

# СТЕКЛЯННЫЕ МОДЕЛИ ДЛЯ ВАЛЮТНОГО РЫНКА: АРХИТЕКТУРА И СВОЙСТВА АДДИТИВНЫХ ИНТЕРПРЕТИРУЕМЫХ МОДЕЛЕЙ В АВТОМАТИЗИРОВАННОЙ ТОРГОВЛЕ

**Климин Павел Юрьевич**

Аспирант, Санкт-Петербургский государственный  
архитектурно-строительный университет  
k2x2p@yandex.ru

## GLASS BOXES FOR THE FOREIGN EXCHANGE MARKET: ARCHITECTURE AND PROPERTIES OF ADDITIVE INTERPRETED MODELS IN AUTOMATED TRADING

**P. Klimin**

*Summary.* The paper considers the application of glass (intrinsically interpretable) models to the tasks of developing automated trading systems in the foreign exchange market. An architecture for building a trading robot is proposed, in which the core is a generalized additive model with tree boosting (Explicable Boosting Machine, EBM). The model is trained in Python, exported to ONNX format, and then used in the MQL5 Expert Advisor to generate trading signals in real time. It is shown that the additive representation allows you to simultaneously ensure high-quality forecasting of the direction of price movement and a transparent decomposition of the contribution of features to each trading decision. Interpretability metrics specialized for financial time series are discussed: the stability of the forms of functions, the consistency of the signs of effects with economic intuition, and the «turnover» of factors that determine transactions. The final section connects the theoretical constructions with the practical results of testing the adviser on historical data.

*Keywords:* explicable AI, glass model, Explicable Boosting Machine, ONNX, MQL5, foreign exchange market, algorithmic trading, automated trading systems.

*Аннотация.* В работе рассматривается применение стеклянных (intrinsically interpretable) моделей к задачам разработки автоматизированных торговых систем на валютном рынке. Предлагается архитектура построения торгового робота, в которой ядром является обобщённая аддитивная модель с бустингом деревьев (Explicable Boosting Machine, EBM). Модель обучается в среде Python, экспортируется в формат ONNX и далее используется в советнике MQL5 для генерации торговых сигналов в реальном времени. Показано, что аддитивное представление позволяет одновременно обеспечить высокое качество прогноза направления движения цены и прозрачную декомпозицию вклада признаков в каждое торговое решение. Обсуждаются специализированные для финансовых временных рядов метрики интерпретируемости: устойчивость форм функций, согласованность знаков эффектов с экономической интуицией и «оборот» факторов, обуславливающих сделки. Финальный раздел связывает теоретические построения с практическими результатами тестирования советника на исторических данных.

*Ключевые слова:* объяснимый ИИ, стеклянная модель, Explainable Boosting Machine, ONNX, MQL5, валютный рынок, алгоритмическая торговля, автоматизированные торговые системы.

### Введение

Применение машинного обучения в алгоритмической торговле традиционно опирается на ансамбли деревьев или глубокие нейросети, которые дают высокую прогностическую точность, но фактически функционируют как «чёрные ящики». Непрозрачность логики принятия решений усложняет контроль рыночных рисков, отладку торговых алгоритмов и соответствие нормативным требованиям к объяснимости автоматизированных решений в финансовой сфере.

Параллельно развивается направление объяснимого искусственного интеллекта (Explainable AI, XAI), в рам-

ках которого выделяется класс стеклянных моделей (glassbox models) — архитектур, изначально спроектированных так, чтобы их решения могли быть интерпретированы напрямую, без аппроксимации методами постфактум-анализа (post hoc-методы). Ключевой представитель этого класса Explainable Boosting Machine (EBM), обобщённая аддитивная модель с циклическим градиентным бустингом деревьев и автоматическим поиском взаимодействий.

**Цель статьи** — исследование и формальное описание «стеклянной» аддитивной модели типа Explainable Boosting Machine (EBM) в задачах анализа финансовых временных рядов, а также обоснование возможности

интеграции данной модели в программную инфраструктуру торговых советников платформы MQL5.

**Постановка задачи и требования к торговым сигналам**

Рассматривается временной ряд котировок валютной пары, например EURUSD, на равномерной сетке времени  $t_1, t_2, \dots, t_T$ . Пусть  $C_t$  — цена закрытия бара в момент  $t$ .

Формируем бинарную целевую переменную  $y_t = \begin{cases} 1, & C_{t+1} - C_t > 0, \\ 0, & C_{t+1} - C_t \leq 0, \end{cases}$  описывающую направление следующего бара. Вектор признаков  $x_t \in \mathbb{R}^p$  включает технические и волатильностные показатели, вычисленные по данным на интервале  $[t - K, t]$ :

- нормированные доходности  $r_t = \frac{C_t - C_{t-1}}{C_{t-1}}$  и их скользящие средние;
- индикатор средней истинной волатильности  $ATR_t$  и его производные;
- осцилляторы типа RSI, стохастик;
- относительное положение цены в дневном диапазоне  $pos_t = \frac{C_t - L_t^{(D)}}{H_t^{(D)} - L_t^{(D)}}$ ;
- категориальные признаки, отражающие состояние тренда (направление движения и т.п.).

**Задача прогнозирования.** Найти параметризованную модель  $f_\theta$ , дающую оценку вероятности  $\hat{p}_t = \mathbb{P}(y_t = 1 | x_t) = f_\theta(x_t)$  путём минимизации усреднённой логистической потери  $\mathcal{L}(\theta) = -\frac{1}{N} \sum_{t \in \mathcal{I}_{train}} [y_t \log \hat{p}_t + (1 - y_t) \log(1 - \hat{p}_t)]$ .

**Задача формирования сигналов.** Пусть  $g_t = \log \frac{\hat{p}_t}{1 - \hat{p}_t}$  — логит-оценка,  $a\theta_{long}, \theta_{short} > 0$  — пороги. Тогда торговый сигнал выглядит следующим образом:

$$s_t = \begin{cases} +1, & g_t \geq \theta_{long}, \\ -1, & g_t \leq -\theta_{short}, \\ 0, & \text{иначе,} \end{cases}$$

Размер позиции может быть выбран пропорционально «уверенности» модели:  $v_t = v_{max} \cdot \frac{|g_t|}{\theta_{max}} \cdot \mathbf{1}_{\{s_t \neq 0\}}$ , где  $\theta_{max}$  — технический ограничитель.

К системе предъявляются одновременно требования:

**Прогностическое качество.** Высокие значения AUC, F1, низкая лог-потеря, положительное математическое

ожидание сделок и коэффициент Шарпа на независимом периоде.

**Интерпретируемость.** Для каждой сделки  $(t, s_t)$  должна быть доступна декомпозиция вида  $g_t = g_0 + \sum_{j=1}^p \phi_j(x_{t,j})$ , где  $\phi_j(x_{t,j})$  — вклад  $j$ -го признака в логит, именно эта сумма должна использоваться роботом как есть, без скрытых преобразований.

**Прозрачное развёртывание модели (transparent deployment).** Используемая в работе модель должна совпадать с обученной моделью вплоть до значений таблиц  $f_j$  и  $f_{kl}$ , без дополнительного «ручного» кодирования формул.

**Стеклянные модели на основе обобщённых аддитивных моделей**

Классическая логистическая модель имеет вид

$$\hat{p}_t = \sigma \left( \beta_0 + \sum_{j=1}^p \beta_j x_{t,j} \right), \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

Она прозрачна: знак и величина  $\beta_j$  дают линейный эффект признака  $x_{t,j}$  на лог-шансы. Однако для финансовых рядов линейности часто недостаточно: зависимость доходности от, например, RSI носит насыщаемый нелинейный характер, а внутренняя структура шума требует учёта взаимодействий признаков.

**Обобщённые аддитивные модели (GAM)**

GAM-подход обобщает линейную модель, заменяя линейный эффект произвольной одномерной функцией:

$$g(\mathbb{E}[y|x]) = \beta_0 + \sum_{j=1}^p f_j(x_j) \quad (1)$$

где  $g(\cdot)$  — звено (для логистической регрессии  $g(p) = \log \frac{p}{1-p}$ ), а  $f_j$  — сплайны, деревья малой глубины и т.д.).

Функции  $f_j$  можно визуализировать в виде графиков «значение признака → вклад в логит», тем самым получая глобальные объяснения модели. При этом аддитивная структура обеспечивает строгую декомпозицию вклада по признакам.

Для задач, где важны парные взаимодействия (например, совместное поведение индикатора волатильности и трендового фильтра), вводится расширение  $GAM^2$ :

$$g(\mathbb{E}[y|x]) = \beta_0 + \sum_{j=1}^p f_j(x_j) + \sum_{k < l} f_{kl}(x_k, x_l) \quad (2)$$

**Explainable Boosting Machine как стеклянная модель**

Explainable Boosting Machine (EBM) реализует модель вида (2), но обучает функции  $f_j$  и  $f_{kl}$  с использованием циклической схемы градиентного бустинга по деревьям решений малой глубины и бэггинга (bootstrap aggregating).

На уровне оптимизации задача может быть записана как

$$\min_{\{f_j\}, \{f_{kl}\}} \frac{1}{N} \sum_t \ell \left( y_t, \sigma \left( \beta_0 + \sum_j f_j(x_{t,j}) + \sum_{k < l} f_{kl}(x_{t,k}, x_{t,l}) \right) \right) + \lambda \Omega(\{f_j\}, \{f_{kl}\}) \quad (3)$$

где  $\ell$  — логистическая потеря,  $\Omega$  — регуляризатор гладкости и разреженности (контролирует число активных взаимодействий),  $\lambda$  — параметр регуляризации.

На практике каждая функция  $f_j$  задаётся в виде табличной ступенчатой зависимости на равномерной или квантильной сетке значений признака,  $f_{kl}$  задаётся аналогично, но на двумерной сетке узлов  $(x_k, x_l)$ . При этом обучение идёт в циклах: каждое обновление изменяет одну из функций  $f_j$  или  $f_{kl}$ , что обеспечивает стабильность и независимость от порядка признаков.

Таким образом, для любого наблюдения  $x_t$  логит  $g_t = \beta_0 + \sum_j f_j(x_{t,j}) + \sum_{k < l} f_{kl}(x_{t,k}, x_{t,l})$  является строго аддитивной суммой компонент, и каждую компоненту можно трактовать как вклад соответствующего фактора в торговое решение. В отличие методов постфактум-анализа (LIME, SHAP), здесь объяснение не приближённое, а совпадает с прогнозом по определению.

**Архитектура интеграции модели в MQL5-советник**

Для обучения модели в среде Python необходимо загрузить исторические котировки, которые загружаются через API торгового терминала, формируется выборка  $\{(x_t, y_t)\}$  с учётом скользящего окна и лагов.

В качестве модели EBM используется реализация ExplainableBoostingClassifier из библиотеки interpret, ее настройки включают ограничение глубины базовых деревьев (обычно 2–3), ограничение числа взаимодействий  $f_{kl}$  и бэггинг по подвыборкам для повышения устойчивости.

Для валидации результатов на выделенном тестовом периоде рассчитываются AUC, Brier-score, precision/recall и торговые метрики (матожидание сделки, коэффициент Шарпа, максимальная просадка).

**Экспорт в формат ONNX**

Для переноса модели в торговый робот используется стандарт ONNX, поддерживаемый многими средами и языками, включая C++ и C#, что позволяет реализовать вызов из MQL5 через внешнюю DLL.

Используется специальная утилита *ebm2onnx*, реализующая преобразование обученной EBM в ONNX-граф. Она поддерживает бинарную классификацию, экспорт вероятностей и логитов, а также экспорт локальных объяснений, то есть вектор вкладов  $\phi_j(x_{t,j})$  как отдельный выход модели.

Таким образом, одно обращение к модели из советника возвращает как  $\hat{p}_t$ , так и набор компонент  $\{\phi_j\}$ , непосредственно используемых в журналировании.

В MQL5 реализуется следующий цикл:

- На каждом новом баре формируется вектор признаков  $x_t$ .
- Внешняя функция (через ONNX Runtime) возвращает  $\hat{p}_t$  и  $\phi_j(x_{t,j})$ .
- На основе  $g_t$  и выбранных порогов рассчитывается сигнал  $s_t$  и объём  $v_t$ .

Совершённые сделки логируются в файл с сохранением времени и параметров входа/выхода, значений  $g_t$ ,  $\hat{p}_t$  и топ- $k$  вкладов признаков  $\phi_{j_1}, \dots, \phi_{j_k}$ .

В качестве примера рассматривается валютная пара EURUSD. Выборка разделяется на обучающую часть (60%), валидационную (20%) для подбора гиперпараметров и тестовую (20%) для финального бэктеста.

Для избежания утечек информации разбиение производится по времени (walk-forward тестирование).

Для оценки качества прогнозов используются следующие метрики:

AUC ROC:  $AUC = \int_0^1 TPR(FPR^{-1}(u)) du$ , которая измеряет способность модели отделять растущие бары от падающих;

Brier-скор:  $Brier = \frac{1}{N} \sum_t (\hat{p}_t - y_t)^2$ , характеризующая калибровку вероятностей, а также используется логарифмическая потеря  $\mathcal{L}(\theta)$  из (3).

На тестовом интервале по сигналам модели строится кривая эквити (equity) с учётом комиссий и спреда. Рассматриваются:

- Математическое ожидание одной сделки

$$\mathbb{E}[R] = \frac{1}{M} \sum_{i=1}^M R_i^i$$

- Коэффициент Шарпа Sharpe =  $\frac{\mathbb{E}[R] - r_f}{\sigma_R} \sqrt{252}$ , где  $R_t$  — дневная доходность портфеля,  $r_f$  — безрисковая ставка,  $\sigma_R$  — стандартное отклонение дневной доходности;
- Максимальная относительная просадка;
- Доля прибыльных сделок и отношение средних выигрыша/проигрыша.

Предлагаются три вспомогательные метрики, учитывающие специфику торговых стратегий.

**Согласованность знаков эффектов.** Пусть  $J_{\text{prior}}$  — множество признаков, для которых заранее задано ожидаемое направление влияния (например, рост волатильности не должен увеличивать склонность к длинной позиции).

Определим

$$I_{\text{sign}} = \frac{1}{|J_{\text{prior}}|} \sum_{j \in J_{\text{prior}}} \mathbb{1}[\text{sign}(\Delta f_j(x_j)) = s_j^{\text{prior}}],$$

где  $\Delta f_j(x_j)$  измеряет изменение функции  $f_j$  при переходе от нижнего к верхнему квартилю распределения признака, а  $S_t = \{j : |\phi_j(x_{t,j})| \geq \alpha \cdot \max_k |\phi_k(x_{t,k})|\}$  — ожидаемый знак эффекта.

**Стабильность форм функций во времени.** Обозначим через  $f_j^{(1)}$  и  $f_j^{(2)}$  оценки функции  $f_j$ , обученные на двух непересекающихся отрезках истории.

Введём меру  $I_{\text{stab}}(j) = 1 - \frac{\|f_j^{(1)} - f_j^{(2)}\|_2}{\|f_j^{(1)}\|_2 + \|f_j^{(2)}\|_2}$ , усредняя

по всем узлам сетки признака. Значения, близкие к 1, свидетельствуют о структурной устойчивости зависимости.

**Оборот объяснений по сделкам.** Пусть  $T_{\text{trades}}$  — множество моментов открытия сделок. Для каждой сделки  $t$  определим множество «главных» признаков  $S_t = \{j : |\phi_j(x_{t,j})| \geq \alpha \cdot \max_k |\phi_k(x_{t,k})|\}$ , где  $\alpha \in (0, 1)$ .

Тогда  $I_{\text{turn}} = \frac{1}{|T_{\text{trades}}| - 1} \sum_{t_i, t_{i+1} \in T_{\text{trades}}} \left( 1 - \frac{|S_{t_i} \cap S_{t_{i+1}}|}{|S_{t_i} \cup S_{t_{i+1}}|} \right)$

измеряет долю сделок, в которых доминирующие факторы резко сменяются. Слишком высокий оборот может свидетельствовать о нестабильности стратегии.

### Результаты моделирования на исторических данных и их интерпретация

Использование стеклянной модели типа EBM в контексте валютного рынка даёт ряд преимуществ. Наличие

явных функций  $f_j$  позволяет выявлять неадекватные зависимости (например, «чем выше спред, тем больше вероятность входа»), отслеживать деградацию модели по мере смены рыночного режима через метрику стабильности  $I_{\text{stab}}$ , а также быстро локализовать причины аномальных серий убытков, анализируя вклад  $\phi_j$  по убыточным сделкам.

Архитектура «Python-EBM → ONNX → MQL5» отделяет исследовательскую часть от торговой, но при этом гарантирует бинарную идентичность модели: в советнике используются те же таблицы  $f_j$  и  $f_{kl}$ , что и в процессе обучения, без переписывания логики на MQL5.

Ограничения подхода связаны с необходимостью аккуратной инженерии признаков в терминале (чтобы они точно соответствовали тем, что использовались при обучении). Потенциально возможно снижение качества по сравнению с глубокими нейросетями на очень сложных задачах.

Для тестирования модели была выбрана валютная пара EUR/USD на временном масштабе M15 (один бар длиной 15 минут) за период с 2020 по 2025 года (Рис. 1). Полученные в ходе экспериментов результаты показали, что стеклянная модель с аддитивной структурой способна обеспечивать качество прогнозирования, сопоставимое с «чёрными ящиками». На ряде конфигураций наблюдалось небольшое снижение статистических метрик по сравнению с агрессивно настроенным градиентным бустингом, однако это снижение компенсировалось более устойчивым поведением на независимом периоде и меньшей чувствительностью к переобучению.

С точки зрения торговых результатов стратегия, основанная на стеклянной модели, продемонстрировала положительное математическое ожидание сделок и умеренную просадку при условии консервативного выбора плеча. Более важным, однако, оказался качественный эффект: наличие детализированных объяснений по каждой сделке существенно упростило разбор аномальных периодов и позволило выявить несколько систематических сценариев, при которых модель склонна ошибаться. В большинстве таких ситуаций ключевую роль играли признаки, связанные с переходом рынка из трендового состояния во флэт или наоборот.

Интерпретируемые метрики показали, что для основной части выборки знаки эффектов признаков соответствуют экономической логике. Например, рост внутрисуточной волатильности обычно снижал склонность модели к увеличению размера позиции, а усиление среднесрочного тренда повышало вероятность входа в направлении тренда.

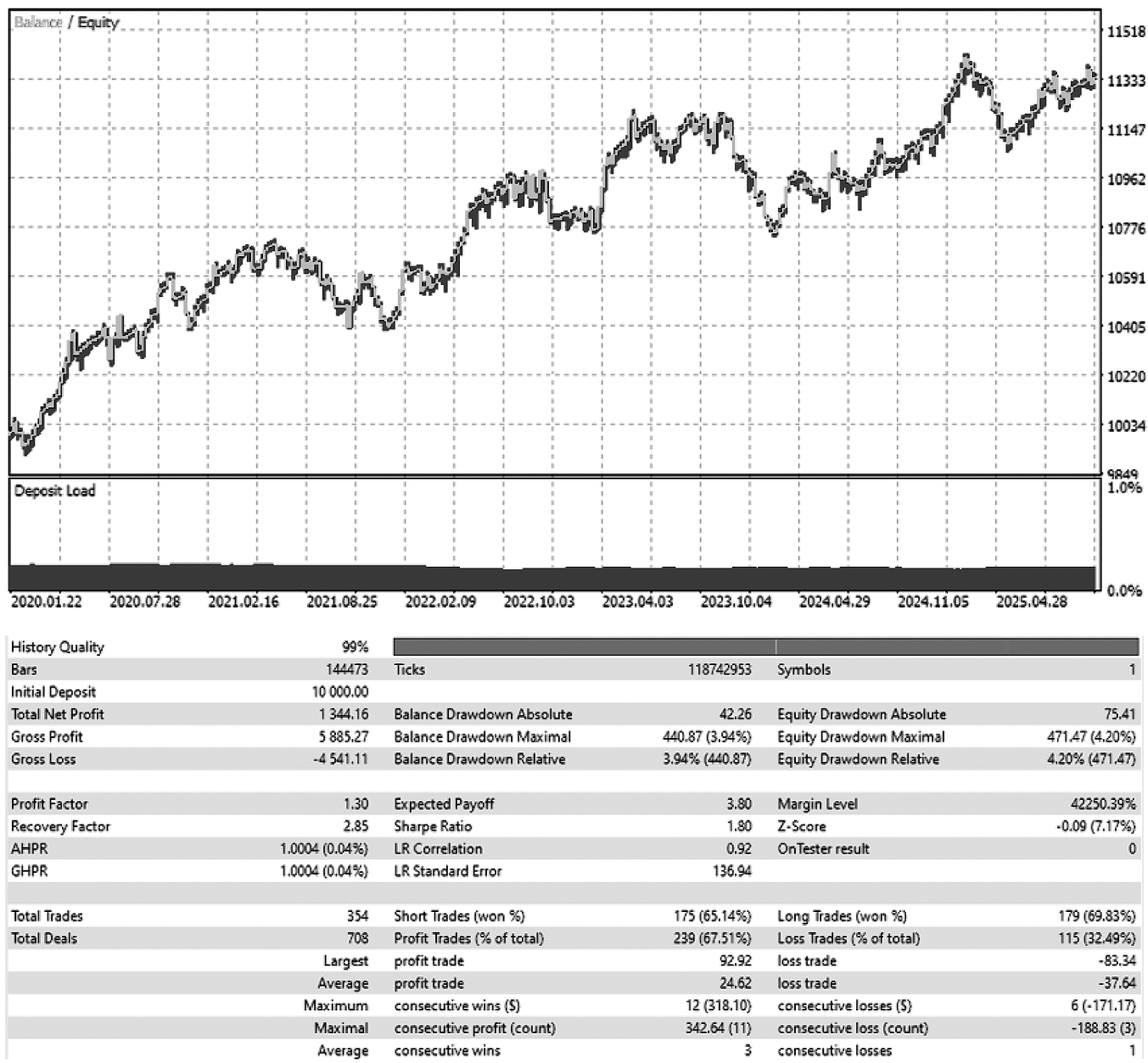


Рис. 1. Результаты тестирования на валютной паре EURUSD, 2020–2025 гг.

### Заключение

В работе предложен целостный подход к применению стеклянных аддитивных моделей в задачах алгоритмической торговли на валютном рынке. Показано, что обобщённые аддитивные модели вида (2), реализованные в форме Explainable Boosting Machine, позволяют совместить высокую прогностическую силу с строгой аддитивной декомпозицией вклада признаков в лог-шансы торгового решения. Описана практическая схема «Python-модель → экспорт в ONNX → MQL5-советник», реализующая перенос стеклянной модели в торговый терминал без потери прозрачности и без «ручного» перекодирования формул, что критично для надёжного промышленного применения.

Предложены специализированные показатели интерпретируемости для финансовых временных рядов, дополняющие стандартные статистические и торговые метрики качества.

Тестирование на исторических данных подтверждает применимость стеклянных моделей для реальных торговых задач и демонстрирует, что отказ от «чёрных ящиков» не обязательно ведёт к существенной потере доходности. Напротив, возможность видеть, почему автоматизированная торговая система совершает сделки, создаёт основу для систематического улучшения стратегий и более ответственного использования ИИ на финансовых рынках.

---

ЛИТЕРАТУРА

1. Hastie T., Tibshirani R. Generalized Additive Models. London: Chapman & Hall, 1990. 335 p.
2. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009. 745 p.
3. Friedman J.H. Annals of Statistics. 2001. Vol. 29. No. 5. Pp. 1189–1232.
4. Lou Y., Caruana R., Gehrke J., Hooker G. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2013. Pp. 623–631.
5. Caruana R., Lou Y., Gehrke J., Koch P., Sturm M., Elhadad N. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2015. Pp. 1721–1730.
6. Doshi-Velez F., Kim B. Communications of the ACM. 2017. Vol. 60. No. 10. pp. 56–64.
7. Molnar C. Interpretable Machine Learning. 2nd ed. Munich, 2022. 320 p.
8. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: MIT Press, 2016. 775 p.
9. Tsay R.S. Analysis of Financial Time Series. 3rd ed. Hoboken: Wiley, 2010. 677 p.
10. Chan E. Algorithmic Trading: Winning Strategies and Their Rationale. Hoboken: Wiley, 2013. 224 p.
11. De Prado M.L. Advances in Financial Machine Learning. Hoboken: Wiley, 2018. 696 p.
12. Murphy J.J. Technical Analysis of the Financial Markets. New York: New York Institute of Finance, 1999. 576 p.

---

© Климин Павел Юрьевич (k2x2p@yandex.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»