.(Wikipedia, Factor investing)

Chapter 2: Literature Review

In the Professors' Report, we showed that about 70% of all active returns on the overall fund can be explained by exposures to systematic factors.

The Professors' Report to the Norwegian Ministry of Finance lists four criteria for determiningWhat should be a factor. A factor should:

1. Be justified by academic research

2. Have exhibited significant premiums that are expected to persist in the future

3. Have return history available for bad times

4. Be implementable in liquid, traded instruments

(Andrew Ang et al., 2013)

Factor investing relies on a large body of empirical research on the determinants of expected returns, research that has so far mostly been conducted in the equity class but has been progressing in other classes. Indeed, the goal of these empirical studies is to identify characteristics, possibly including risk exposures that have an impact on average returns, so these attributes can then be used as criteria to sort securities into groups with different expected returns, and to construct long-only strategies with expected returns above the market.

(Lionel Martellini et al., 2020)

Factor-based investing seeks to achieve specific investment risk-and-return outcomes, greater transparency, increased control, and lower costs. When evaluating a factor-based investing framework, investors should consider not only their tolerance for active risk but the investment rationales supporting specific factors, the cyclicality of factor performance, and their own tolerance for these swings in performance.

(Scott N. Pappas, CFA; Joel M. Dickson, Ph.D.et al., 2015)

Chapter 3: Problem Statement

Systematic Risk are those risks which affect all companies within a market in one way or another i.e. World Equities, National Treasuries, Bond Risk Premium, Inflation Protection, Currency Protection to name a few.

Unsystematic Risk are company Specific Risks i.e. Strength of Management, Range of Products, Geographic Location, Financial Position and Innovational Factor to name a few.

We can diversify some of the portfolio risk away by investing in investments with different levels of risk measured by a company's Beta which is the measure of the average historic volatility of an Asset class return to the broader market risk. Modern Portfolio theory states that 95% of the unsystematic risk can be eliminated by simple mundane diversification towards investment of Asset classes.

However, we can't take our investment decisions by lesser useful diversification which may be based on few Individual factors. This is because randomly it may be possible that one or the other factor individually tend to perform well at different parts of the economic cycle. However, they will be less correlated with market moves of different Asset classes.

An investment's systematic risk is far more important than its unsystematic risk. Systematic sources of risks are common to most investments resulting in a perfect positive correlation and no diversification benefit. Large portfolios will not be affected by unsystematic risk but will be influenced by systematic risk factors. IN such a scenario, a multi-factor investment diversified across factors may help to reduce the effect of this cyclicality.

Chapter 4: Objectives of the Study

Enlarging our asset portfolios by adding these systematic risk factors along with already existing market risk factors is the single most important objective of factor investing. (Z´elia Cazalet et al., 2014)

We will be using statistical method to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. Simply put, the factor loading of a variable will quantify the extent to which the variable will be related with a given factor.

Factor Modeling aims at reducing the complexity of modeling asset price movements. For instance, trying to build a model that completely explains stock price movements is near impossible. In order to build a model for our favorite stock one would need to model supply, demand, sentiment, current and expected future earnings of the stock, news, interest rates, risk premia.

It's near impossible to calibrate a hugely complicated model. Instead, factor investors assume that there are N important factors that drive a portion of the asset returns. They then say that at the portfolio level, asset specific movements can be averaged out, and only those N variables remain. So to understand what drives the portfolio returns this project aims atmodeling the effect of that small number of factors.

Alternatively, understanding the factor loadings of the individual assets allows us to estimate the covariance of our returns. Factor loadings represent the hedging ratio which one would use to minimize the volatility of our portfolio.

This paper demonstrates the applicability of Machine Learning algorithms which includes OLS, Lasso, CV Lasso and CV Elastic Net algorithms built on Historical values of the factor Variables to understand the pattern of their volatility with respect to time and attempts to interpret the source of the asset return of asset classes and attribute it to the factor returns. The data is obtained from Federal Reserve Economic Data. FRED is an online database consisting of hundreds of thousands of economic data time series from scores of national, international, public, and private sources.

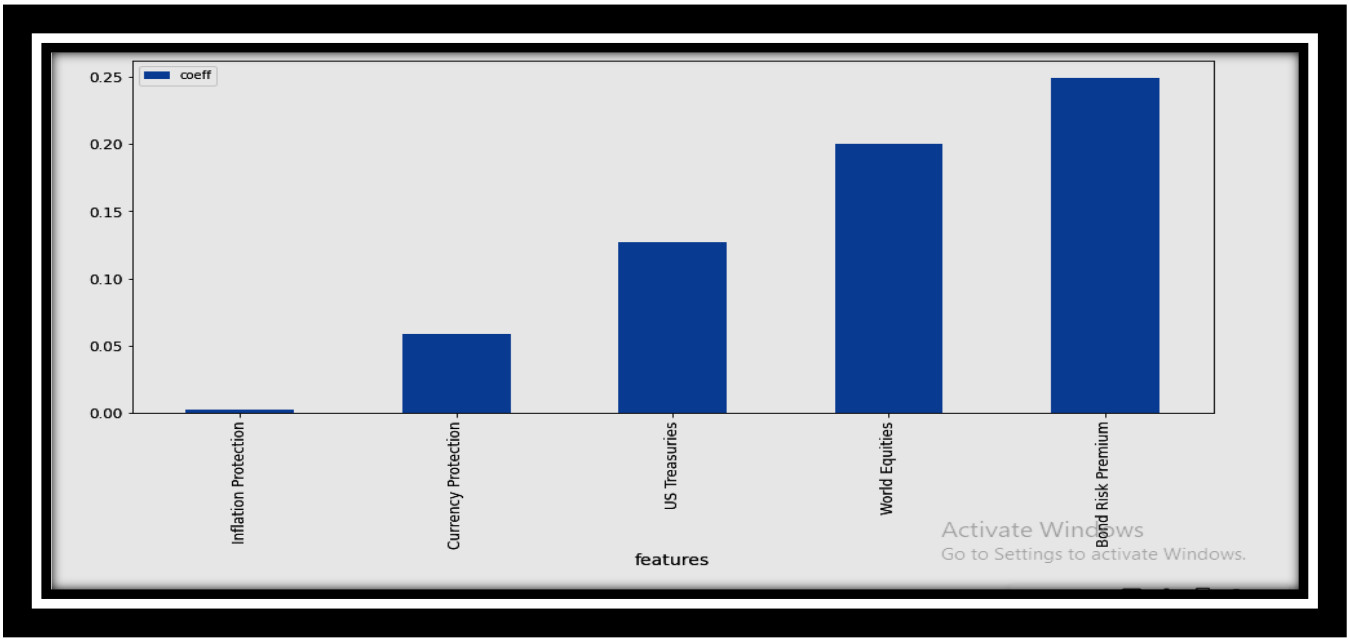

Figure. 4.1. Proposed Report. The Figure shown above is the proposed report of this project, which should benefit the investor to understand how the factor loadings are related with one amongst particular asset classes which we have taken up for investigations namely US Equities, Real Estate, Commodities and Corp Bonds.

Chapter 5: Project Methodology

The CRISP-DM framework has been used here for the project.

“Cross-industry standard process for data mining, known as CRISP-DM, is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model” (Wikipedia, 2020)

It comprises of the following six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. The project will be explained in these 6 phases in the following pages.

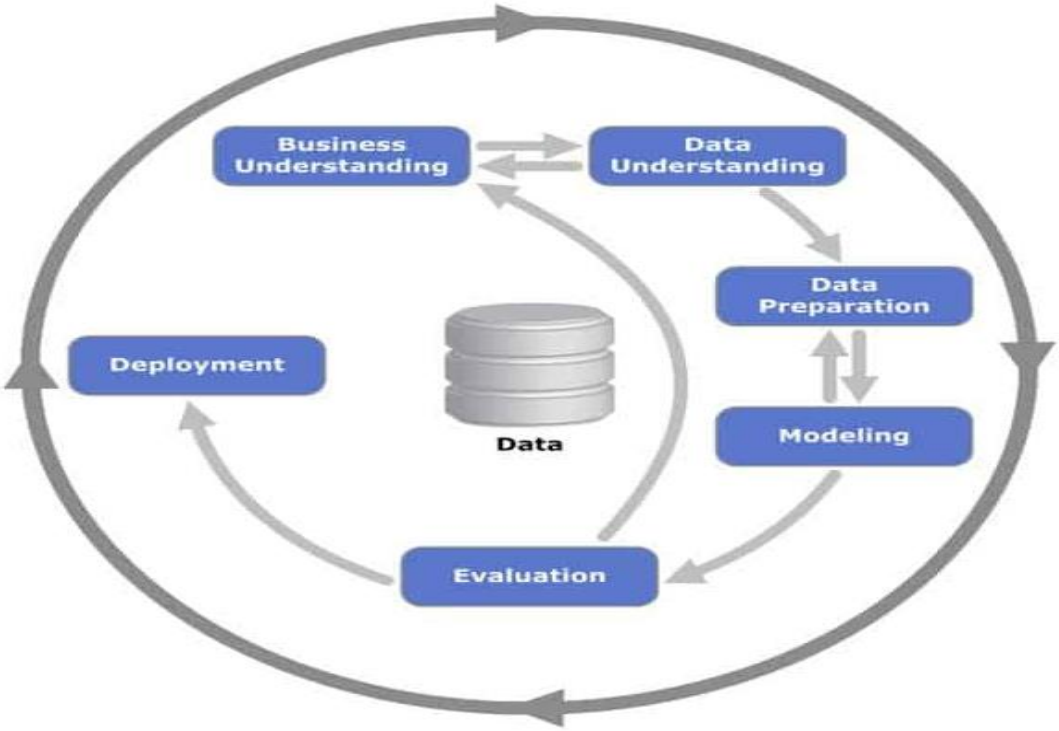

Figure 5.1 “CRISP-DM Process Diagram (Wikipedia, 26 April 2012)

Chapter 6: Business Understanding

Certain factors have historically earned a long-term risk premium and represent exposure to systematic sources of risk. Factor investing is the investment process that aims to harvest these risk premia through exposure to factors.

There are fewer Assumptions in factor analysis which must be taken care. There should be no outliers in data. The sample size of the data under observation should be considerably big. There should not be any multicollnearity between variables in factor analysis otherwise Factor modeling will not give us accurate predictions.There should be no homoscedasticity between the variables on which factor analysis is being done.Factor analysis can use non-linear variables but on transfer they should change to linear variables so that comparative analysis for regression coefficients is easier and simple.(Statswork, August 24, 2019)

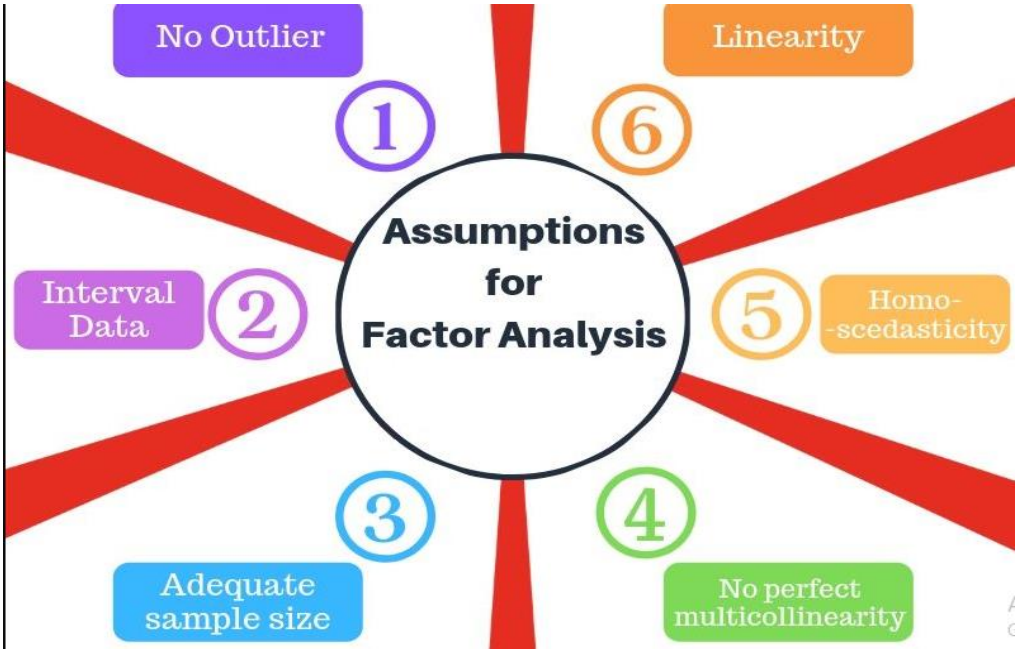

Figure 6.1 – Assumptions in factor analysis

Chapter 7: Data Understanding

We are interested in explaining the asset returns with a five-factor model:

1) World Equity: This factor represents worldwide equity returns.

2) US Treasury: This factor contains return from treasury bonds in United States, the bonds with the least risk.

3) Bond Risk Premia: This is a credit factor that captures extra yield from risky bonds. Defined as the spread between high risk bonds and US Treasury bonds.

4) Inflation Protection: This is a "style" factor that considers the difference between real and nominal returns, thus balances the need for both.

5) Currency Protection: This is also a "style" factor that includes risk premium for US domestic assets.

Our Asset classes for which we would build prediction models will include US Equities, Real Estate, Commodities and Corp Bonds.

	Date	World Equities	US Treasuries	Bond Risk Premium	Inflation Protection	Currency Protection	US Equities	Real Estate	Commodities	Corp Bonds
0	1/1/1985	0.028511	0.031500	0.006408	-0.016265	0.030292	0.081301	0.056605	0.021351	0.048963
1	2/1/1985	-0.009204	-0.044692	0.057381	0.006362	0.010258	0.030075	0.016448	-0.015217	-0.042029
2	3/1/1985	0.075134	0.028719	-0.024396	-0.002848	-0.020739	-0.007299	-0.006716	0.037171	0.032666
3	4/1/1985	-0.002459	0.023084	-0.004869	0.003089	0.008187	-0.012255	0.000906	-0.035116	0.037125
4	5/1/1985	0.040245	0.086780	-0.044417	0.004077	-0.002219	0.064516	0.027241	0.004351	0.104199
5	6/1/1985	0.022419	0.014284	-0.001572	0.027610	-0.017278	0.020979	0.023207	-0.046275	0.012482
6	7/1/1985	0.048480	-0.014958	0.023357	-0.010178	0.000007	0.004566	0.009221	-0.006240	-0.006449
7	8/1/1985	0.027709	0.021934	-0.001815	0.016312	0.011122	-0.009091	-0.054684	0.014064	0.037397
8	9/1/1985	0.048713	0.002172	0.007427	-0.004868	-0.034424	-0.027523	-0.031145	0.060092	0.000941
9	10/1/1985	0.058439	0.033903	-0.031340	0.026578	-0.007977	0.037736	0.031415	0.075345	0.037840
10	11/1/1985	0.044028	0.044784	-0.019856	0.018646	0.001869	0.054545	-0.016072	0.013692	0.039552
11	12/1/1985	0.043064	0.055787	-0.018211	0.002057	-0.000919	0.053879	0.006132	-0.020173	0.049738

Figure 7.1 – Historical Data for Asset classes and factor variables:

Chapter 8: Data Preparation

- Returns values on each date are extracted from the Historical return indexes of the 5 factor variables namely World Equity, US Treasury, Bond Risk Premia, Inflation Protection and Currency Protection.
- Returns values on each date are extracted from the 4 asset classes as well namely US Equities, Real Estate, Commodities and Corp Bonds.
- Then all data values is scaled between 0 and 1.This way we are able to convert unstandardized coefficients to standardized coefficients so as to limit the coefficient values between 0 and 1.
- The entire 33 Years data is Split into Train Data and Test Data.
- Then we will transform the Train Data and Test Data into a Supervised Learning Format.

To solve a given problem of supervised learning, one has to perform the following steps:

1. We should develop understanding on the type of training data on which we are trying to build the model. Data can be of various categories namely scalar and categorical data and we need to handle them differently as may be required.
2. We need to clearly know our input data which is also called as feature variables. Here we have all factor variables are our feature variables.
3. We have 4 asset classes which would be 4 different target variables and hence we will be building 4 different Models with each of the 4 different asset classes.
4. We need to be careful that we don't use large number of feature variables for building our models and at the same time we must have adequate data available to predict the output correctly.
5. We should choose the optimal modeling algorithm which would be best suited to build our model so that our model accuracy is higher.

Chapter 9: Data Modeling:

The preprocessed data discussed in the previous phase is fed in to the OLS Model. Initially we will build Linear Regression-OLS Model via Scikit-learn. Here we will keep one of the asset class namely US Equities as Target variable whereas all factor variables will be used as Feature variables.

STEP2 SPLIT TEST AND TRAIN

```
[ ] train_x,test_x,train_y,test_y=train_test_split(x_scld,y,test_size=0.20,random_state=1)
    train_x.shape
    test_x.shape
    train_y.shape
    test_y.shape
```

```
(324, 5)(81, 5)(324,)(81,)
```

STEP 3:CREATE INSTANT OF THE MODEL

```
[ ] lm=LinearRegression()
    lm
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

STEP 4:FIT THE MODEL

```
[ ] lm.fit(train_x,train_y)
```

Figure 9.1 – Linear Regression-OLS Model via Scikit-learn

OLS has major drawbacks. OLS has no mechanism to filter out noise variables. We must find out ways of filtering noise variables.

To demonstrate how OLS can be susceptible to noise, we will introduce a noise variable positively correlated with the World Equities factor. Then we will rerun the OLS regression. The OLS regression chooses to average the two signals, changing the loading on the World Equity factor.

LASSO regression, is the simplest version of regularized regression. Regularized regression means that we add a penalty term to the optimization problem to penalize the model's complexity. For the sake of intuition, imagine if we could penalize the number of non-zero coefficients we add to the model. In that instance we would expect that the model would only consider the variables that really influence Y, and ignore the noise variables. LASSO regression does exactly that.

```
train_x,test_x,train_y,test_y=train_test_split(x_scld,y,test_size=0.20,random_state=1)
```

```
from sklearn.linear_model import Lasso
lambda1 =0.1
```

```
lassoReg = Lasso(alpha=lambda1/(2*all_data.shape[0]), fit_intercept=True)
```

```
lassoReg.fit(train_x,train_y)
```

```
Lasso(alpha=0.0001234567901234568, copy_X=True, fit_intercept=True,
      max_iter=1000, normalize=False, positive=False, precompute=False,
      random_state=None, selection='cyclic', tol=0.0001, warm_start=False)
```

Figure 9.2 – Linear Regression-LASSO Model via Scikit-learn

It is the same as the OLS regression, but with a second penalty term. As this directly penalizes the use of non-zerocoefficients and can set factor loadings to zero.

Next we will further try optimizing our Modeling by using LASSO regression with cross validation. We will break the training set into k folds, and define a list of λ values. For each fold, and for each λ, wewill train the model k−1 other folds, and calculate the error on the test fold.

At the end of this, we will have k out of sample errors for each value of lambda. Then pick the λ which satisfies the average error across our out of sample tests. Hence, by using cross validation we will be able to pick the optimal lambda value for lasso.

```
alphaMax = maxLambda1 / (2*all_data.shape[0])
alphas = np.linspace(1e-6, alphaMax,nLambdas)
lassoTest = Lasso(random_state = randomState, fit_intercept=True)
tuned_parameters = [{'alpha': alphas}]
clf = GridSearchCV(lassoTest, tuned_parameters, cv=nFolds, refit=True)
```

```
clf.fit(train_x,train_y)
lassoBest = clf.best_estimator_
alphaBest = clf.best_params_['alpha']
```

```
GridSearchCV(cv=10, error_score=nan,
             estimator=Lasso(alpha=1.0, copy_X=True, fit_intercept=True,
                             max_iter=1000, normalize=False, positive=False,
                             precompute=False, random_state=7777,
                             selection='cyclic', tol=0.0001, warm_start=False),
             iid='deprecated', n_jobs=None,
             param_grid=[{'alpha': array([1.00000000e-06, 4.10749470e-06, 7.21498940e-06, 1.03224841e-05,
       1.34299788e-0...
       2.49599576e-04, 2.52707071e-04, 2.55814565e-04, 2.58922060e-04,
       2.62029555e-04, 2.65137050e-04, 2.68244544e-04, 2.71352039e-04,
       2.74459534e-04, 2.77567028e-04, 2.80674523e-04, 2.83782018e-04,
       2.86889512e-04, 2.89997007e-04, 2.93104502e-04, 2.96211997e-04,
       2.99319491e-04, 3.02426986e-04, 3.05534481e-04, 3.08641975e-04])}],
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
lassoBest
```

```
Lasso(alpha=5.0719915201396676e-05, copy_X=True, fit_intercept=True,
      max_iter=1000, normalize=False, positive=False, precompute=False,
      random_state=7777, selection='cyclic', tol=0.0001, warm_start=False)
```

```
alphaBest
```

```
5.0719915201396676e-05
```

Figure 9.3 – Linear Regression-using LASSO regression with cross validation

Now that we've discussed cross validation and LASSO regression, we can mix and match penalized regressions to create regressions with specific properties. For instance, we know from literature that LASSO regression can be used for variable selection. We also know that Ridge regression shrinkscoefficients to provide a more robust solution. Combined, it's called an Elastic Net, and it can provide the benefits of both LASSO regression and Ridge regression. It is being observed that elastic net does better than LASSO when we have many highly correlated variables.

```
maxLambda= .25
maxL1Ratio = .99
nLambdas = 100
nL1Ratios = 100
randomState = 7777
nFolds = 10
from sklearn.linear_model import Ridge
from sklearn.linear_model import ElasticNet
```

```
alphaMax = maxLambda/(2*all_data.shape[0])
alphas = np.linspace(1e-6, alphaMax,nLambdas)
l1RatioMax = maxL1Ratio
l1Ratios = np.linspace(1e-6, l1RatioMax,nL1Ratios)

elasticNetTest = ElasticNet(random_state = randomState, fit_intercept=True)

tuned_parameters = [{'alpha': alphas, 'l1_ratio': l1Ratios}]

clf = GridSearchCV(elasticNetTest, tuned_parameters, cv=nFolds, refit=True)
```

```
clf.fit(train_x,train_y)
lassoBest = clf.best_estimator_
alphaBest = clf.best_params_['alpha']
elasticNetBest = clf.best_estimator_
alphaBest = clf.best_params_['alpha']
l1RatioBest = clf.best_params_['l1_ratio']
```

```
GridSearchCV(cv=10, error_score=nan,
             estimator=ElasticNet(alpha=1.0, copy_X=True, fit_intercept=True,
                                  l1_ratio=0.5, max_iter=1000, normalize=False,
                                  positive=False, precompute=False,
                                  random_state=7777, selection='cyclic',
                                  tol=0.0001, warm_start=False),
             iid='deprecated', n_jobs=None,
             param_grid=[{'alpha': array([1.00000000e-06, 4.10749470e-06, 7.21498940e-06, 1.03224841...
       8.00000192e-01, 8.10000182e-01, 8.20000172e-01, 8.30000162e-01,
       8.40000152e-01, 8.50000141e-01, 8.60000131e-01, 8.70000121e-01,
       8.80000111e-01, 8.90000101e-01, 9.00000091e-01, 9.10000081e-01,
       9.20000071e-01, 9.30000061e-01, 9.40000051e-01, 9.50000040e-01,
       9.60000030e-01, 9.70000020e-01, 9.80000010e-01, 9.90000000e-01])}],
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
print('best lambda1 = ' + str(alphaBest*2*all_data.shape[0]*l1RatioBest))
print('best lambda2 = ' + str(all_data.shape[0]*alphaBest*(1-l1RatioBest)))
```

```
best lambda1 = 0.041420193680816245
best lambda2 = 0.001090041696928885
```

Figure 9.4 – Linear Regression-using Elastic Net with cross validation

Chapter 10: Data Evaluation

Once the Model is built, the scaled output data of the model has to be retransformed.

5 factor variables are input variables namely World Equity, US Treasury, Bond Risk Premia, Inflation Protection and Currency Protection.

Target variable is taken as US Equities.

Now the model should be evaluated for all the return values of US Equities.

We will predict the values of US equities Returns and Compare the predicted Returns plot with the actual Returns plot. Similar such plots can be computed separately for other Asset classes namely Real Estate, Commodities and Corp Bonds.

Based on OLS Model following is the heat map plot to demonstrate correlation between US Equities Returns and other five factor variables Returns namely World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection.

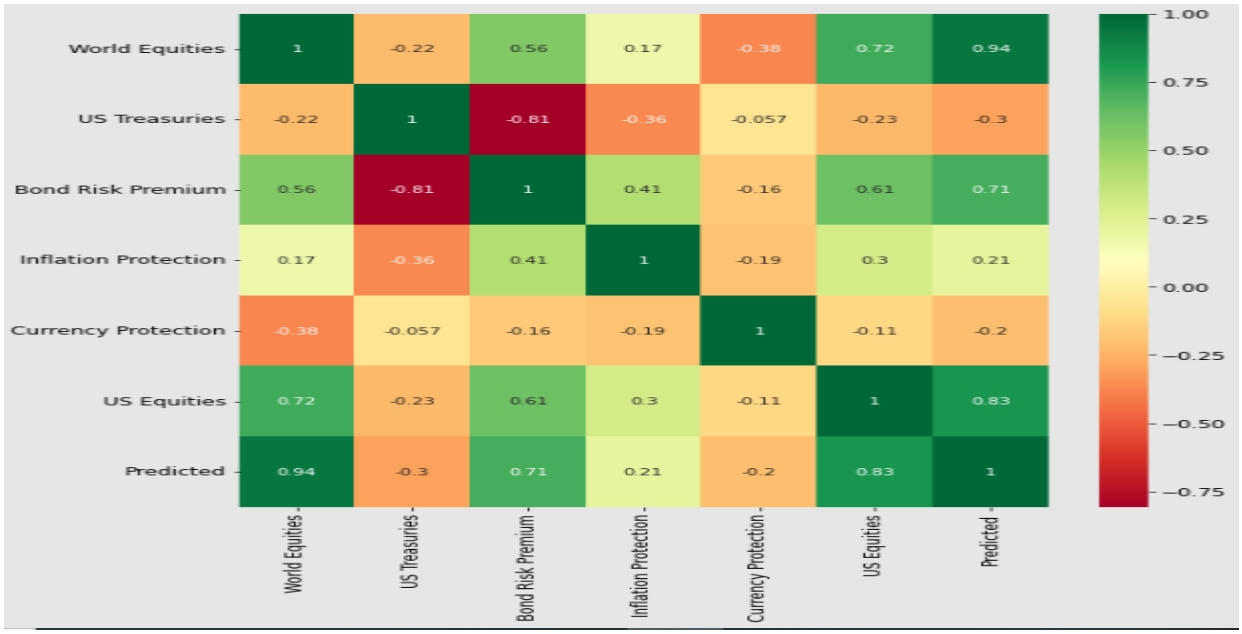

Figure 10.1– Heat map plot to demonstrate correlation between US Equities Returns and other five factor variables

The Insight drawn would be that World Equities and Bond Risk Premium are highly positively correlated whereas other factor variables don't seem to have significant impact on US Equities Returns.

We also observe that Correlation between Actual US Equities Returns and Predicted returns is 0.83.

Based on CV elastic Net following is the heat map plot to demonstrate correlation between Real Estate Returns and other five factor variables Returns namely World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection.

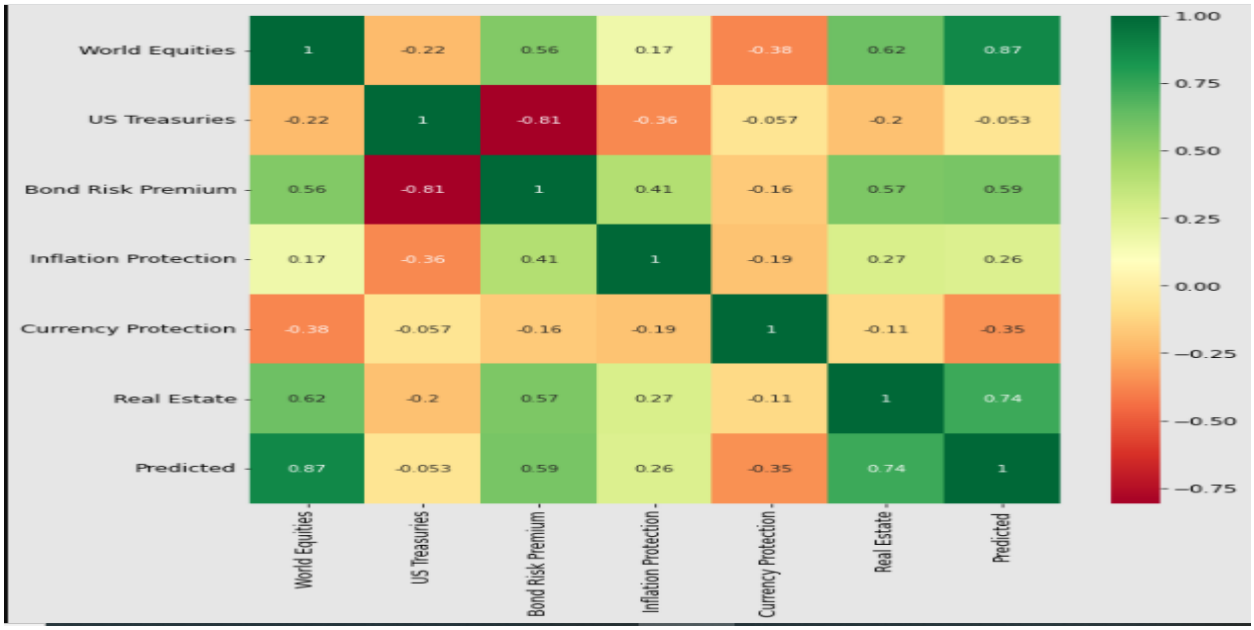

Figure 10.2– Heat map plot to demonstrate correlation between Real Estate Returns and other five factor variables Returns

The Insight drawn would be that World Equities and Bond Risk Premium are highly positively correlated whereas other factor variables don't seem to have significant impact on Real Estate Returns.

We also observe that Correlation between Actual Real Estate Returns and Predicted returns is 0.74.

Based on CV elastic Net following is the heat map plot to demonstrate correlation between Commodities Returns and other five factor variables Returns namely World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection.

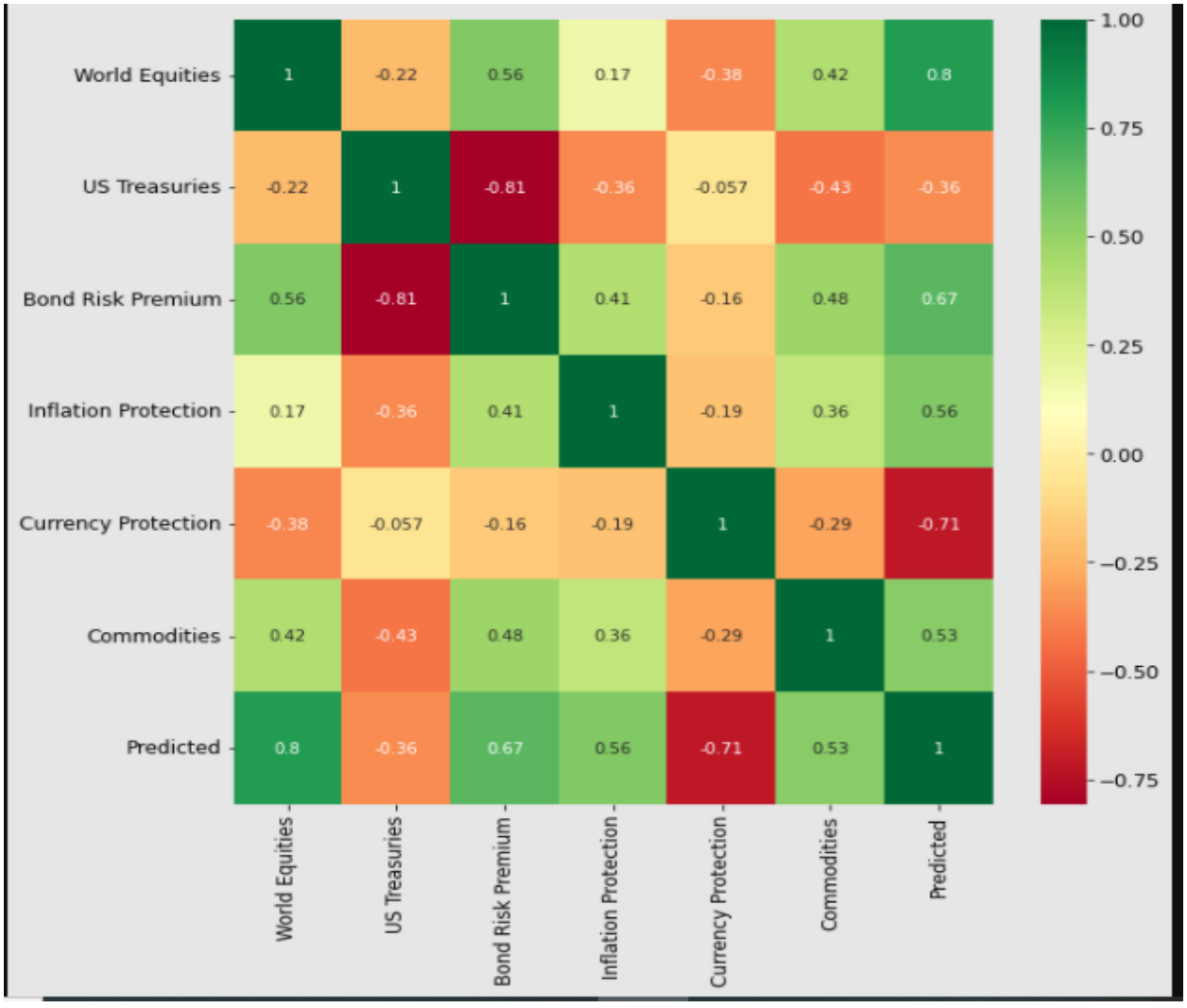

Figure 10.3– Heat map plot to demonstrate correlation between Commodities Returns and other five factor variables Returns

The Insight drawn would be that World Equities and Bond Risk Premium are positively correlated whereas US Treasuries is negatively correlated with Commodities Returns.

We also observe that Correlation between Actual Commodities Returns and Predicted returns is 0.53 which is relatively lesser.

Based on CV elastic Net following is the heat map plot to demonstrate correlation between Corp Bonds Returns and other five factor variables Returns namely World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection.

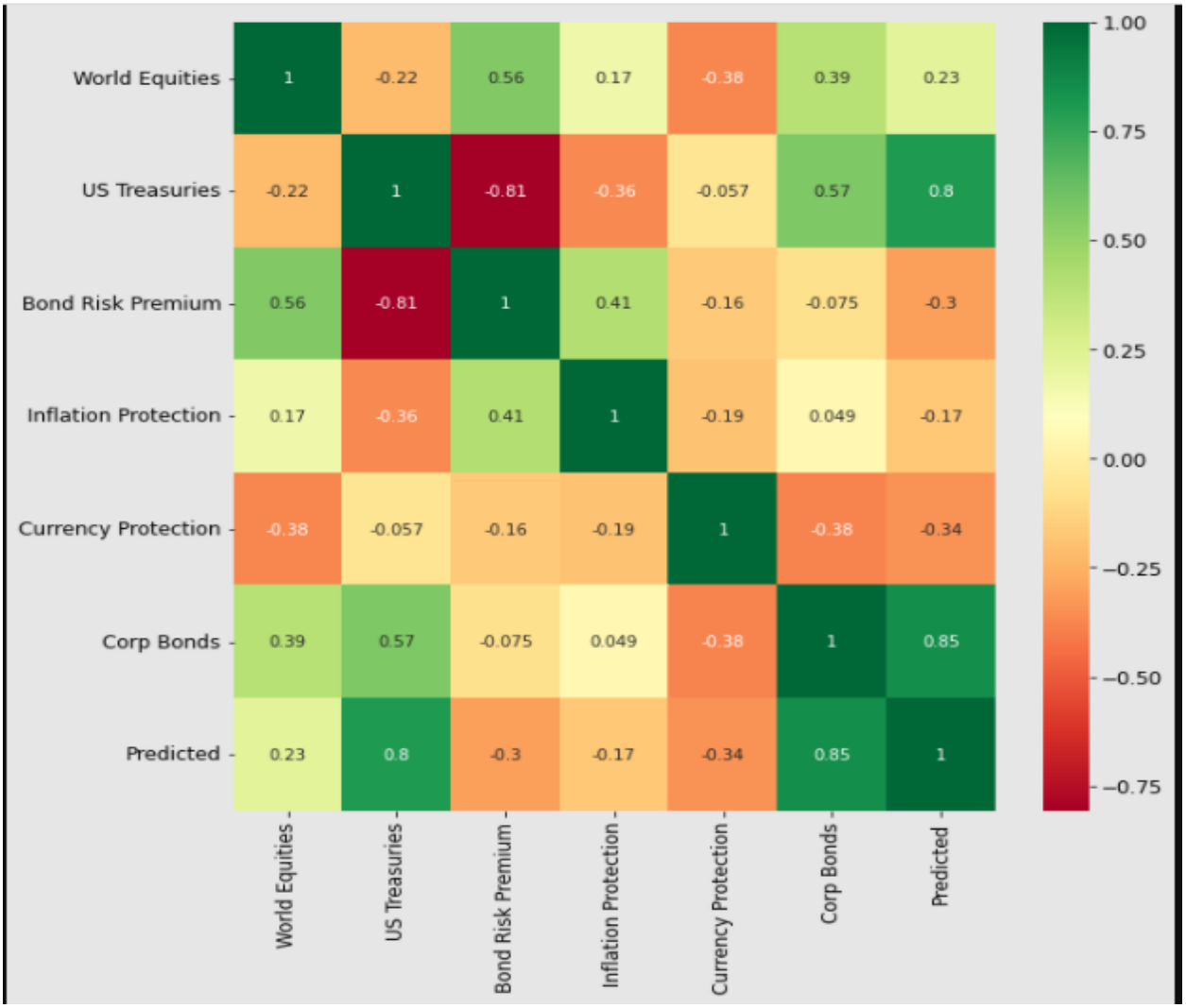

Figure 10.4– Heat map plot to demonstrate correlation between Corp Bonds Returns and other five factor variables Returns

The Insight drawn would be that World Equities and US Treasuries are positively correlated and Currency Protection is negatively correlated whereas other factor variables don't seem to have significant impact on Corp Bonds Returns.

We also observe that Correlation between Actual Commodities Returns and Predicted returns is 0.85.

Chapter 11: Deployment

Major Action Items Implemented:

US Equities as Target Variable from Asset Class Category:

I have used World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection as Factor Variables.

I have used Asset class namely US Equities as Target Variable to predict the Returns based on the Factor Variables as Independent variables.

I have employed Linear Regression-OLS Model, Lasso Regression, CV Lasso and CV Elastic Net Modeling techniques to predict US Equities Returns.

My Leader Board gives me the following results:

	MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA	MEAN SQUARE ERROR (MSE) FOR TEST DATA	MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA	R-SQUARE VALUE FOR TEST DATA	R-SQUARE VALUE FOR TRAIN DATA
Regression-OLS Model	0.02039	0.00071	0.01331	68	61
Lasso Regression	0.021	0.00077	0.01531	66	60
CV Lasso	0.021	0.00073	0.01379	67	60
CV Elastic Net	0.021	0.00074	0.01383	67	60

Figure 11.1– Leader Board-comparison of Metrics for Predictions on US EQUITIES against Factor Variables(World Equities, US Treasuries, Bond Risk Premium, Inflation Protection, Currency Protection)by different prediction Models

Clearly it can be observed that mean Absolute Error, Mean Square error, Median Absolute Error is minimum in case of OLS Model. So we can use OLS Model to predict US Equities Returns against five factor variables used as Independent variables.

Real Estate as Target Variable from Asset Class Category:

I have used World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection as Factor Variables.

I have used Asset class namely Real Estate as Target Variable to predict the Returns based on the Factor Variables as Independent variables.

I have employed Linear Regression-OLS Model, Lasso Regression, CV Lasso and CV Elastic Net Modeling techniques to predict Real Estate Returns.

My Leader Board gives me the following results:

	MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA	MEAN SQUARE ERROR (MSE) FOR TEST DATA	MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA	R-SQUARE VALUE FOR TEST DATA	R-SQUARE VALUE FOR TRAIN DATA
Linear Regression-OLS Model	0.03466	0.00363	0.02454	44	38
Lasso Regression	0.035	0.0039	0.02454	40	37
CV Lasso	0.035	0.00387	0.02466	41	38
CV Elastic Net	0.035	0.00386	0.02233	41	38

Figure 11.2– Leader Board-comparison of Metrics for Predictions on Real Estate against Factor Variables(World Equities, US Treasuries, Bond Risk Premium, Inflation Protection, Currency Protection)by different prediction Models

Clearly it can be observed that mean Absolute Error, Mean Square error is better in case of OLS Model Whereas CV elastic Net gives better Median Absolute error. Mean Absolute Error and Mean Square error also works fine here. so we can use either OLS Model or CV elastic Net to predict Real Estate Returns against five factor variables used as Independent variables

Commodities as Target Variable from Asset Class Category:

I have used World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection as Factor Variables.

I have used Asset class namely Commodities as Target Variable to predict the Returns based on the Factor Variables as Independent variables.

I have employed Linear Regression-OLS Model, Lasso Regression, CV Lasso and CV Elastic Net Modeling techniques to predict Commodities Returns.

My Leader Board gives me the following results:

	MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA	MEAN SQUARE ERROR (MSE) FOR TEST DATA	MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA	R-SQUARE VALUE FOR TEST DATA	R-SQUARE VALUE FOR TRAIN DATA
Linear Regression-OLS Model	0.03775	0.00239	0.02633	28.0	17.0
Lasso Regression	0.038	0.00241	0.0261	28	16
CV Lasso	0.038	0.0024	0.02617	28	17
CV Elastic Net	0.038	0.0024	0.02629	28	17

Figure 11.3– Leader Board-comparison of Metrics for Predictions on Commodities against Factor Variables(World Equities, US Treasuries, Bond Risk Premium, Inflation Protection, Currency Protection)by different prediction Models

Clearly it can be observed that mean Absolute Error, Mean Square error, Median Absolute Error is almost the same for all Modeling algorithms.

CORP BONDS AS TARGET VARIABLE FROM ASSET CLASS CATEGORY:

I have used World Equities, US Treasuries, Bond Risk Premium, Inflation Protection and Currency Protection as Factor Variables. I have used Asset class namely Corp Bonds as Target Variable to predict the Returns based on the Factor Variables as Independent variables.

I have employed Linear Regression-OLS Model, Lasso Regression, CV Lasso and CV Elastic Net Modeling techniques to predict Corp Bonds Returns.

My Leader Board gives me the following results:

	MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA	MEAN SQUARE ERROR (MSE) FOR TEST DATA	MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA	R-SQUARE VALUE FOR TEST DATA	R-SQUARE VALUE FOR TRAIN DATA
Linear Regression-OLS Model	0.00741	0.00014	0.00501	71.0	77.0
Lasso Regression	0.008	0.0002	0.00509	57.0	72.0
CV Lasso	0.007	0.00014	0.00487	71.0	77.0
CV Elastic Net	0.007	0.00014	0.00487	71.0	77.0

Figure 11.4– Leader Board-comparison of Metrics for Predictions on Corp Bonds against Factor Variables(World Equities, US Treasuries, Bond Risk Premium, Inflation Protection, Currency Protection)by different prediction Models

Clearly it can be observed that mean Absolute Error, Mean Square error, Median Absolute Error is minimum in case of CV elastic Net. So we can use CV elastic Net Model to predict Corp Bonds Returns against five factor variables used as Independent variables.

Chapter 12: Analysis and Results

5 factor variables are input variables namely World Equity, US Treasury, Bond Risk Premia, Inflation Protection and Currency Protection. Target variable is taken as US Equities.

Now the model should be evaluated for all the predicted return values of US Equities vs. Actual values. Similar evaluations can be done by building separate models for other Target variables namely Real Estate, Commodities and Corp Bonds.

In the context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group.

The Mean Squared Error (MSE) is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model's predictions and the ground truth, square it, and average it out across the whole dataset.

The median absolute error is robust to outliers. The loss is calculated by taking the median of all absolute differences between the target and the prediction.

R-squared is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted. The value of R-square can also be negative when the models fitted are worse than the average fitted model.R-square values are computed for both test data and train data separately.

Model performance is being evaluated on the basis of above discussed metrics for the different Models built for our project.

```
from sklearn.metrics import mean_absolute_error
print("MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(mean_absolute_error(predicted_values,test_y),5)
```

```
MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.02039
```

```
from sklearn import metrics
print("MEAN SQUARE ERROR (MSE) FOR TEST DATA IS")
np.round(metrics.mean_squared_error(test_y,predicted_values),5)
```

```
MEAN SQUARE ERROR (MSE) FOR TEST DATA IS
0.00071
```

```
from sklearn.metrics import median_absolute_error
print("MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(median_absolute_error(predicted_values,test_y),5)
```

```
MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.01331
```

```
"R-SQUARE VALUE FOR TEST DATA IS"
np.round(lm.score(test_x,test_y)*100,0)
```

```
'R-SQUARE VALUE FOR TEST DATA IS'
68.0
```

```
"R-SQUARE VALUE FOR TRAIN DATA IS"
np.round(lm.score(train_x,train_y)*100,0)
```

```
'R-SQUARE VALUE FOR TRAIN DATA IS'
61.0
```

Figure 12.1– Model Performance Evaluation for Linear Regression-OLS Model

```
#STEP6-EVALUATE MODEL PERFORMANCE
from sklearn.metrics import mean_absolute_error
print("MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(mean_absolute_error(predicted_values,test_y),3)
```

```
MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.021
```

```
from sklearn import metrics
print("MEAN SQUARE ERROR (MSE) FOR TEST DATA IS")
np.round(metrics.mean_squared_error(test_y,predicted_values),5)
```

```
MEAN SQUARE ERROR (MSE) FOR TEST DATA IS
0.00077
```

```
from sklearn.metrics import median_absolute_error
print("MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(median_absolute_error(predicted_values,test_y),5)
```

```
MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.01531
```

```
"R-SQUARE VALUE FOR TEST DATA IS"
np.round(lassoReg.score(test_x,test_y)*100,0)
```

```
'R-SQUARE VALUE FOR TEST DATA IS'
66.0
```

```
"R-SQUARE VALUE FOR TRAIN DATA IS"
np.round(lassoReg.score(train_x,train_y)*100,0)
```

```
'R-SQUARE VALUE FOR TRAIN DATA IS'
60.0
```

Figure 12.2– Model Performance Evaluation for LASS0 Regression

```
from sklearn.metrics import mean_absolute_error
print("MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(mean_absolute_error(predicted_values,test_y),3)
```

```
MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.021
```

```
from sklearn import metrics
print("MEAN SQUARE ERROR (MSE) FOR TEST DATA IS")
np.round(metrics.mean_squared_error(test_y,predicted_values),5)
```

```
MEAN SQUARE ERROR (MSE) FOR TEST DATA IS
0.00073
```

```
from sklearn.metrics import median_absolute_error
print("MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(median_absolute_error(predicted_values,test_y),5)
```

```
MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.01379
```

```
"R-SQUARE VALUE FOR TEST DATA IS"
np.round(clf.score(test_x,test_y)*100,0)
```

```
'R-SQUARE VALUE FOR TEST DATA IS'
67.0
```

```
"R-SQUARE VALUE FOR TRAIN DATA IS"
np.round(clf.score(train_x,train_y)*100,0)
```

```
'R-SQUARE VALUE FOR TRAIN DATA IS'
60.0
```

Figure 12.3– Model Performance Evaluation for CV LASS0 Regression

```
from sklearn.metrics import mean_absolute_error
print("MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(mean_absolute_error(predicted_values,test_y),3)
```

```
MEAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.021
```

```
from sklearn import metrics
print("MEAN SQUARE ERROR (MSE) FOR TEST DATA IS")
np.round(metrics.mean_squared_error(test_y,predicted_values),5)
```

```
MEAN SQUARE ERROR (MSE) FOR TEST DATA IS
0.00074
```

```
from sklearn.metrics import median_absolute_error
print("MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS")
round(median_absolute_error(predicted_values,test_y),5)
```

```
MEDIAN ABSOLUTE ERROR (MAE) FOR TEST DATA IS
0.01383
```

```
"R-SQUARE VALUE FOR TEST DATA IS"
np.round(clf.score(test_x,test_y)*100,0)
```

```
'R-SQUARE VALUE FOR TEST DATA IS'
67.0
```

```
"R-SQUARE VALUE FOR TRAIN DATA IS"
np.round(clf.score(train_x,train_y)*100,0)
```

```
'R-SQUARE VALUE FOR TRAIN DATA IS'
60.0
```

Figure 12.4– Model Performance Evaluation forCV Elastic Net

Chapter 13: Conclusions and Recommendations for future work

This project solely focus on the Factor analysis of the Asset classes using OLS, Lasso, CV Lasso and CV Elastic Net algorithms where the model studies the return values from 5 different factor variables namely World Equity, US Treasury, Bond Risk Premia, Inflation Protection and Currency Protection and predicts the return values of asset class namely US Equities. Similarseparate models can be built for other asset classes namely Real Estate, Commodities and Corp Bonds Based on which the portfolio management with respect to asset classes need to be reviewed and refined accordingly.

Recommendations for Future Work: we assumed that factor loadings are constant over time. However, the assumption that the factor loadings are constant over time is restrictive, and not true. In fact, factor loadings are highly dependent on the time period.

We haven't discussed how to address one major drawback of factor analysis, namely that over different time periods the factor loadings can change.

In the future projects we can show how to define regimes using modern machine learning techniques. We will then use the techniques already discussed in this project to estimate the factor loadings for each of Normal and Crash regimes.

Bibliography

Factor Investing-This version: 06-10-2013 Andrew Ang

Facts and Fantasies About Factor Investing

Z´elia Cazalet -Quantitative Research, Lyxor Asset Management, Paris, zelia.cazalet@lyxor.com

Thierry Roncalli -Quantitative Research, Lyxor Asset Management, Paris thierry.roncalli@lyxor.com, October 2014

The Characteristics of Factor Investing

David Blitz and Milan Vidojevic

Factor Investing in Liability-Driven and Goal-Based Investment Solutions

Lionel Martellini &Vincent Milhau,March 2020

Factor-based investing-Vanguard Research

Scott N. Pappas, CFA; Joel M. Dickson, Ph.D.,April 2015

Flexible Indeterminate Factor-Based Asset Allocation. Portfolio Management 42, no. 5: 79–93. doi:10.3905/jpm.2016.42.5.079.

Blyth, Stephen, Mark C Szigety, and Jake Xia. 2016.

Smart Beta and Beyond: Maximising the Benefits of Factor Investing

An EDHEC-Risk Institute Publication,February 2018

The Education of Beta: Can Alternative Indexes Make Your Portfolio Smarter?

EUGENE PODKAMINER