Hello everyone,

It has been a while since I posted an indicator, so thought I would share this project I did for fun.

This indicator is an attempt to develop a pseudo Random Forest classification decision matrix model for Pinescript.
This is not a full, robust Random Forest model by any stretch of the imagination, but it is a good way to showcase how decision matrices can be applied to trading and within Pinescript.

As to not market this as something it is not, I am simply calling it the "Simple Decision Matrix Classification Algorithm". However, I have stolen most of the aspects of this machine learning algo from concepts of Random Forest modelling.

How it works:

With models like Support Vector Machines (SVM), Random Forest (RF) and Gradient Boosted Machine Learning (GBM), which are commonly used in Machine Learning Classification Tasks (MLCTs), this model operates similarity to the basic concepts shared amongst those modelling types. While it is not very similar to SVM, it is very similar to RF and GBM, in that it uses a "voting" system.

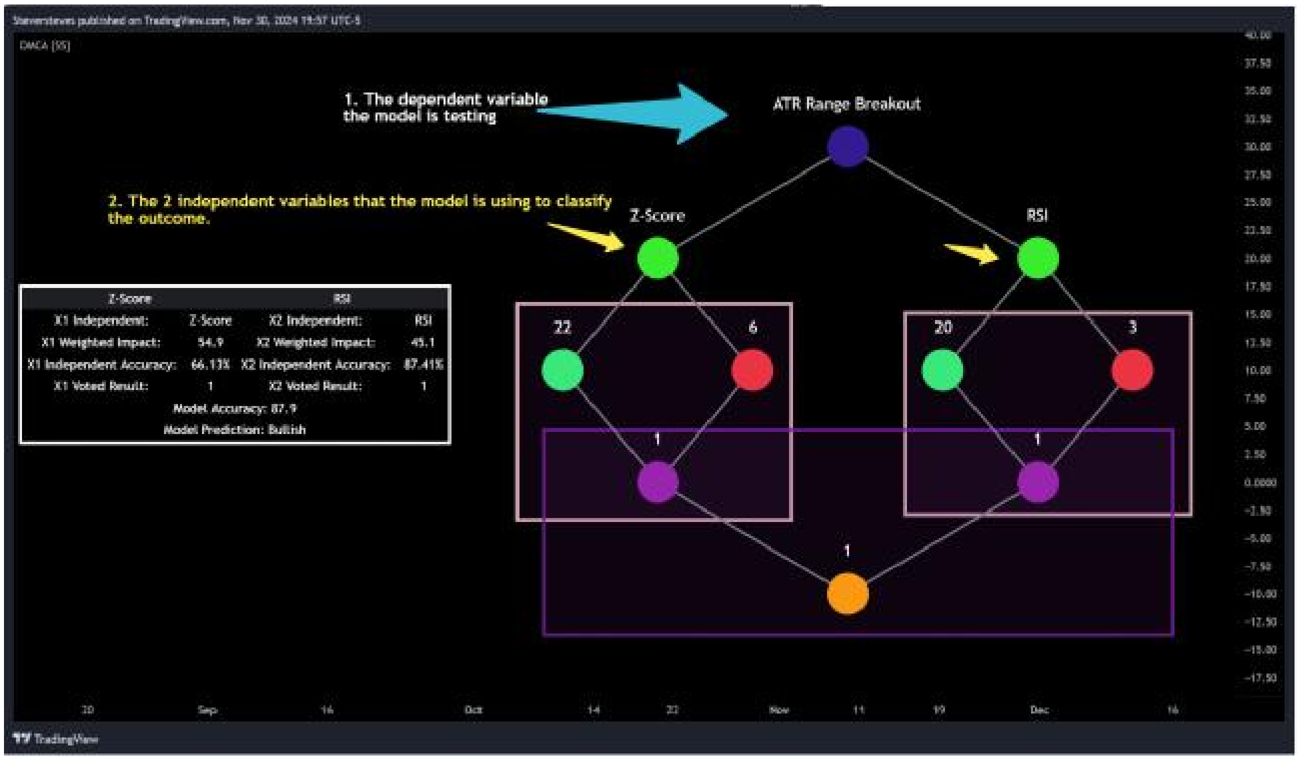
What do I mean by voting system?

How most classification MLAs work is by feeding an input dataset to an algorithm. The algorithm sorts this data, categorizes it, then introduces something called a confusion matrix (essentially sorting the data in no apparently order as to prevent over-fitting and introduce "confusion" to the algorithm to ensure that it is not just following a trend).
From there, the data is called upon based on current data inputs (so say we are using RSI and Z-Score, the current RSI and Z-Score is compared against other RSI's and Z-Scores that the model has saved). The model will process this information and each "tree" or "node" will vote. Then a cumulative overall vote is casted.

How does this MLA work?

This model accepts 2 independent variables. In order to keep things simple, this model was kept as a three node model. This means that there are 3 separate votes that go in to get the result. A vote is casted for each of the two independent variables and then a cumulative vote is casted for the overall verdict (the result of the model's prediction).

The model actually displays this system diagrammatically and it will likely be easier to understand if we look at the diagram to ground the example:


Stevenstevens published on TradingView.com, Nov 30, 2024 19:57 UTC-5
1. The dependent variable the model is testing
ATR Range Breakout
2. The 2 independent variables that the model is using to classify the outcome.
Z-Score
RSI
22
6
20
3
1
1
1
TradingView

Z-Score		RSI	
X1 Independent:	Z-Score	X2 Independent:	RSI
X1 Weighted Impact:	54.9	X2 Weighted Impact:	45.1
X1 Independent Accuracy:	66.13%	X2 Independent Accuracy:	87.41%
X1 Voted Result:	1	X2 Voted Result:	1

Model Accuracy: 87.9
Model Prediction: Bullish

In the diagram, at the very top we have the classification variable that we are trying to predict. In this case, we are trying to predict whether there will be a breakout/breakdown outside of the normal ATR range (this is either yes or no question, hence a classification task).

So the question forms the basis of the input. The model will track at which points the ATR range is exceeded to the upside or downside, as well as the other variables that we wish to use to predict these exceedences. The ATR range forms the basis of all the data flowing into the model.

Then, at the second level, you will see we are using Z-Score and RSI to predict these breaks. The circle will change colour according to "feature importance". Feature importance basically just means that the indicator has a strong impact on the outcome. The stronger the importance, the more green it will be, the weaker, the more red it will be.

We can see both RSI and Z-Score are green and thus we can say they are strong options for predicting a breakout/breakdown.

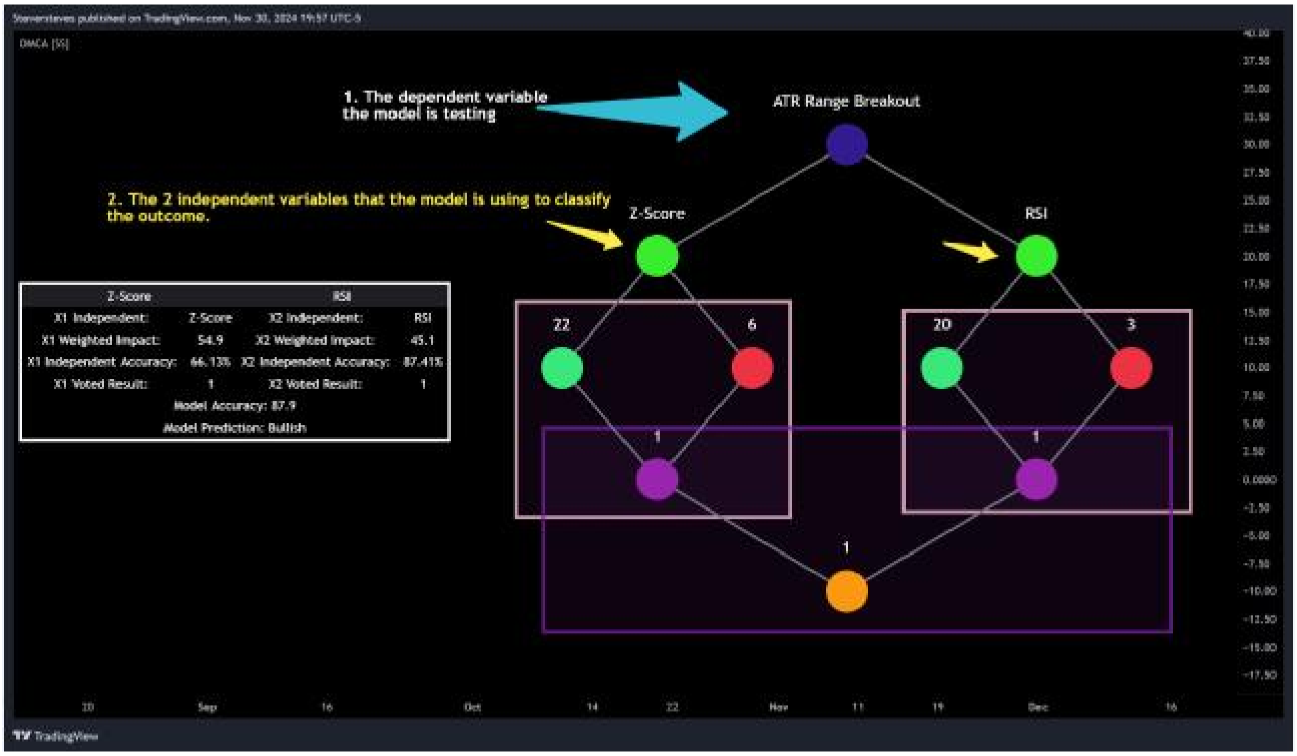
So then we move down to the actual voting mechanisms. You will see the 2 pink boxes. These are the first lines of voting. What is happening here is the model is identifying the instances that are most similar and whether the classification task we have assigned (remember out ATR exceedance classifier) was either true or false based on RSI and Z-Score.
These are our 2 nodes. They both cast an individual vote. You will see in this case, both cast a vote of 1. The options are either 1 or 0. A vote of 1 means "Yes" or "Breakout likely".

However, this is not the only voting the model does. The model does one final vote based on the 2 votes. This is shown in the purple box. We can see the final vote and result at the end with the orange circle. It is 1 which means a range exceedance is anticipated and the most likely outcome.

The Data Table Component

The model has many moving parts. I have tried to represent the pivotal functions diagrammatically, but some other important aspects and background information must be obtained from the companion data table.

If we bring back our diagram from above:


Stevenstevens published on TradingView.com, Nov 30, 2024 19:57 UTC-5
DMCA [SS]
1. The dependent variable the model is testing
ATR Range Breakout
2. The 2 independent variables that the model is using to classify the outcome.
Z-Score
RSI
22
6
20
3
1
1
1
TradingView

Z-Score		RSI	
X1 Independent:	Z-Score	X2 Independent:	RSI
X1 Weighted Impact:	54.9	X2 Weighted Impact:	45.1
X1 Independent Accuracy:	66.13%	X2 Independent Accuracy:	87.41%
X1 Voted Result:	1	X2 Voted Result:	1
Model Accuracy: 87.9			
Model Prediction: Bullish			

We can see the data table to the left.
The data table contains 2 sections, one for each independent variable. In this case, our independent variables are RSI and Z-Score.
The data table will provide you with specifics about the independent variables, as well as about the model accuracy and outcome.

If we take a look at the first row, it simply indicates which independent variable it is looking at. If we go down to the next row where it reads "Weighted Impact", we can see a corresponding percent. The "weighted impact" is the amount of representation each independent variable has within the voting scheme. So in this case, we can see its pretty equal, 45% and 55%, This tells us that there is a slight higher representation of z-score than RSI but nothing to worry about.

If there was a major over-respresentation of greater than 30 or 40%, then the model would risk being skewed and voting too heavily in favour of 1 variable over the other.

If we move down from there we will see the next row reads "independent accuracy". The voting of each independent variable's accuracy is considered separately. This is one way we can determine feature importance, by seeing how well one feature augments the accuracy. In this case, we can see that RSI has the greatest importance, with an accuracy of around 87% at predicting breakouts. That makes sense as RSI is a momentum based oscillator.

Then if we move down one more, we will see what each independent feature (node) has voted for. In this case, both RSI and Z-Score voted for 1 (Breakout in our case).
You can weigh these in collaboration, but its always important to look at the final verdict of the model, which if we move down, we can see the "Model prediction" which is "Bullish".

If you are using the ATR breakout, the model cannot distinguish between "Bullish" or "Bearish", must that a "Breakout" is likely, either bearish or bullish. However, for the other classification tasks this model can do, the results are either Bullish or Bearish.

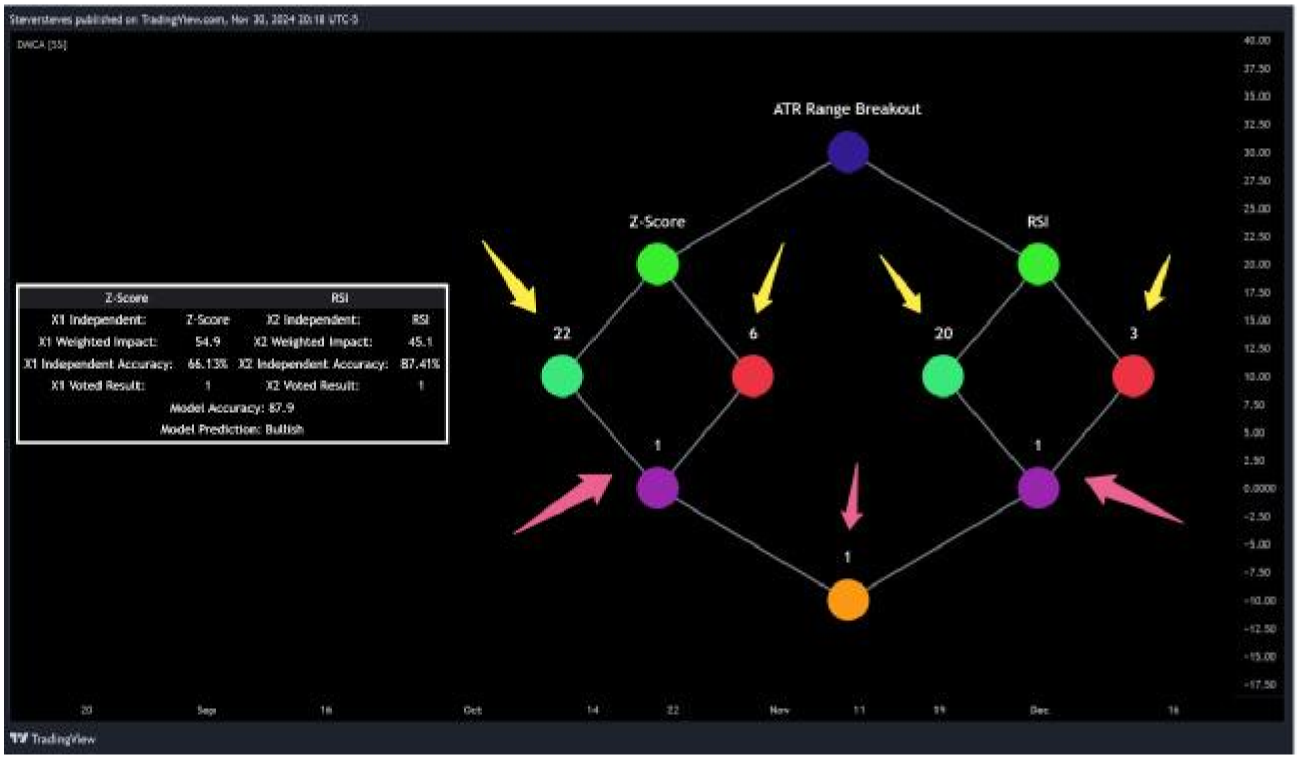
Using the Function:
Okay so now that all that technical stuff is out of the way, let's get into using the function. First of all this function innately provides you with 3 possible classification tasks. These include:

1. Predicting Red or Green Candle
2. Predicting Bullish / Bearish ATR
3. Predicting a Breakout from the ATR range

The possible independent variables include:
1. Stochastics,
2. MFI,
3. RSI,
4. Z-Score,
5. EMAs,
6. SMAs,
7. Volume

The model can only accept 2 independent variables, to operate within the computation time limits for pine execution.

Let's quickly go over what the numbers in the diagram mean:


Steversteves published on TradingView.com, Nov 30, 2024 20:18 UTC-5
DNCA (55)
ATR Range Breakout
Z-Score
RSI
22
6
20
3
1
1
1
TradingView

Z-Score		RSI	
X1 Independent:	Z-Score	X2 Independent:	RSI
X1 Weighted Impact:	54.9	X2 Weighted Impact:	45.1
X1 Independent Accuracy:	66.13%	X2 Independent Accuracy:	87.41%
X1 Voted Result:	1	X2 Voted Result:	1
Model Accuracy: 87.9			
Model Prediction: Bullish			

The numbers being pointed at with the yellow arrows represent the cases the model is sorting and voting on. These are the most identical cases and are serving as the voting foundation for the model.

The numbers being pointed at with the pink candle is the voting results.

Extrapolating the functions (For Pine Developers:

So this is more of a feature application, so feel free to customize it to your liking and add additional inputs. But here are some key important considerations if you wish to apply this within your own code:

1. This is a BINARY classification task. The prediction must either be 0 or 1.
2. The function consists of 3 separate functions, the 2 first functions serve to build the confusion matrix and then the final "random_forest" function serves to perform the computations. You will need all 3 functions for implementation.
3. The model can only accept 2 independent variables.

I believe that is the function. Hopefully this wasn't too confusing, it is very statsy, but its a fun function for me! I use Random Forest excessively in R and always like to try to convert R things to Pinescript.

Hope you enjoy!

Safe trades everyone!