

Анализ вклада НДС в цены объектов недвижимости

Данные по рынку агломерации Санкт-Петербург на
май 2025

К. А. Мурашев¹

¹Sovconsult DOO
Founder

17 июня 2025 г.

Содержание I

- 1 Постановка задачи
- 2 Технологическая основа
 - Техпроцесс кратко
 - Применённые технологии
- 3 Этапы
 - Работа нейросети-автокодировщика
 - Дозаполнение пропущенных значений
 - Построение регрессионной модели
 - Построение случайного леса решающих деревьев регрессии
 - Градиентный бустинг XGBoost
 - Многослойный перцептрон

Содержание II

- 4 Технология расчёта вкладов признаков
 - Вектор Шепли

- 5 Результаты
 - Качество моделей
 - Рынок офисов
 - Рынок торговой недвижимости
 - Рынок складской недвижимости

Цель I

Определить вклад НДС различных уровней (5, 7, 20 %) на цены объектов недвижимости. Цена $\neq$ стоимость. Стоимость не зависит от НДС, но стоимость не является наблюдаемой величиной, это распределение, *оценку* которого можно получить методами анализа данных о *ценах* ofert и сделок.

Краткое описание проекта I

Разработка единой модели ценообразования коммерческой недвижимости и предсказания цен для предотвращения фрода в финансовой сфере. Сбор данных о ценах ofert и сделок с коммерческой недвижимостью в период с 2024-01-01 по 2025-04-30. Всего ≈ 65000 наблюдений. Регион интереса: Санкт-Петербург и Ленобласть. В частности необходимо найти противодействие классическому фроду “застраховать по “сметной стоимости” и сжечь выгоднее, чем продать на рынке”.

Техпроцесс кратко I

- Сбор данных: скрэппер для объявлений плюс базы сделок Росреестра.
- Очистка данных: удаление аномалий вроде множества сделок за 1 рубль.
- Построение единого уровня цен по сделкам на основе Double Machine Learning (DML), цены сделок и ofert подгоняются друг под друга. Очень отдалённо напоминает алгоритм на основе многомерного совместного распределения, описанный М. Б. Ласкиным. Используется Random Forest и разложение в ряды Тейлора для элиминации

Техпроцесс кратко II

ошибок, вызванных разницей в фичах между сделками и офертами.

- Новый подход к разбору неформализованной части. Вместо регулярных выражений теперь используется NLP. Токенизация позволяет получить фичи готовые для подачи в нейросеть-автокодировщик без участия человека.
- Обработка всех входных признаков нейросетью-автокодировщиком. Сокращение размерности пространства признаков.

Техпроцесс кратко III

- Подача “фич” в расчётную модель. Использование разбиения 70/30 train/test и такого же для валидации при тюнинге гиперпараметров моделей, где они есть (все кроме регрессионного анализа, в котором только кросс-валидация без тестирования).
- Расчёт на основе стека моделей.

Применённые технологии I

В 2023–2025 проводился перевод технологической базы с преимущественно регрессионных моделей через ансамбли деревьев к нейронным сетям. На данный момент использует стек моделей:

- Legacy model и бенчмарк: Гедоническая регрессия на базе смешанной модели: логарифмы непрерывных переменных, включая цены, индикаторные бинарные переменные на базе one-hot-encoding.
- Высокопроизводительные модели с быстрым обучением: Ансамбли решающих деревьев Random Forest (хорошо устойчив к переобучению) и XGBoost (хорошо убирает большие ошибки).

Применённые технологии II

- Нейросеть классической архитектуры: MLP (многослойный перцептрон).
- Экспериментальное применение: GNN (графовые нейронные сети). В теории могут превосходить MLP.
- Экспериментальное применение: KAN (интерпретируемая нейронная сеть Колмогорова – Арнольда). Может выдавать аналитическую формулу наподобие уравнения регрессии.

Для сокращения размерности (после токенизации было примерно 400 переменных) используется нейросеть-автокодировщик, относящаяся к классу моделей “обучения без учителя”. Данные сети способны сокращать размерность таким образом, при котором сохраняется максимум информации, и существует возможность достаточно точно восстановить исходные данные.

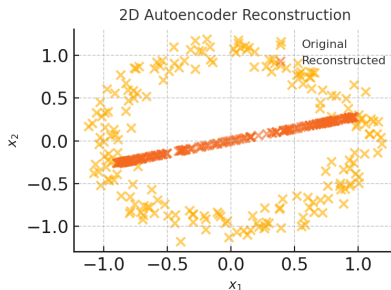


Рис. 1: Исходные данные до и после работы автокодировщика.

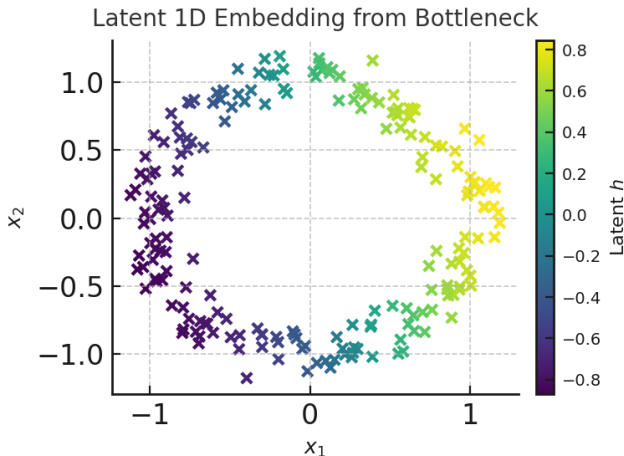


Рис. 2: Восстановленные данные.

Базовая архитектура нейросети-автокодировщика I

$$\begin{aligned} \textbf{Encoder: } \mathbf{h} &= f_{\text{enc}}(\mathbf{x}) = \sigma(W_{\text{enc}} \mathbf{x} + \mathbf{b}_{\text{enc}}), \\ \mathbf{x} \in \mathbb{R}^d &\longrightarrow \mathbf{h} \in \mathbb{R}^k, \quad k < d. \\ \textbf{Decoder: } \hat{\mathbf{x}} &= f_{\text{dec}}(\mathbf{h}) = W_{\text{dec}} \mathbf{h} + \mathbf{b}_{\text{dec}}, \quad \hat{\mathbf{x}} \in \mathbb{R}^d. \end{aligned} \tag{1}$$

$$\textbf{Loss (Reconstruction Error): } \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2.$$

Минимизация этой функции потерь заставляет сеть обучаться компактному представлению, которое

Базовая архитектура нейросети-автокодировщика II

сохраняет наиболее важную информацию, объясняющую вариативность в данных. Минимизация квадрата Евклидова расстояния на проекции подпространства меньшей размерности эквивалентна оценке максимального правдоподобия в рамках модели Гауссовского шума:

$$x = f_{dec}(f_{enc}(x)) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I). \quad (2)$$

Базовая архитектура нейросети-автокодировщика III

Реальные данные часто лежат на низкоразмерном латентном пространстве, встроенном в высокоразмерное пространство. Меньше переменных — лучше модель.
Задача: протащить данные через “бутылочное горлышко” так чтобы информация практически не потерялась.
Градиент (вектор первых частных производных рассматриваемой скалярной функции)

$$\nabla_{\hat{x}} \mathcal{L} = -2(x - \hat{x}) \quad (3)$$

Двигает $\hat{x}$ к x .

Дозаполнение пропущенных значений I

Используется MLP с тремя скрытыми слоями.

Realistic MLP Topology for Tabular Classification (41 inputs)

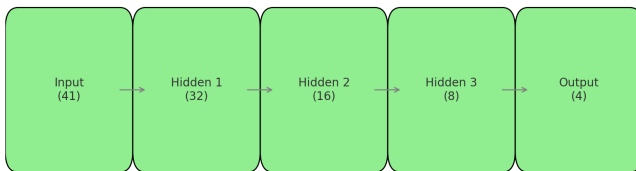


Рис. 3: Примерная топология нейросети для дозаполнения пропущенных значений.

Построение регрессионной модели I

Используется смешанная модель, в которой все непрерывные переменные логарифмируются, а бинарные остаются как есть.

$$\log Y = \log X_{1(\text{continuous})} + \log X_{2(\text{continuous})} + \dots + X_{n(\text{binary})} + \dots + \log \epsilon \quad (4)$$

Соответственно, первые являются эластичностями, т. е. отношением производных:

$$\frac{\partial \log Y}{\partial \log X_j} \quad (5)$$

Построение регрессионной модели II

Это позволяет давать прямую интерпретацию коэффициентов данной модели. Для бинарных переменных их влияние определяется как:

$$\exp(\beta X_j) - 1. \quad (6)$$

Регрессионные модели — хорошо проработанный инструмент. Однако они отлавливают только линейную связь, добиться которой часто можно только преобразованиями. В настоящее время существует ряд более продвинутых моделей. Поэтому регрессионные модели используются, в основном, в качестве бенчмарка относительно которого рассматриваются другие модели.

Решающие деревья I

Решающее дерево регрессии — это алгоритм машинного обучения, используемый для предсказания числовых значений целевой переменной. Он работает путём последовательного разделения входных данных на основе признаков и построения дерева, где каждый узел представляет условие, а лист представляет предсказанное значение. Разделение происходит бинарно, т.е. каждый раз на две части так, чтобы отличие целевого признака было минимальным в каждом новом листе. Для построения дерева не требуется глобальный алгоритм оптимизации, вместо этого используется локальный жадный алгоритм, целью которого является

Решающие деревья II

минимизация дисперсии, что эквивалентно минимизации MSE.

$$MSE = \frac{1}{S} \sum_{i \in S} (y_i - \bar{y}_S)^2 \quad (7)$$

$$\delta MSE(S) = \frac{S_L}{S} MSE(S_L) - \frac{S_R}{S} MSE(S_R) \quad (8)$$

Выбирается разбиение с максимальной дельтой.

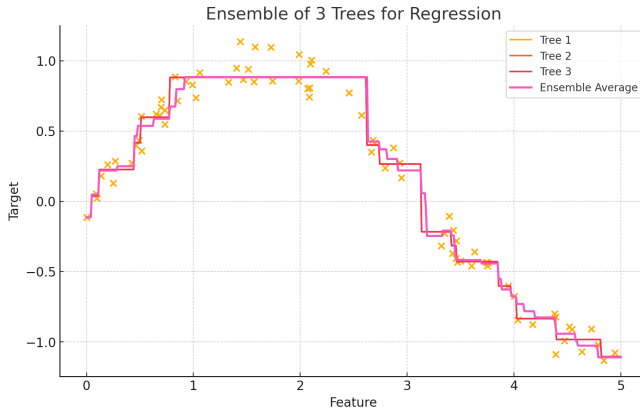


Рис. 4: Дерево регрессии аппроксимирует полином.

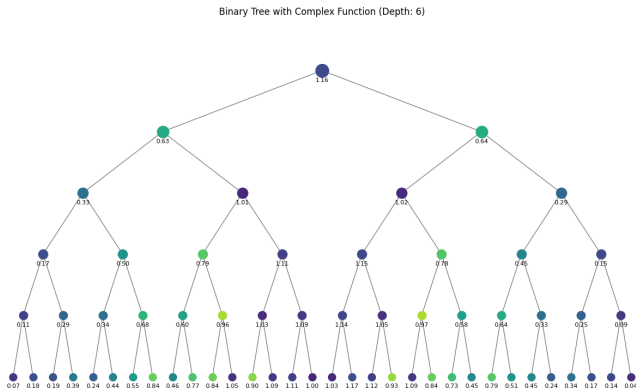


Рис. 5: Дерево регрессии в графовом виде.

Каждое дерево даёт 2^n прямых отрезков, где n — разрешённая максимальная глубина дерева. Алгоритм выявляет нелинейные и не монотонные связи и не зависит от распределений. Недостатком является высокая склонность к переобучению, т. е. чрезмерной подгонки под обучающие данные.

Случайный лес I

Случайный лес (Random Forest) — это ансамбль из большого числа относительно упрощённых и “недообученных” по отдельности деревьев. Каждое такое “карликовое дерево” получает случайную часть данных $n^* = n(1 - \frac{1}{e})$ и признаков $k^* = \sqrt{k}$. Дерево дорастает до критерия остановки. За счёт большого числа таких деревьев итоговый результат получается существенно лучше, чем у отдельного большого дерева. Алгоритм объединяет идеи бэггинга и вложенных подпространств.

Случайный лес II

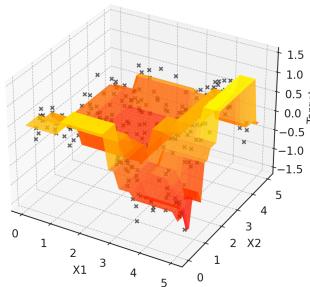
Лес проводит голосование и усредняет свои результаты. Итоговый результат получается значительно стабильнее. Изначально считалось, что алгоритм в целом не подвержен переобучению. В дальнейшем было показано, что такой риск есть, но он относительно небольшой. Недостатком алгоритма является то, что плохие деревья также участвуют в голосовании. Также алгоритм иногда может давать перевес в пользу непрерывных переменных против бинарных.

$$\hat{f}_{\text{RF}}(x) = \frac{1}{B} \sum_{b=1}^B T_b(x), \quad (9)$$

Случайный лес III

- $\hat{f}_{\text{RF}}(x)$ — предсказание леса для объекта X .
- $T_b(x)$ — предсказание b -го дерева.
- B — общее число деревьев.

Decision Tree (depth=5)



Random Forest (100 trees)

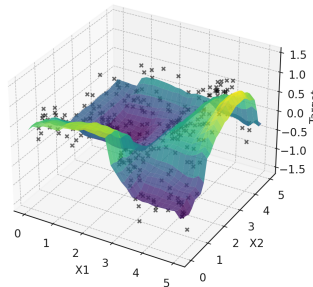


Рис. 6: Случайный лес даёт более сглаженную аппроксимацию.

Градиентный бустинг XGBoost I

Бустинг в отличие от бэггинга предполагает не параллельное, а последовательное построение ансамбля слабых моделей деревьев. Идея бустинга отчасти схожа с идеей разложения в ряд Тейлора. “Классические” исполнения бустинга — ADA Boost и GBM.

XGBoost — оптимизированный пакет для градиентного бустинга. Последовательное построение ансамбля подгоняет новые деревья по ошибкам текущего ансамбля, используя первые и вторые производные ошибок для более точного приближения. XGBoost содержит сотни улучшений относительно

Градиентный бустинг XGBoost II

GBM. В настоящее время один из лучших алгоритмов машинного обучения.

Многослойный перцептрон I

Сеть прямого распространения сигнала (Feed Forward) и обратного распространения ошибки (Back Propagation).
Используется функция активации ReLU.

$$ReLU(x) = \max(0, x) \quad (10)$$

$$ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (11)$$

$$ReLU'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (12)$$

Для расчёта вклада признаков в т. ч. НДС использовался Вектор Шепли — понятие из кооперативной теории игр. Значение вектора Шепли — это средний предельный вклад признака в стоимость.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)] \quad (13)$$

$$\phi_i = \mathbb{E}_{S \subseteq N \setminus \{i\}} [f(x_{S \cup \{i\}}) - f(x_S)] \quad (14)$$

$$\sum_{i=1}^n \phi_i = f(x) - \mathbb{E}[f(x)] \quad (15)$$

$$\phi_i^w = \sum_{S \subseteq N \setminus \{i\}} w(S) [v(S \cup \{i\}) - v(S)] \quad (16)$$

Качество моделей I

Таблица 1: Метрики качества моделей

Модель	MAPE (Средняя абс. ошибка на тестовой подвыборке)
Гедоническая регрессия	19.68 %
Случайный лес	12.85 %
XGBoost	10.77 %
MLP	8.11 %

Влияние НДС на цены продажи I

Параметризуется примерно нормальным распределением.

Влияние НДС на цены продажи II

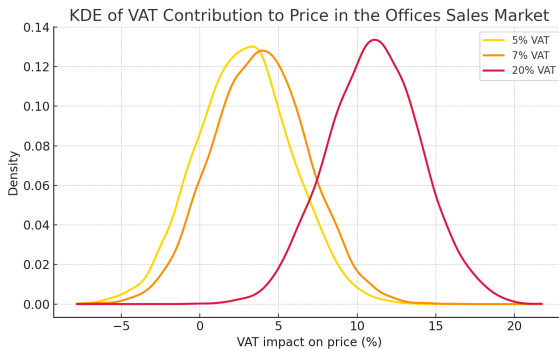


Рис. 7: Распределение вклада НДС в стоимость (продажа).

Влияние НДС на цены продажи III

Таблица 2: VAT Impact Distribution Summary (Office Sale Market)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	2.83	3.00	-3.18	8.84
7%	3.85	3.02	-2.20	9.89
20%	11.06	2.96	5.14	16.99

Влияние НДС на цены продажи IV

Вклад НДС не одинаковый для разных объектов. Можно очень упрощённо показать связь его вклада со стоимостью самого объекта. P — цена в млн рублей.

$$VAT_{20\%_surcharge} = 5 + 9 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (17)$$

$$VAT_{7\%_surcharge} = 1.75 + 3.15 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (18)$$

$$VAT_{5\%_surcharge} = 1.25 + 2.25 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (19)$$

Влияние НДС на цены продажи V

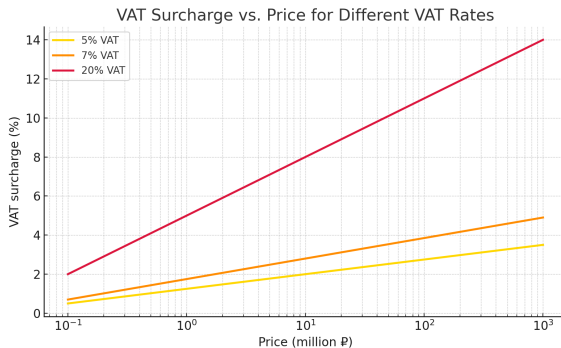


Рис. 8: Вклад НДС в цены продажи офисов как *сильно упрощённая* функция цены.

Влияние НДС на цены аренды I

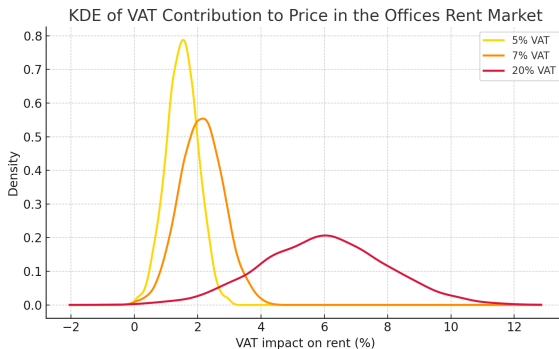


Рис. 9: Распределение вклада НДС в стоимость (аренда).

Влияние НДС на цены аренды II

Таблица 3: VAT Impact Distribution Summary (Office Rent Market)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	1.51	0.50	0.51	2.51
7%	2.12	0.70	0.73	3.51
20%	5.98	1.99	2.01	9.95

Влияние НДС на цены аренды III

R — ставка аренды за 1 кв. м.

$$VAT_{20\%_surcharge} = 0.05 + 0.11 \times \left(\frac{\log_{10} \frac{R}{800}}{5} \right) \quad (20)$$

$$VAT_{7\%_surcharge} = 0.0175 + 0.0385 \times \left(\frac{\log_{10} \frac{R}{800}}{5} \right) \quad (21)$$

$$VAT_{5\%_surcharge} = 0.0125 + 0.0275 \times \left(\frac{\log_{10} \frac{R}{800}}{5} \right) \quad (22)$$

Влияние НДС на цены аренды IV

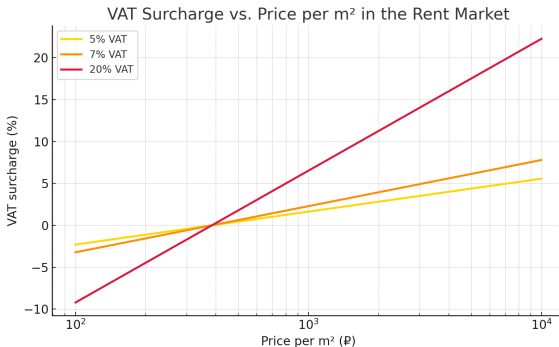


Рис. 10: Вклад НДС в цены аренды офисов как *сильно упрощённая* функция цены.

Влияние НДС на цены продажи I

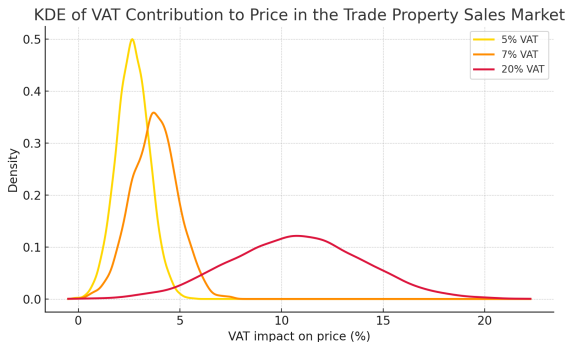


Рис. 11: Распределение вклада НДС в стоимость (продажа).

Влияние НДС на цены продажи II

Таблица 4: VAT Impact Distribution Summary (Trade Property Sales)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	2.68	0.81	1.06	4.30
7%	3.75	1.12	1.50	6.00
20%	10.77	3.27	4.23	17.30

Влияние НДС на цены продажи III

P — цена в млн руб.

$$VAT_{20\%_surcharge} = 7.5 + 7.5 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (23)$$

$$VAT_{7\%_surcharge} = 2.63 + 2.63 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (24)$$

$$VAT_{5\%_surcharge} = 1.88 + 1.88 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (25)$$

Влияние НДС на цены продажи IV

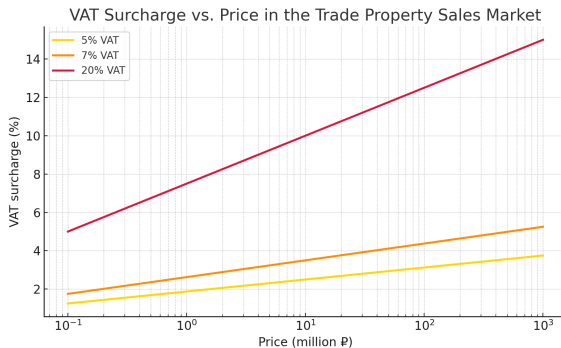


Рис. 12: Вклад НДС в цены продажи торговой недвижимости как *сильно упрощённая* функция цены.

(Влияние НДС на цены аренды)

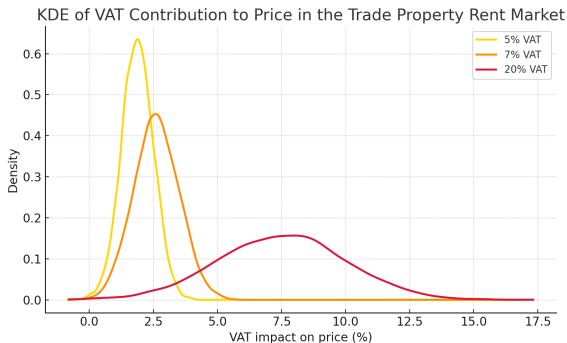


Рис. 13: Распределение вклада НДС в стоимость (аренда).

Таблица 5: VAT Impact Distribution Summary (Trade Property Rent Market)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	1.88	0.62	0.63	3.13
7%	2.62	0.88	0.85	4.38
20%	7.51	2.49	2.53	12.50

R — ставка аренды за 1 кв. м.

$$VAT_{20\%_surcharge} = 5 + 10 \times \left(\frac{\log_{10} \frac{R}{500}}{\log_{10} 30} \right) \quad (26)$$

$$VAT_{7\%_surcharge} = 1.75 + 3.5 \times \left(\frac{\log_{10} \frac{R}{500}}{\log_{10} 30} \right) \quad (27)$$

$$VAT_{5\%_surcharge} = 1.25 + 2.5 \times \left(\frac{\log_{10} \frac{R}{500}}{\log_{10} 30} \right) \quad (28)$$

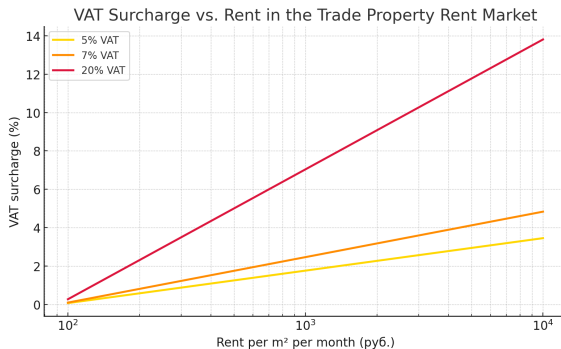


Рис. 14: Вклад НДС в цены аренды торговой недвижимости как *сильно упрощённая* функция цены.

Влияние НДС на цены продажи I

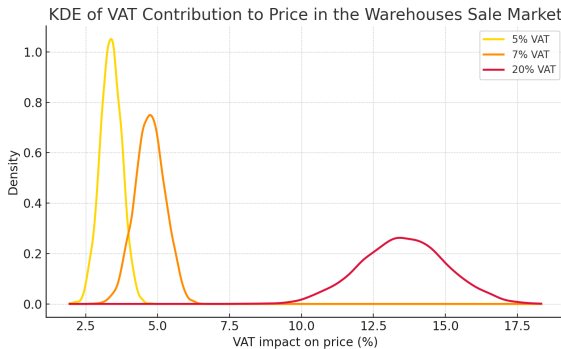


Рис. 15: Распределение вклада НДС в стоимость (продажа).

Влияние НДС на цены продажи II

Таблица 6: VAT Impact Distribution Summary (Warehouses Sale Market)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	3.38	0.38	2.62	4.14
7%	4.73	0.52	3.69	5.77
20%	13.54	1.49	10.56	16.52

Влияние НДС на цены продажи III

P — цена в млн руб.

$$VAT_{20\%_surcharge} = 12 + 6 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (29)$$

$$VAT_{7\%_surcharge} = 4.2 + 2.1 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (30)$$

$$VAT_{5\%_surcharge} = 3 + 1.5 \times \frac{\log_{10} P}{\log_{10} 1000} \quad (31)$$

Влияние НДС на цены продажи IV

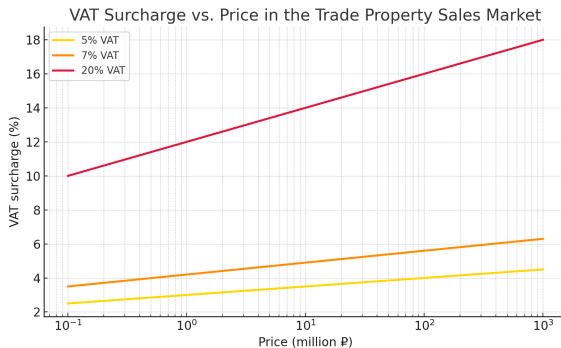


Рис. 16: Вклада НДС в цены продажи складской недвижимости как *сильно упрощённая* функция цены.

Влияние НДС на цены аренды I

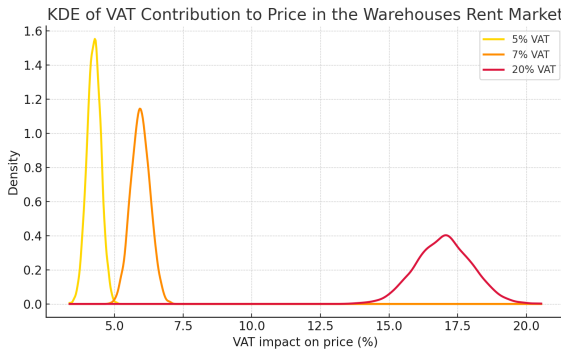


Рис. 17: Распределение вклада НДС в стоимость (аренда).

Влияние НДС на цены аренды II

Таблица 7: VAT Impact Distribution Summary (Warehouses Rent Market)

VAT Rate	Mean (%)	Std Dev (%)	Lo 2σ Bound (%)	Up 2σ Bound (%)
5%	4.25	0.25	3.75	4.75
7%	5.94	0.35	5.24	6.64
20%	17.00	1.01	14.98	19.02

Влияние НДС на цены аренды III

R — ставка аренды за 1 кв. м.

$$VAT_{20\%_surcharge} = 10 + 8 \times \left(\frac{\log_{10} \frac{R}{800}}{\log_{10} 5} \right) \quad (32)$$

$$VAT_{7\%_surcharge} = 3.5 + 2.8 \times \left(\frac{\log_{10} \frac{R}{500}}{\log_{10} 5} \right) \quad (33)$$

$$VAT_{5\%_surcharge} = 2.5 + 2 \times \left(\frac{\log_{10} \frac{R}{500}}{\log_{10} 5} \right) \quad (34)$$

Влияние НДС на цены аренды IV

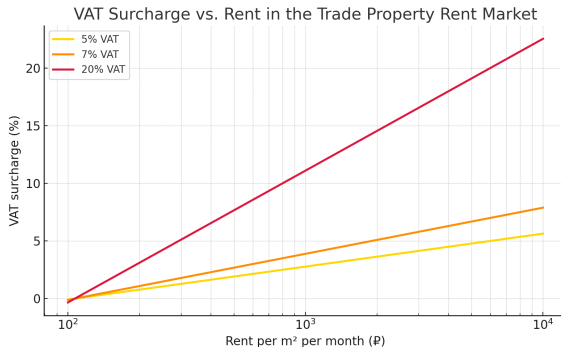


Рис. 18: Вклад НДС в цены аренды складской недвижимости как *сильно упрощённая* функция цены.