



# Defining and Computing Trend Probability for Machine Learning Models in Financial Markets

## Abstract

Traditional financial models often focus on predicting future asset prices, relying on time series forecasting and technical indicators. However, predicting exact price movements can be unreliable due to market complexity and inherent randomness. This article introduces a novel framework for defining and computing the **probability of market trends**, offering a more flexible and realistic target for machine learning models. By using moving averages and a sigmoid transformation, we convert market dynamics into a probabilistic trend

signal, bridging technical analysis with modern data-driven models. This probability will also serve as the foundational target for machine learning models developed to explore the influence of **divine influence** in financial markets.

## 1 Introduction

Machine learning (ML) models in financial markets typically focus on forecasting asset prices or classifying price movements. However, precise price prediction is inherently difficult due to the complexity and noise in market data. A more effective approach involves predicting the **probability of market trends** (uptrend, downtrend, or neutral), aligning better with real-world trading decisions.

This article presents a systematic method to compute trend probabilities using price data, which can serve as a target variable in supervised ML models for market prediction. Additionally, this probability will be integrated into machine learning models designed to capture how **divine influence** could subtly guide market behaviors.

## 2 Methodology for Computing Trend Probability

### 2.1 Conceptual Overview

The goal is to quantify the market's directional bias as a probability rather than a binary outcome. This involves:

1. **Capturing Market Trends:** Using moving averages to identify trend direction and strength.
2. **Normalizing Trend Signals:** Scaling the trend signal relative to price movements.
3. **Probability Transformation:** Applying a sigmoid function to produce a probability between 0 and 1.

### 2.2 Mathematical Formulation

#### Step 1: Calculate Moving Averages

Let  $P_t$  denote the price at time  $t$ . Compute short-term and long-term moving averages:

$$MA_{extShort}(t) = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

$$MA_{extLong}(t) = \frac{1}{m} \sum_{i=0}^{m-1} P_{t-i}$$

Where:

- $n$ : Short-term window (e.g., 5 periods)
- $m$ : Long-term window (e.g., 20 periods)

### Step 2: Compute the Trend Signal

The difference between the short-term and long-term moving averages captures the market's directional bias:

$$\Delta MA(t) = MA_{extShort}(t) - MA_{extLong}(t)$$

### Step 3: Normalize the Trend Signal

Normalize the trend signal relative to the current price to scale the result:

$$\text{Normalized Trend}(t) = \frac{\Delta MA(t)}{P_t}$$

### Step 4: Apply Sigmoid Transformation

Transform the normalized signal into a probability:

$$P_{\text{trend}}(t) = \sigma(k \cdot \text{Normalized Trend}(t)) = \frac{1}{1 + e^{-k \cdot \frac{\Delta MA(t)}{P_t}}}$$

Where:

- $\sigma$  is the sigmoid function.
- $k$  is a scaling factor controlling sensitivity.

### 3 Python Implementation and Visualization

Here is the complete version of the Python code with a single threshold of 0.5 for deciding between Buy and Sell signals.



### 4 Integration with Machine Learning Models

The computed **Trend Probability** will serve as the target variable for machine learning models designed to detect subtle influences on market behavior. Specifically, this model will be used to explore how divine influence could subtly adjust traders' decision-making processes.

- **Classification Models:** Classify whether the market is in an uptrend or downtrend based on probability.
- **Regression Models:** Predict continuous trend probabilities to adjust trading strategies dynamically.

A natural approach to completing the machine learning framework is to use **Trend Probabilities** as input features ( $x_i$ ) for the models. These input probabilities can be computed from various market indicators, with **volume-based**

**trend probabilities** serving as a practical example. By leveraging the dynamics of trading volume, this approach captures the momentum and sentiment in the market, which aligns well with the analogy of the trader as a neuron.

In this framework, each trader is conceptualized as a **neuron** in a neural network, where decisions are based on weighted inputs. The **volume-based trend probability** functions as a refined input signal that encapsulates the collective behavior of the market. Similar to how neurons process signals in the brain, the trader integrates this probabilistic input with other influencing factors—such as price trends, emotions, and biases—to arrive at a decision (buy or sell).

Using probabilities as inputs provides a more flexible and realistic representation of human decision-making, as it reflects the inherent uncertainties and nuances in trading behavior. This abstraction not only strengthens the model's alignment with real-world market dynamics but also supports the hypothesis that traders act as decision nodes influenced by both market-driven factors and external adjustments, such as divine influence.

## 5 Conclusion

This framework introduces a **probabilistic approach** to market trend prediction, shifting from precise price forecasting to trend likelihood estimation. By leveraging moving averages and a sigmoid transformation, this model produces a more flexible and interpretable target for machine learning models. This probability will also act as a foundational component for models investigating **divine influence** on market behavior.

Future work could involve integrating additional indicators, optimizing scaling factors, and testing this approach with real market data for model training and validation.