

**PORTFOLIO THEORY AND ASSET PRICING**

Submission Number: 1

Group Number: 3 - A

Group Members:

Mantobaye Moundigbaye [mantobaye@yahoo.fr](mailto:mantobaye@yahoo.fr)

Louis Regnier-Vigouroux [louisregnierv@gmail.com](mailto:louisregnierv@gmail.com)

Yonas Menghis Berhe [Yonix500@gmail.com](mailto:Yonix500@gmail.com)

Summary of Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Untrimmed | | Trimmed | |
|  | Score | MSE | Score | MSE |
| Lasso | 0.8277 | 1.5526e-05 |  |  |
| PCA correlation matrix | 0.9809 | 4.8912e-07 | 0.9793 | 0.0207 |
| PCA covariance matrix | 0.9934 | 4.5197e-07 | 0.9839 | 4.1199e-07 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Untrimmed | | Trimmed | |
|  | Score | Accur / Depth | Score | Accur / Depth |
| Regression Tree | 0.8057 | -0.5 / 7 | 0.78315 | -0.5 / 7 |
| SVM | 0.9915 |  | 0.9718 |  |

8.1) Out of all the models we run, the SVM model gives the best explanatory results. It has the highest score whether the data is trimmed or not. Moreover, it is robust against outliers and uses fewer parameters to model the outcome variable.

8.2) Other extreme value techniques:

The method chosen to address extreme values is the trimming of the data on both ends to remove outliers. There exist other techniques to handle this. Among these are:

1. *Transforming values*: Scaling, Log transformation, Cube Root Normalization, …
2. *Imputation*: using mean, median, zero values. Imputing prevents loss of data.
3. Treating extreme values separately in the statistical model.

**9.1) Report to Trading Manager:**

The technical solution presented aims at finding the model that explains best the fluctuations of the equity fund returns over time. It uses data from 11 SPDR ETFs as explanatory variables and applies different models. Across all these models, Support Vector Machine (SVM) showed the best performance.

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform the data and then uses it to find an optimal decision boundary between the possible outputs.

Simply put, it does some complex data transformations with different kernel functions to transform the raw data from low dimension to high dimension so that it becomes easier to separate the data using a hyperplane. It then figures out how to separate your data based on the labels or outputs already defined.

In our case, we have some continuous data and are dealing with a regression-type rather than a classification-type problem. According to this information we decided to apply support vector regressions SVR which is the regression version of SVM. SVR uses the same principle as the SVM. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of data points. Here the regression is not trying to minimize the error between the original and predicted values but instead it is trying to fit the line with the threshold level which is the distance between the hyperplane and decision boundary line.

In order to implement the SVR we first split the data randomly into training and testing data sets. Then, feature scaling is carried out to make sure that all the variable have the same magnitude on the outcome variable and normalize the data within a particular range.

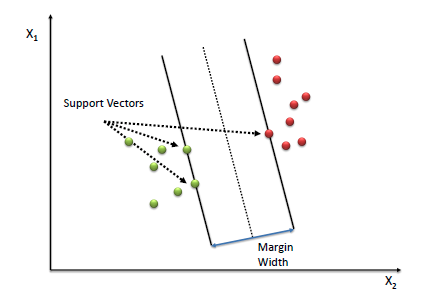
Next, we select our desired parameters, that is using the equity fund feature as an outcome variable and the all the SPDRs as explanatory variable designates as Y and X respectively in our code.

We then fit the model over the training data using the LinearSVR. This function provides a faster implementation than normal SVR but only considers the linear kernel. It is appropriate for large continues data sets.

Once the model is fit over the training data, we used the score evaluation metric to see how the model performed. This evaluation shows how well the model fit the data and it helps in comparing it with other models. As we can see the SVR shows highest performance both in the trimmed and untrimmed data.

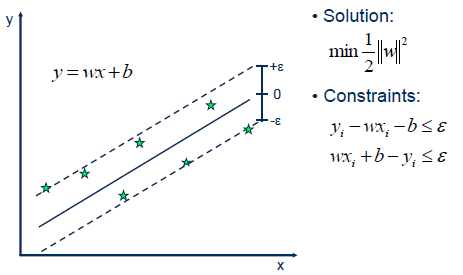
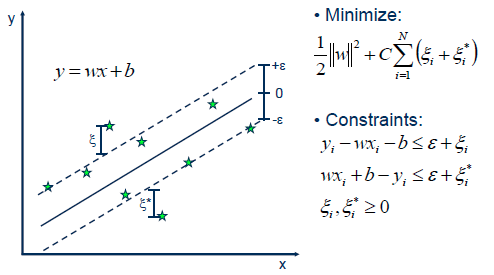
**9.2) Full Documentation**

Support Vector Machines or SVM is a supervised machine learning algorithms that performs classification by finding the most hyperplane that maximizes the margin between the two classes. SVMs define the maximum margin of the hyper plane by mapping data to high dimensional space using a variety of kernel function where it is easier to classify with linear decision surfaces. The vectors that define the hyperplane are the support vectors.

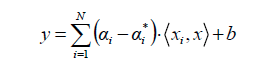


Support Vector Regressions or SVR are one class of SVM which gives an output of real numbers and defines the acceptable level of error in our model to find the appropriate line or hyperplane to fit the data.

where yᵢ is the target, wᵢ is the coefficient, and xᵢ is the predictor feature and b is constant. For any value that falls outside of ϵ, we can denote its deviation from the margin as ξ. An additional hyperparameter C can be tuned to give us the lowest Mean Absolute Error (MAE) by doing a grid search.

SVR minimizes the coefficient vector, not the squared error. The error term is handled in the constraints where we set less than or equal to a specified margin, called the maximum error, ϵ (epsilon). Then the epsilon can be tuned to obtain the desired accuracy of our model. The type of SVR we applied to our data set is called the Linear SVR and its expressed as follows.

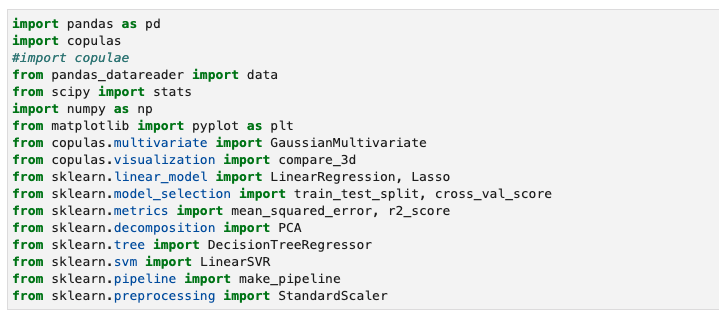


**Model Implementation**

The implementation is carried out with Python and Jupyter Notebooks. To actually implement the support vector regression model, we used scikit-learn

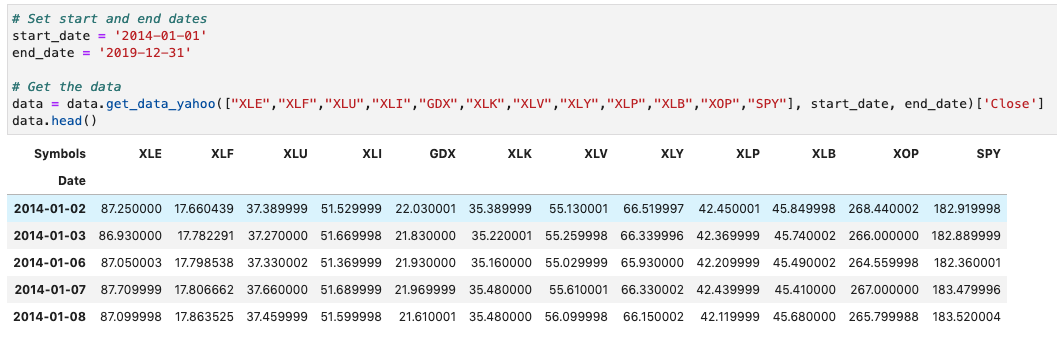
**Step 1: Importing the libraries**

The first step is to import all the necessary libraries and packages we need. If models are not available in the given python environment, then packages needs to be installed first using the pip command.



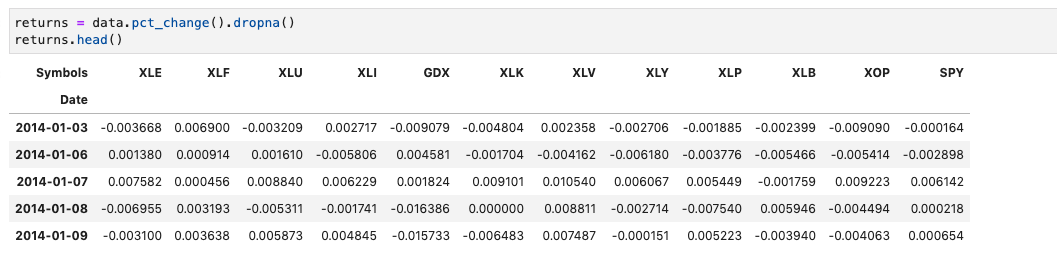
**Step 2: Loading the data set**

Next, we Imported the SP500 and SPDRs data from yahoo finance



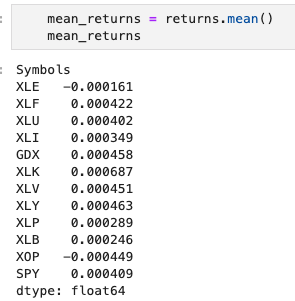
**Step 2: Data Preprocessing**

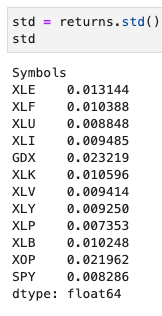
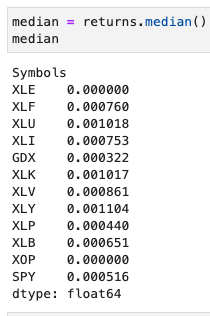
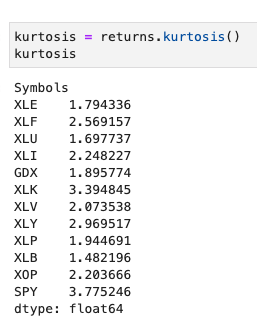
At this stage first we calculate the SP500 returns then we use this data to drop missing values from the data set.

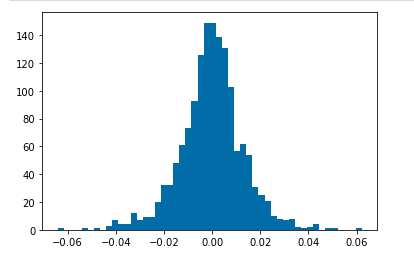


**Step 3: Explanatory Data Analysis**

Here all the major statistical moments are obtained including mean, median, standard deviation, Kurtosis and Skewness. This helps us to understand the nature and distribution of the data. In addition, a trimmed mean has been calculated by eliminating the highest and the lowest 5% and considering these values as outliers. This data set give us the chance to see how the models performs in a data set with extreme values. Finally, we examine whether the data comes from the normal distribution or not.





**Step 4: Data Splitting**

Before implementing the model, we need to split the data into training and testing data sets. Here we assign 70% the data for training the model and 30% for testing.



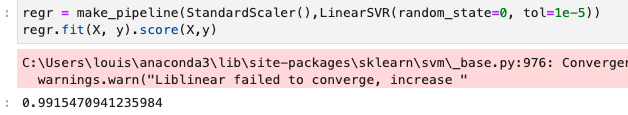


**Step 4: Modelling and Evaluation**

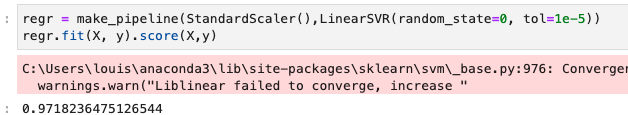
Finally, we construct a data pipeline that first scales the features to a specific range and then we implement the LinearSVR function from the sckit-learn packages to fit a line in our training set.

After that we use the score evaluation metric to see how the model performed in the testing data set. This has been carried out on both trimmed and untrimmed data sets.

Untrimmed data set



Trimmed data set



9.3)

This assignment was performed with the collaboration of all group members. For simplicity and efficiency, each group member has been designated as the lead for a specific task:

* Data collection and Modelling by Louis Regnier-Vigouroux
* Model adjustment and Copulas by Mantobaye Moundigbaye
* Discussion about the results and choosing the best model and report writing by Yonas Menghis Berhe

As this is a collaborative work, each group member helped others to perform their tasks and provided comments to help improve the submitted solution.