Predicting Stock Movement with Random Forest

Momentum indicators are technical analysis tools used by traders to determine the speed or rate of change of a security's price. They help identify the strength of a trend and potential points where the trend might reverse.

Indicator Calculation: Relative Strength Index (RSI):-

Definition:

RSI is a popular momentum indicator that determines whether the stock is overbought or oversold. A stock is said to be overbought when the demand unjustifiably pushes the price upwards. This condition is generally interpreted as a sign that the stock is overvalued, and the price is likely to go down. A stock is said to be oversold when the price goes down sharply to a level below its true value. This is a result caused due to panic selling. RSI ranges from 0 to 100, and generally, when RSI is above 70, it may indicate that the stock is overbought and when RSI is below 30, it may indicate the stock is oversold.

Formula:

$$RSI = 100 - \frac{100}{1 + RS}$$

Now there might be a few extra steps in between, but the general idea is the same across each indicator. Now for the RSI indicator, We need to identify the up days and down days. After that, I need to make sure I have the absolute values for down days, or else the calculation won't be correct, so modify that column and then calculate the EMA of both the Up and Down columns. Finally, calculate the Relative strength metric and pass that through to the RSI calculation.

Indicator Calculation: Stochastic Oscillator

Definition From Paper:

Stochastic Oscillator follows the speed or the momentum of the price. As a rule, momentum changes before the price changes. It measures the level of the closing price relative to the low-high range over a period of time.

Formula:

$$K = 100 * \frac{(C - L_{14})}{(H_{14} - L_{14})}$$

$$where,$$

$$C = \text{Current Closing Price}$$

$$L_{14} = \text{Lowest Low over the past 14 days}$$

$$H_{14} = \text{Highest High over the past 14 days}$$

%D = %D: 3-day simple moving average of %K to smooth out the %K values.

Interpretation:

- The %K line oscillates between 0 and 100.
- Readings above 80 are considered overbought, suggesting a potential sell signal.
- Readings below 20 are considered oversold, suggesting a potential buy signal.
- Divergence between the %K and %D lines can indicate potential trend reversals or strength.

1. **Model Enhancement**: By including %K and %D divergence as a feature, your model can potentially learn to recognize these patterns autonomously, improving its ability to forecast price movements more accurately.
2. **Trend Reversals**: Divergence between %K and %D can signal potential shifts in momentum or trend reversals. This information can add predictive power to your model by capturing these critical points in price movements.

Indicator Calculation: Williams %R

The Williams %R is a momentum indicator that measures overbought and oversold levels. It compares the closing price to the high-low range over a specified period. The Williams %R is similar to the Stochastic Oscillator but is plotted on a negative scale.

Williams %R ranges from -100 to 0. When its value is above -20, it indicates a sell signal and when its value is below -80, it indicates a buy signal.

$$R = \frac{(H_{14} - C)}{(H_{14} - L_{14})} * -100$$

$$where,$$

$$C = \text{Current Closing Price}$$

$$L_{14} = \text{Lowest Low over the past 14 days}$$

$$H_{14} = \text{Highest High over the past 14 days}$$

- **Choose the look-back period** (typically 14 days).
- **Identify the highest high** and **lowest low** over the look-back period.
- **Calculate the Williams %R** using the formula.

Indicator Calculation: Moving Average Convergence Divergnece (MACD):

EMA stands for Exponential Moving Average. When the MACD goes below the SingalLine, it indicates a sell signal. When it goes above the SignalLine, it indicates a buy signal.

Formula:

$$MACD = EMA_{12}(C) - EMA_{26}(C)$$

$$SignalLine = EMA_9(MACD)$$

$$where,$$

$$MACD = \text{Moving Average Convergence Divergence}$$

$$C = \text{Closing Price}$$

$$EMA_n = \text{n day Exponential Moving Average}$$

The Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. Additionally, a 9-period EMA of the MACD, called the "signal line," is plotted on top of the MACD line, which can function as a trigger for buy and sell signals.

- **Calculate Short-term and Long-term EMAs**:

  - The `ewm` method is used to calculate the Exponential Moving Averages (EMAs). The `span` parameter defines the period, and `adjust=False` ensures that the calculation is consistent with standard EMA formulas.

- **Calculate MACD Line**:

  - The MACD line is simply the difference between the short-term EMA and the long-term EMA.

- **Calculate Signal Line**:

  - The Signal line is the 9-period EMA of the MACD line. Ensure this calculation is uncommented so it runs properly.

- **Calculate MACD Histogram**:
  - The MACD histogram is the difference between the MACD line and the Signal line.

Indicator Calculation: Price Rate Of Change

The Price Rate of Change (ROC) is a momentum oscillator that measures the percentage change in price between the current price and the price a certain number of periods ago. It is a simple yet effective indicator to gauge the momentum and strength of a trend. **Formula:**

$$PROC_t = \frac{C_t - C_{t-n}}{C_{t-n}}$$

$$where,$$

$$PROC_t = \text{Price Rate of Change at time t}$$

$$C_t = \text{Closing price at time t}$$

Steps to Calculate ROC

1. **Choose the look-back period** (N). A common period is 9 (by some research) days, but this can vary depending on the analysis.
2. Subtract the price from N periods ago from the current price.
3. Divide the difference by the price from N periods ago.
4. Multiply the result by 100 to get a percentage.

Indicator Calculation: On Balance Volume

On balance volume (OBV) (Granville 1976) utilizes changes in volume to estimate changes in stock prices. This technical indicator is used to d buying and selling trends of a stock, by considering the cumulative volume: it cumulatively adds the volumes on days when the prices group, and subtracts the volume on the days when prices go down, compared to the prices of the previous day.

Formula:

$$OBV(t) = \begin{cases} \text{OBV(t - 1) + Vol(t) if C(t) > C(t-1)} \\ \text{OBV(t - 1) - Vol(t) if C(t) < C(t-1)} \\ \text{OBV(t - 1) if C(t) = C(t-1)} \end{cases} \text{ where,OBV (t) = on balance volume at time}$$

Key Concepts of OBV:

1. **Volume as a Leading Indicator**: OBV assumes that volume precedes price movements. It suggests that a rise in OBV indicates strong buying pressure, which could lead to higher prices, while a decline in OBV indicates strong selling pressure, potentially leading to lower prices.
2. **Cumulative Total:** OBV is a running total of volume. It is calculated by adding or subtracting the volume of a security on up days or down days, respectively.

Calculation of OBV:

The OBV is calculated as follows:

- OBV Today = OBV Yesterday + Volume, if the closing price is higher than the previous closing price.
- OBV Today = OBV Yesterday - Volume, if the closing price is lower than the previous closing price.
- OBV Today = OBV Yesterday, if the closing price is unchanged.

Interpretation:

- **Rising OBV**: Indicates that volume on up days is stronger than on down days, suggesting accumulation (buying pressure).
- **Falling OBV**: Indicates that volume on down days is stronger than on up days, suggesting distribution (selling pressure).
- **Divergence**: If OBV moves in the opposite direction of the price trend, it can indicate a potential reversal.

Building the Model: Creating the Prediction Column

Now that we have our technical indicators calculated and our price data cleaned up, we are almost ready to build our model. However, we are missing one critical piece of information that is crucial to the model, the column we wish to predict. Now at this point, our data frame doesn't have that column, but we will create it before we feed the data into the model.

1) Create the Prediction column
2) Removing NaN Values.
3) Checking the correlation between features
4) Splitting the Data.

Those columns will serve as our X, and our Y column will be the Prediction column, the column that specifies whether the stock closed up or down compared to the previous day.

X_cols = df[['RSI' , '%K', '%K_%D_Divergence','MACD','Williams_%R', 'MACD_EMA', 'Price_Rate_Of_Change', 'On Balance Volume', 'MACD_Histogram']]

Y_cols = df['PREDICTIONS']

Hence we've selected our columns, we need to split the data into a training and test set. Using SciKit learn train_test_split for splitting the data , which will take our X_Cols and Y_Cols and split them based on the size we input. In our case, we have the test_size be '20.

After we've split the data, we can create our RandomForestClassifier model.

Model Evaluation: Accuracy

We built our model, so let's see how accurate it is. Used SciKit learn for evaluating our model

Evaluation matrix : - Accuracy matrix.

When evaluating the performance of a model for stock price prediction, choosing the right evaluation metric is crucial. While accuracy is a common metric, it might not always be the best choice, especially in the context of stock price prediction. Here’s why you might prefer using the balanced accuracy metric over simple accuracy:

1. Imbalanced Data

- **Stock Market Characteristics**: Stock prices often do not have a balanced distribution of outcomes. For instance, predicting whether a stock will go up or down might not be a 50-50 split. There might be more instances of prices going up than down, or vice versa.
- **Balanced Accuracy**: This metric accounts for imbalances in the data by averaging the true positive rate (sensitivity) and true negative rate (specificity). This gives a more accurate picture of the model's performance across both classes (e.g., price going up and price going down).

2. Sensitivity to Class Imbalance

- **Accuracy**: In the presence of class imbalance, accuracy can be misleading. For example, if 90% of the days the stock price goes up, a model that predicts "up" all the time will have 90% accuracy but is useless for predicting the days the price goes down.
- **Balanced Accuracy**: By considering both the true positive rate and true negative rate, balanced accuracy provides a more nuanced view of how well the model is performing across different classes, ensuring that both are given equal importance.

3. Better Decision-Making

- **Investment Strategies**: In stock trading, making correct predictions for both upward and downward movements is critical for strategies such as short selling or stop-loss orders. A model that only predicts one class well might lead to significant financial losses.
- **Balanced Accuracy**: By ensuring that the model is evaluated on its ability to predict both classes equally well, balanced accuracy supports better decision-making in trading strategies.

Hyperparameter tuning

Hyperparameter tuning can significantly improve the performance of your Random Forest model. I use two methods to tune hyperparameters:

Grid Search is an exhaustive search technique where you specify a grid of hyperparameters and the algorithm tests all possible combinations. It is thorough but can be computationally expensive, especially if the grid is large.

Random Search is a technique where hyperparameter values are sampled randomly from the specified distributions. It is generally faster than Grid Search because it does not try every combination, and it can still find good hyperparameters, especially if the search space is large and complex.

Finally by using the best parameters I got an accuracy of :

Best parameters found: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}

Best balanced accuracy score from Grid Search: 85.3080881510705

Model Evaluation: Classification Report

To get a more detailed overview of how the model performed, we can build a classification report that will compute the F1_Score, the Precision, the Recall, and the Support. Now, I'm assuming you don't know what these metrics are, so let's take some time to go over them.

Accuracy:

Accuracy measures the portion of all testing samples classified correctly and is defined as the following:

$$Accuracy = \frac{tp + tn}{(tp + tn) + (fp - fn)}$$

$$\begin{aligned} where, \\ tp &= \text{True Positive} \\ tn &= \text{True Negative} \\ fp &= \text{False Positive} \\ fn &= \text{False Negative} \end{aligned}$$

Recall

Recall (also known as sensitivity) measures the ability of a classifier to correctly identify positive labels and is defined as the following:

$$Recall = \frac{tp}{(tp + fn)}$$

$$\begin{aligned} &where, \\ tp &= \text{True Positive} \\ fn &= \text{False Negative} \end{aligned}$$

Specificity

Specificity measures the classifier's ability to correctly identify negative labels and is defined as the following:

$$Specificity = \frac{tn}{(tn + fp)}$$

$$\begin{aligned} &where, \\ tn &= \text{True Negative} \\ fp &= \text{False Positive} \end{aligned}$$

Percision

Precision measures the proportion of all correctly identified samples in a population of samples which are classified as positive labels and is defined as the following:

$$Percision = \frac{tp}{(tp + fp)}$$

$$\begin{aligned} &where, \\ tp &= \text{True Positive} \\ fp &= \text{False Positive} \end{aligned}$$

Interpreting the Classification Report

When it comes to evaluating the model, there we generally look at the accuracy. If our accuracy is high, it means our model is correctly classifying items.

In some cases, we will have models that may have low precision or high recall. It's difficult to compare two models with low precision and high recall or vice versa. To make results comparable, we use a metric called the F-Score. The F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

Model Evaluation: Confusion Matrix

Feature Importance

With any model, we want to have an idea of what features are helping explain most of the model, as this gives you insight as to why you're getting the results you are. With Random Forest, we can identify some of our most important features or. In some cases, some of our features might not be very important, or in other words, when compared to additional features, don't explain much of the model.

Feature Importance Graphing

ROC Curve

The Receiver Operating Characteristic is a graphical method to evaluate the performance of a binary classifier. A curve is drawn by plotting True Positive Rate (sensitivity) against False Positive Rate (1 - specificity) at various threshold values. ROC curve shows the trade-off between sensitivity and specificity. When the curve comes closer to the left-hand border and the top border of the ROC space, it indicates that the test is accurate. The closer the curve is to the top and left-hand border, the more accurate the test is. If the curve is close to the 45 degrees diagonal of the ROC space, it means that the test is not accurate. ROC curves can be used to select the optimal model and discard the suboptimal ones.