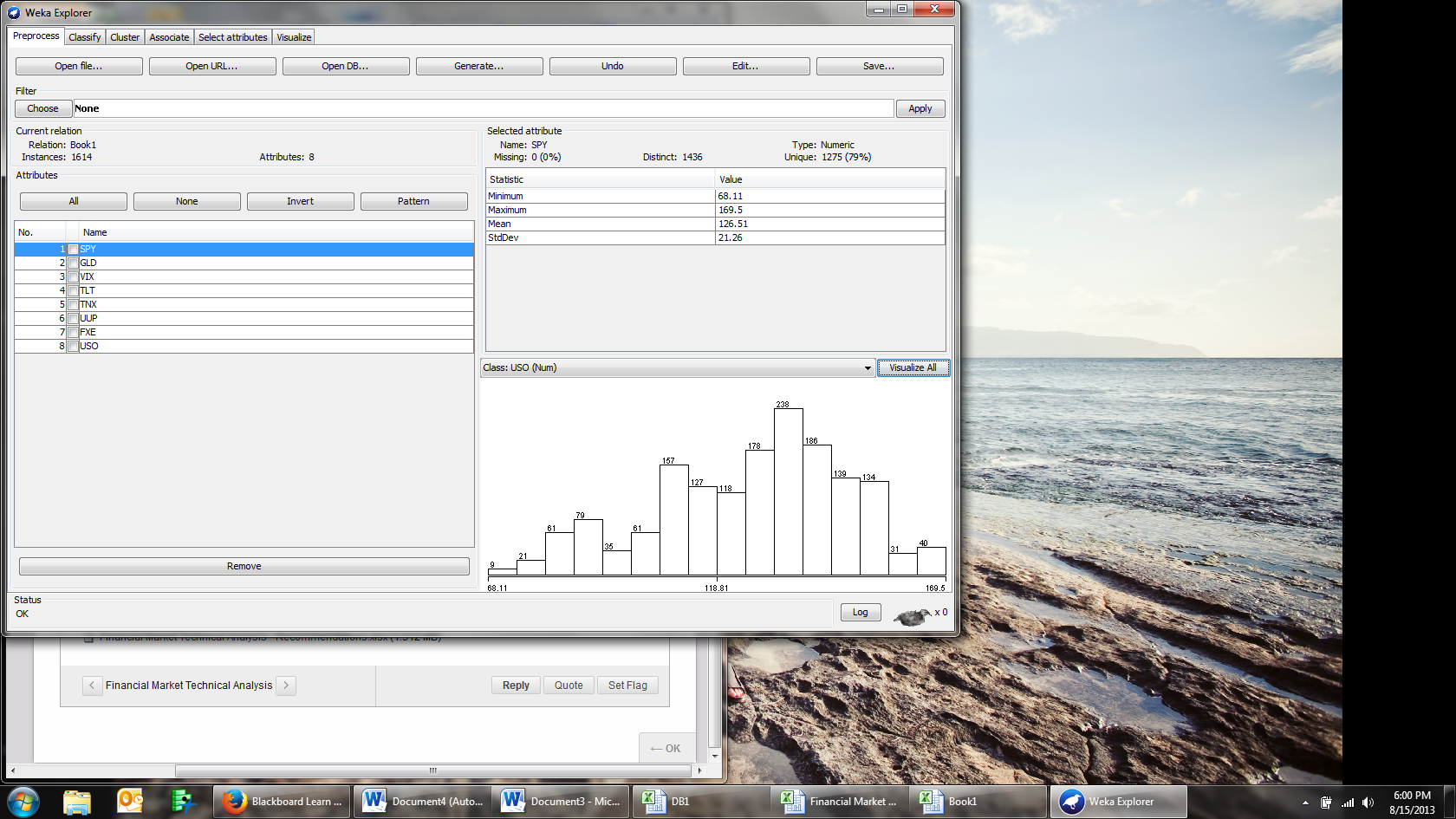
In this analysis, I am looking to create a soft sell strategy such that one can sell SPY before the loss is greater than -.05.

In order to do this, I introduced a MA 100 column. My additional rule is: If the difference (subtraction) percentage between the current SPY price and MA(100) minus SPY price MA(200) is less than .009 sell. This shows that the investment is slowing down over time, such that a loss could get larger in the future. Given that there is no current ‘dump’ strategy, this is my initial stab at it.

Part 2

Shown below is the process I used to analyze the specific indexes in Weka.

As one can see all 1614 instances are uploaded into weka:



Using Linear Regression, I found the overall correlation coefficient for the values:

=== Run information ===

Scheme:weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8

Relation: Book1

Instances: 1614

Attributes: 8

SPY

GLD

VIX

TLT

TNX

UUP

FXE

USO

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Linear Regression Model

USO =

0.5397 \* SPY +

0.4386 \* GLD +

0.0675 \* VIX +

0.4552 \* TLT +

4.3761 \* TNX +

17.6082 \* UUP +

2.9617 \* FXE +

-949.6331

Time taken to build model: 0.32 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient 0.8817

Mean absolute error 7.1787

Root mean squared error 9.1987

Relative absolute error 49.4229 %

Root relative squared error 47.1295 %

Total Number of Instances 1614

I then used a neural network:

=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Relation: Book1

Instances: 1614

Attributes: 8

SPY

GLD

VIX

TLT

TNX

UUP

FXE

USO

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Linear Node 0

Inputs Weights

Threshold 0.18474205599368965

Node 1 1.344917172210045

Node 2 -0.9334896654424535

Node 3 -1.0412229529934953

Node 4 -0.8478311117925162

Sigmoid Node 1

Inputs Weights

Threshold -4.038108886475623

Attrib SPY -3.305540087449812

Attrib GLD -1.059378242267931

Attrib VIX -0.7222720458943221

Attrib TLT 1.9220118579260972

Attrib TNX 5.097060142993473

Attrib UUP 2.3389761088884433

Attrib FXE 5.3219388136356045

Sigmoid Node 2

Inputs Weights

Threshold -0.7506763040547699

Attrib SPY 1.9824203580892872

Attrib GLD 0.5301551356874393

Attrib VIX 2.3091129269199575

Attrib TLT 0.9516637204579683

Attrib TNX -0.017231738192730387

Attrib UUP 1.988650988215034

Attrib FXE 0.05828512213870866

Sigmoid Node 3

Inputs Weights

Threshold -4.470502941693606

Attrib SPY -10.02740320035218

Attrib GLD -0.029747892703931914

Attrib VIX -3.050639919431909

Attrib TLT -1.0253442031119933

Attrib TNX -2.4563865290453473

Attrib UUP 0.08518365360729421

Attrib FXE 3.509223354831573

Sigmoid Node 4

Inputs Weights

Threshold -0.5622532655625692

Attrib SPY 0.9093129510269953

Attrib GLD 1.0052371768859028

Attrib VIX -0.3409376659367143

Attrib TLT -0.13404498593068753

Attrib TNX 2.0223831529984477

Attrib UUP -3.959844410676648

Attrib FXE -5.867664938918518

Class

Input

Node 0

Time taken to build model: 1.96 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient 0.982

Mean absolute error 2.8582

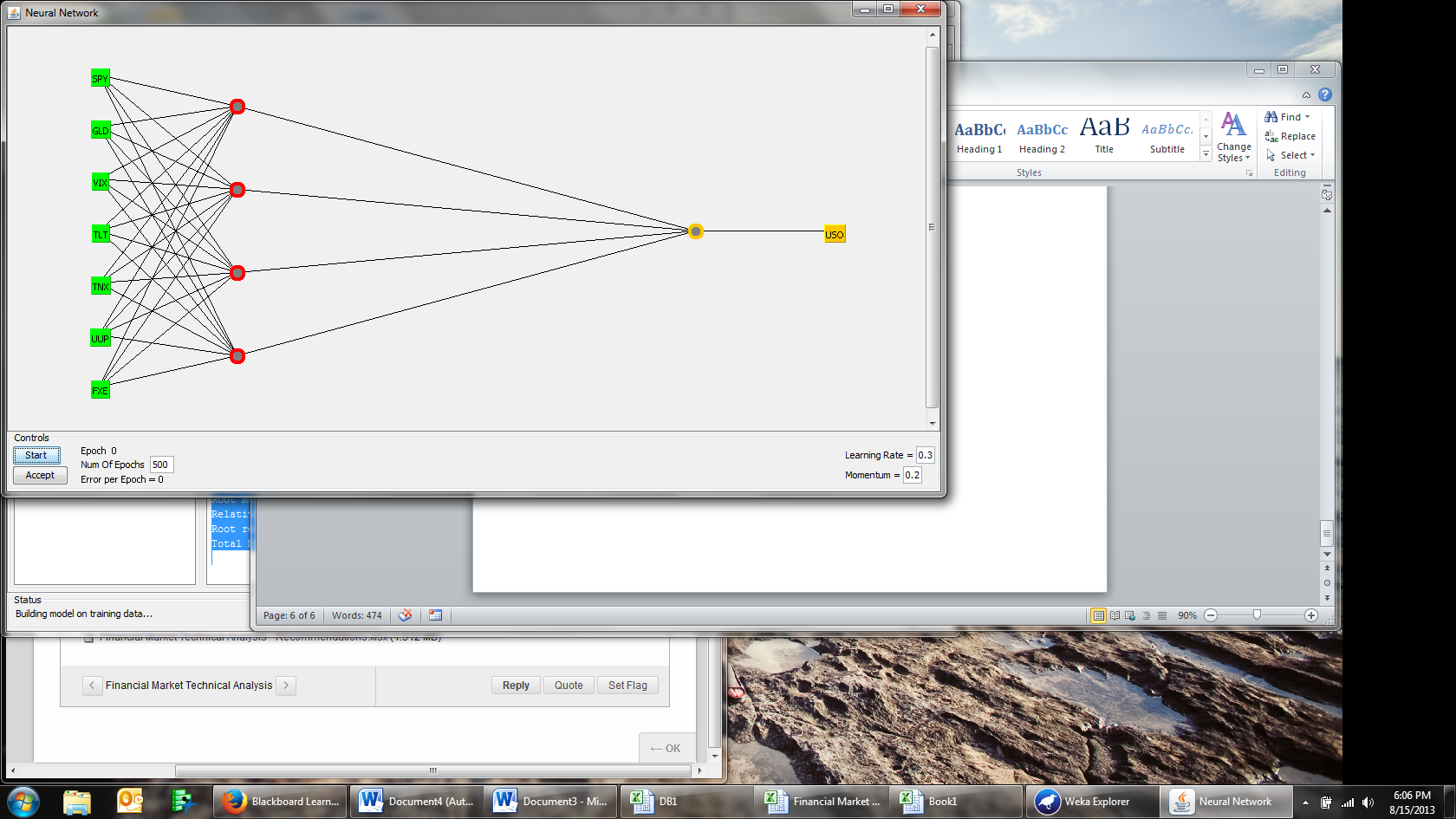
Root mean squared error 3.7613

Relative absolute error 19.6778 %

Root relative squared error 19.2708 %

Total Number of Instances 1614

This is what it looked like:



Actionable Insights:

Moving forward, I like to play with this more. I do not have enough time to build a better model. It is my opinion that our final project should encompass data like in this DB.