www.volsu.ru

DOI: https://doi.org/10.15688/mpcm.jvolsu.2025.1.3

УДК 004.8
ББК 32.973

Дата поступления статьи: 24.02.2025
Дата принятия статьи: 13.03.2025

ПОСТРОЕНИЕ МОДЕЛИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ РАССУДИТЕЛЬНОГО ТЕКСТА

Александр Владимирович Каныгин

Аспирант кафедры компьютерных наук и экспериментальной математики,
Волгоградский государственный университет
a.kanygin99@gmail.com
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. В статье рассмотрена задача классификации текстов на предмет наличия в них рассуждений (логических связок, аргументации, причинно-следственных отношений). Цель исследования — разработать метод, позволяющий с высокой точностью определять «рассудительный» характер фрагмента текста, используя современные алгоритмы машинного обучения. Особое внимание уделено ансамблевому подходу на основе стекинга: в качестве базовых классификаторов рассматриваются сильные модели (CatBoost, XGBoost, Random Forest и т. п.), а роль мета-модели выполняет логистическая регрессия. Для обоснования выбора стекинга приводятся результаты сравнительного анализа более десяти популярных алгоритмов (Logistic Regression, SVC, Random Forest, CatBoost, XGBoost и др.) по показателям Accuracy, Precision, Recall, F1-score, ROC AUC, PR AUC. Основные этапы исследования включают генерацию и разметку обучающего набора данных, предварительную обработку текстов (токенизацию, лемматизацию, исключение стоп-слов), векторизацию признаков (TF-IDF) и экспериментальное сравнение моделей на контрольной выборке. Предложенная модель стекинга показала лучшие результаты по совокупности метрик, что позволило повысить точность классификации рассудительных текстов до уровня F1, равного 0,905, при ROC AUC, равному 0,887. В заключении обсуждаются перспективы применения описанного подхода для текстов разной длины и стиля, а также потенциальные методы дальнейшего улучшения качества классификации.

Ключевые слова: машинное обучение, ансамблевые методы, стекинг, TF-IDF, аргументация, анализ текстовых данных.

Введение

Актуальность задачи автоматического распознавания рассуждений в текстах определяется стремительным ростом объема цифровых данных и необходимостью их интеллектуальной обработки [1]. В различного рода дискуссиях, научных статьях, отзывах и эссе наличие аргументации и причинно-следственных связей часто определяет ценность и структуру материала. Анализ рассуждений требуется в таких областях, как интеллектуальный поиск, построение диалоговых систем, экспертные системы, образовательные онлайн-платформы и т. д. Существуют различные подходы к решению задач автоматической классификации текстов на основе их семантических, синтаксических и статистических особенностей. Традиционно применяют методы наивной Байесовской классификации, решающие деревья, опорные векторы (SVM) или логистическую регрессию. Однако в последние годы большую популярность получили более сложные ансамблевые алгоритмы (Random Forest, Gradient Boosting, XGBoost, CatBoost), а также глубокие нейронные сети и трансформеры (BERT, GPT и их вариации) [3; 5]. Несмотря на это, оптимальный выбор модели во многом зависит от конкретного корпуса текстов и требуемых метрик.

В контексте данной работы мы рассматриваем задачу «binary classification» (есть рассуждение или нет рассуждения) для русскоязычных фрагментов. Первичным этапом стал сбор и разметка данных: часть текстов была сгенерирована и размечена автоматически, а часть получена из открытых источников, в которых вручную выделены участки с аргументацией. На этапе предварительной обработки были выполнены лемматизация и удаление стоп-слов, а признаки формировались методом TF-IDF. Далее, чтобы провести сопоставительный эксперимент, мы обучили и оценили целый ряд популярных моделей. Наиболее интересные результаты показали XGBoost, CatBoost, а также логистическая регрессия с оптимизированным параметром регуляризации. В результате, для достижения более высокой точности, был предложен метод стекинга (Stacking), в котором указанные алгоритмы выступают в роли базовых классификаторов, а итоговое решение принимается мета-моделью (логистической регрессией).

Новизна исследования заключается в применении сложного ансамблевого подхода (Stacking) для решения задачи автоматического определения рассуждений в тексте и сравнении его эффективности с рядом одиночных и ансамблевых моделей. Экспериментальные результаты показали, что подобная схема обеспечивает прирост качества по метрикам Precision, Recall, F1-score, ROC AUC и PR AUC по сравнению с каждой из моделей, взятой по отдельности.

1. Математическая формализация задачи

Пусть задан корпус текстов $\{T_i\}_{i=1}^{N}$, где каждый текст T_i описывается своим содержимым (набором предложений или абзацев) и после предобработки отображается в вектор признаков $x_i \in X$. Цель — определить для каждого текста двоичную метку $y_i \in \{0, 1\}$, где $y_i = 1$ указывает на то, что в тексте присутствуют рассуждения (аргументация, причинно-следственные связи, логические выводы), а $y_i = 0$ — их отсутствие. Для решения задачи требуется построить классификатор $f(\cdot)$, обладающий максимальными показателями точности (Precision, Recall, F1-score, ROC AUC и PR AUC), то есть

$$\hat{y}_i = f(x_i) \quad \text{при} \quad f : X \to \{0, 1\}, \tag{1}$$

где X — пространство признаков, сформированных из текстов после предобработки. На этапе предобработки для каждого T_i выполняются:

- токенизация и (опционально) лемматизация — приведение слов к начальной форме, исключение стоп-слов;
- векторизация TF-IDF: каждому тексту T_i сопоставляется вектор $x_i \in \mathbb{R}^M$, где M — размерность словаря,

$$w_{(i,j)} = \mathrm{tf}_{ij} \times \log \frac{N}{\mathrm{df}_i}, \tag{2}$$

где $w_{(i,j)}$ — значение TF-IDF для термина i в документе j; tf_{ij} — term frequency (количество вхождений термина i в документе j); df_i — document frequency (число документов, содержащих термин i); N — общее число документов в корпусе. Следует отметить, что координаты $w_{(i,j)}$ являются компонентами вектора признаков $x_j \in \mathbb{R}^M$, то есть

$$x_j = \big(w_{(1,j)},\, w_{(2,j)},\, \ldots,\, w_{(M,j)}\big),$$

соответствующего тексту T_j.

Таким образом, исходный текст T_i переходит в вектор x_i, на основе которого строится классификатор.

Ниже кратко описаны математические формулы для наиболее часто используемых алгоритмов, участвующих в стекинге:

1) **Логистическая регрессия.** Предположим, что имеется обучающая выборка $\{(x_i, y_i)\}_{i=1}^N$, где $x_i \in \mathbb{R}^M$. Логистическая регрессия ищет вектор весов w и смещение b, минимизируя функцию потерь:

$$L(w, b) = -\sum_{i=1}^{N} \Big[y_i \log \sigma(w^T x_i + b) + (1 - y_i) \log\big(1 - \sigma(w^T x_i + b)\big) \Big], \tag{3}$$

где $\sigma(\cdot)$ — сигмоидальная функция $\sigma(z) = \frac{1}{1+e^{-z}}$, принимающая значения в интервале (0, 1).

2) **Методы бустинга (XGBoost, CatBoost).** Идея состоит в последовательном построении композиции деревьев решений. Общая формула ошибки может быть представлена как

$$L(F) = \sum_{i=1}^{N} l(F_k(x_i), y_i) + \Omega(F_k), \tag{4}$$

где $F_k(x) = \sum_{j=1}^{k} h_j(x)$, а $h_j(\cdot)$ — деревья решений на разных шагах. Функционал $\Omega(\cdot)$ отвечает за регуляризацию и сложность модели.

3) **Случайный лес (Random Forest).** Строит ансамбль из L независимых деревьев решений $\{h_l(x)\}$, усредняя их ответы:

$$F(x) = \frac{1}{L}\sum_{l=1}^{L} h_l(x). \tag{5}$$

В работе применяется двухуровневый стекинг (Stacking). Пусть имеется набор базовых (Level-1) моделей $\{M_j\}_{j=1}^{K}$. Каждая модель M_j выдает предсказанные вероятности

$$p_{ij} = M_j(x_i), \tag{6}$$

после чего формируется вектор

$$z_i = (p_{i1}, p_{i2}, \ldots, p_{iK}), \tag{7}$$

который поступает на вход мета-модели $G(\cdot)$. Обычно в роли G выступает линейный или деревоподобный классификатор. В нашем случае мета-моделью выбрана логистическая регрессия:

$$\hat{y}_i = \mathrm{round}\Big[\sigma\big(\alpha^T z_i + \beta\big)\Big], \tag{8}$$

где α и β — параметры мета-классификатора, оцениваемые на валидационной подвыборке или с помощью кросс-валидации, а $\mathrm{round}(\cdot)$ обозначает математическое округление до ближайшего целого. При реализации на C++ необходимо использовать, например, `std::round`, поскольку простое приведение `(int)double` не эквивалентно округлению, а лишь отбрасывает дробную часть.

Примечание. Подробное обоснование выбора логистической регрессии в качестве метамодели представлено в следующей части статьи; здесь лишь подчеркнем, что линейная модель с L_2-регуляризацией хорошо устраняет переобучение и находит оптимальную «смешанную» вероятность на выходе из базовых классификаторов.

На рисунке 1 представлена общая схема (блок-схема) формирования обучающего набора данных: генерация текстов, разметка (автоматическая и ручная), предобработка и формирование выборки.

2. Экспериментальная часть

Общий экспериментальный пайплайн (блок-схема) представлен на рисунке 2. На вход подаются уже размеченные тексты $\{T_i, y_i\}$, прошедшие процедуру предобработки. Каждый текст преобразуется в вектор признаков x_i посредством TF-IDF. Далее применяется несколько моделей (Logistic Regression, SVC, Random Forest, CatBoost, XGBoost и т. д.) для обучения и валидации, и, наконец, результаты оцениваются по набору метрик (Accuracy, Precision, Recall, F1-score, ROC AUC, PR AUC).

Для экспериментов был использован корпус из 963 текстовых фрагментов, в каждом из которых вручную или автоматически (с последующей проверкой) определена двоичная метка: «есть рассуждение» (1) или «нет рассуждения» (0). Тексты варьировались по тематике и объему, чтобы обеспечить необходимую разнообразность. На обучающую выборку было отведено 80 % корпуса, а на тестовую — оставшиеся 20 %. Следует отметить, что выборка не является сбалансированной по классам: доля позитивных примеров

(«есть рассуждение») составляет 720 примеров, в то время как метку «нет рассуждения» имеют 243 примера, что обусловлено особенностями разметки и формирования корпуса.

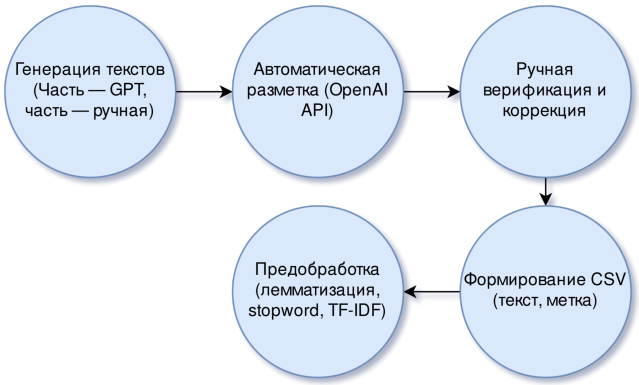

Рис. 1. Пример блок-схемы формирования корпуса (генерация текстов, автоматическая и ручная разметка, предобработка)

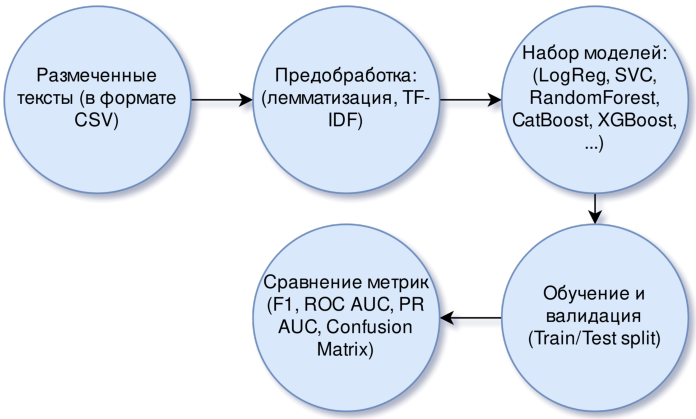

Рис. 2. Структурная схема эксперимента, показывающая последовательность шагов от размеченных данных до оценки результатов

Таким образом, утверждать, что основная часть текстов в общей совокупности носит рассудительный характер, нельзя, поскольку корпус создавался с учетом специфики задачи. Во всех опытах использовался единый `random_state=42`, что гарантировало воспроизводимость результатов. Перед подачей на вход моделям каждый текст прошел стадию очистки и лемматизации, после чего векторизовывался методом TF-IDF, формируя пространственные признаки фиксированной размерности (до 5 000 компонент). В качестве алгоритмов классификации рассматривались логистическая регрессия с подбором гиперпараметра C (Logistic Regression (GS)), XGBoost, SVC, Gradient Boosting, CatBoost, Random Forest, MLP, Extra Trees, KNN, Naive Bayes и Decision Tree. Для каждой модели подсчитывались Accuracy, Precision, Recall, F1-score, ROC AUC и PR AUC, что соответствует подходам, описанным в работе [4]. Все эксперименты проводились в идентичных условиях, что позволило объективно оценивать возможности разных подходов.

На рисунке 3 показаны столбиковые диаграммы по трем ключевым метрикам (F1-score, PR AUC и ROC AUC). Из диаграммы видно, что Logistic Regression (GS), XGBoost и SVC набирают наиболее высокие значения F1 (от 0,90 до 0,91), что согласуется с результатами, опубликованными в [3] и [6].

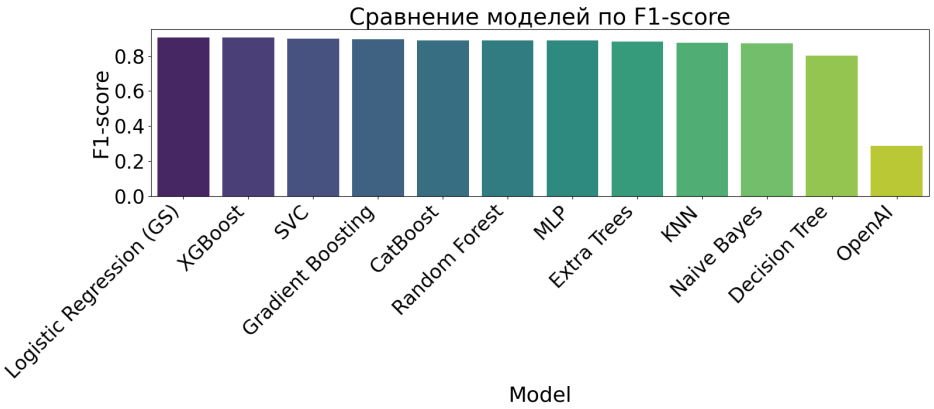

Рис. 3. Сравнение моделей по F1-score

Gradient Boosting, CatBoost и Random Forest лишь незначительно отстают. Модели Extra Trees, KNN, Naive Bayes и Decision Tree занимают нижние строки рейтинга, их F1 не превышает 0,88.

Аналогичная картина наблюдается при сравнении моделей по PR AUC (рис. 4): лучшие результаты снова дают Logistic Regression (GS), XGBoost и частично CatBoost, демонстрируя AUC около 0,94–0,95. Чуть ниже располагаются SVC и Gradient Boosting.

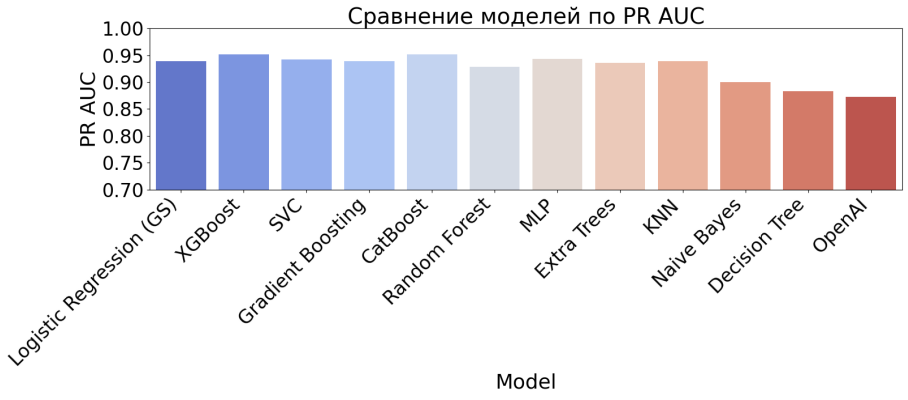

Рис. 4. Сравнение моделей по PR-AUC

В общем случае PR AUC высок у всех алгоритмов, поскольку доля позитивных примеров (рассудительных текстов) достаточно велика и качество вероятностной оценки остается на хорошем уровне. Сходные тенденции наблюдаются и на диаграмме ROC AUC (см. рис. 5): лидируют XGBoost, Logistic Regression (GS) и SVC, достигая 0,88–0,89, затем следуют CatBoost, Random Forest и MLP, а заметно слабее выступают KNN, Naive Bayes и Decision Tree.

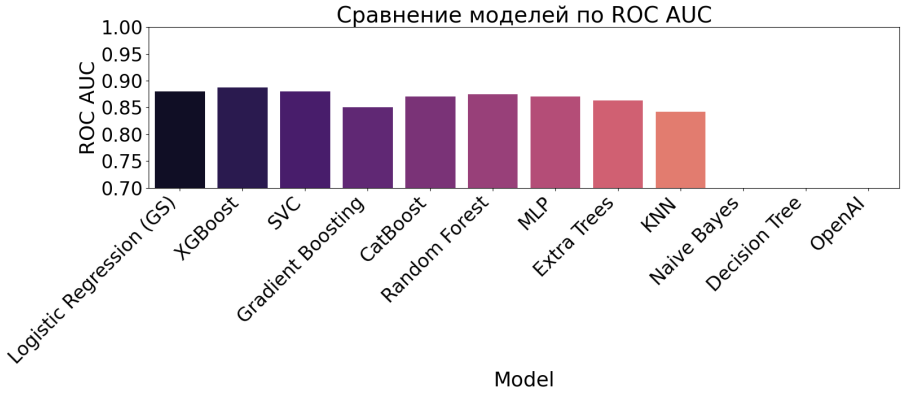

Рис. 5. Сравнение моделей по ROC-AUC

Таблица 1

Результаты автоматической идентификации медийных текстов с применением методов машинного обучения

Модель	Accuracy	Precision	Recall	F1-score	ROC AUC	PR AUC
Logistic Regression (GS)	0,854922	0,866242	0,951048	0,906667	0,879580	0,939617
XGBoost	0,854922	0,875817	0,937063	0,905405	0,886993	0,952278
SVC	0,839378	0,837348	0,972028	0,899676	0,860000	0,942624
Gradient Boosting	0,839378	0,863636	0,930070	0,895623	0,850769	0,939090
CatBoost	0,829016	0,852564	0,930070	0,889632	0,870628	0,951492
Random Forest	0,823834	0,830303	0,958042	0,889610	0,874266	0,929086
MLP	0,829016	0,857143	0,923077	0,888889	0,870350	0,943208
Extra Trees	0,813472	0,854305	0,902098	0,861578	0,863497	0,935847
KNN	0,813472	0,854305	0,902098	0,877851	0,842308	0,939657
Naive Bayes	0,797827	0,817073	0,937063	0,872964	0,668531	0,900384
Decision Tree	0,715026	0,823529	0,783217	0,802867	0,651608	0,863684
Open AI GPT 4	0,373057	0,958333	0,347506	0,275449	0,570420	0,870467

Сводные числовые результаты всех одиночных моделей, упорядоченные по возрастанию F1-score, приведены в таблице 1. Из нее следует, что Logistic Regression (GS) показывает высочайший уровень F1 (0,9067), совсем близко к ней идет XGBoost. Показательно, что часть алгоритмов (например, Random Forest и CatBoost) имеют более высокое значение Recall, но при этом уступают по Precision, что снижает их итоговый F1.

На рисунке 6 представлены ROC- и PR-кривые для всех одиночных моделей. Кривая Logistic Regression (GS) (AUC = 0,88) пролегает выше, чем у Decision Tree (AUC = = 0,65) и Naive Bayes (AUC = 0,67). Вблизи лидеров находятся XGBoost (0,89) и SVC (0,88). CatBoost (0,87) демонстрирует сравнимый уровень, но несколько уступает XGBoost по наклону кривой на отрезке 0,0–0,2 FPR. Также на рисунке видны PR-кривые, где наиболее высокие показатели наблюдаются у XGBoost (AUC = 0,95), CatBoost (AUC = 0,95) и Logistic Regression (GS) (AUC = 0,94). Модели, которые на ROC-кривой оказались в числе средних (Extra Trees, MLP, Gradient Boosting), в PR-пространстве располагаются достаточно близко к лидерам.

На рисунке 7 изображены ROC- и PR-кривые для модели стекинга (Stacking). Площадь под ROC достигает 0,89, а под PR — 0,95. Таким образом, суммарное качество наглядно превосходит средний уровень большинства одиночных методов. Вся ROC-кривая

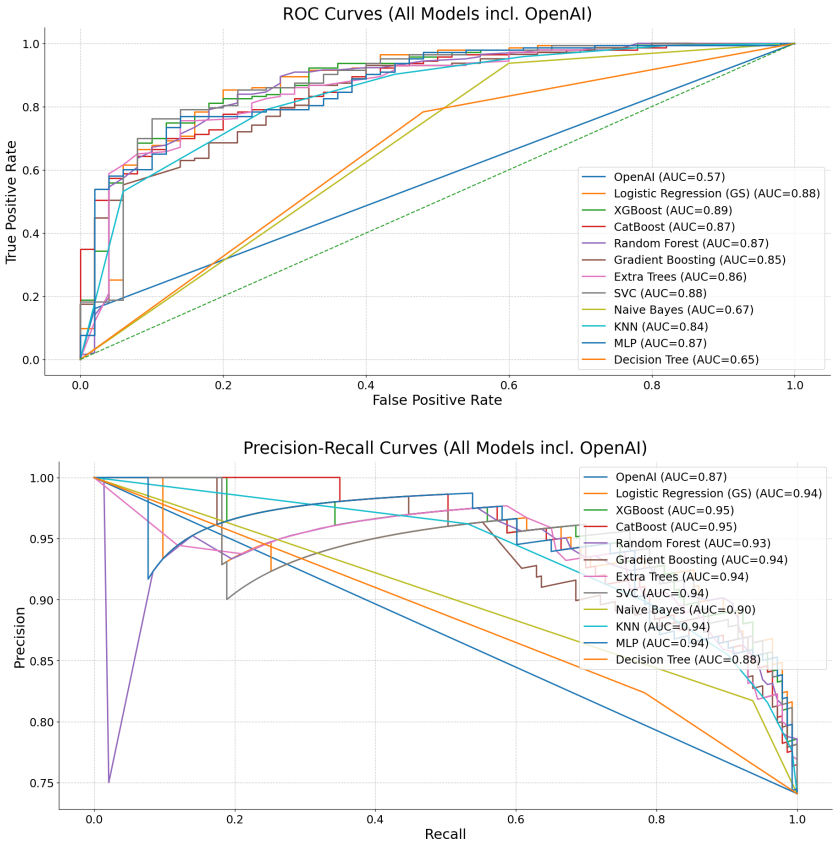

Рис. 6. PR- и ROC-кривые основных моделей

лежит выше диагонали на значительном промежутке FPR, а PR-кривая начинается от значения Precision, близкого к 1,0 в области малых Recall, и даже при Recall, равном 0,9, удерживается на уровне примерно 0,85.

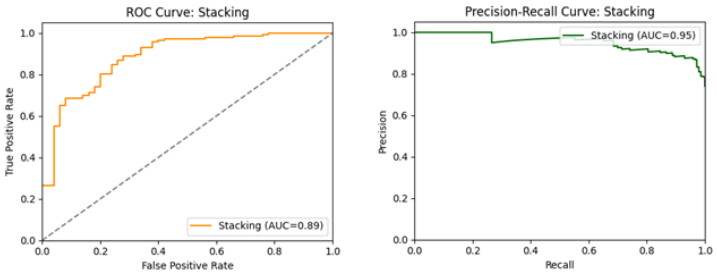

Рис. 7. PR- и ROC-кривые модели стекинга

Для полноты эксперимента на рисунке 8 приведены матрицы ошибок (Confusion Matrix) четырех характерных алгоритмов: Decision Tree, CatBoost, Logistic Regression (GS) и итогового Stacking. По сравнению с Decision Tree (высокий уровень ложноотрицательных срабатываний), Logistic Regression (GS) и CatBoost демонстрируют более сбалансированный расклад. Однако именно при переходе к стекингу заметно снижается число ошибок первого рода (текст с рассуждением классифицируется как «нет рассуждения»), а также сокращается доля неверных положительных ответов. Для обеспечения сбалансированной валидации использовалась стратифицированная кросс-валидация

(StratifiedKFold), которая сохраняет соотношение позитивных и негативных примеров в каждом фолде. Это позволяет избежать смещения при обучении и объективно оценивать качество классификаторов на различных подвыборках.

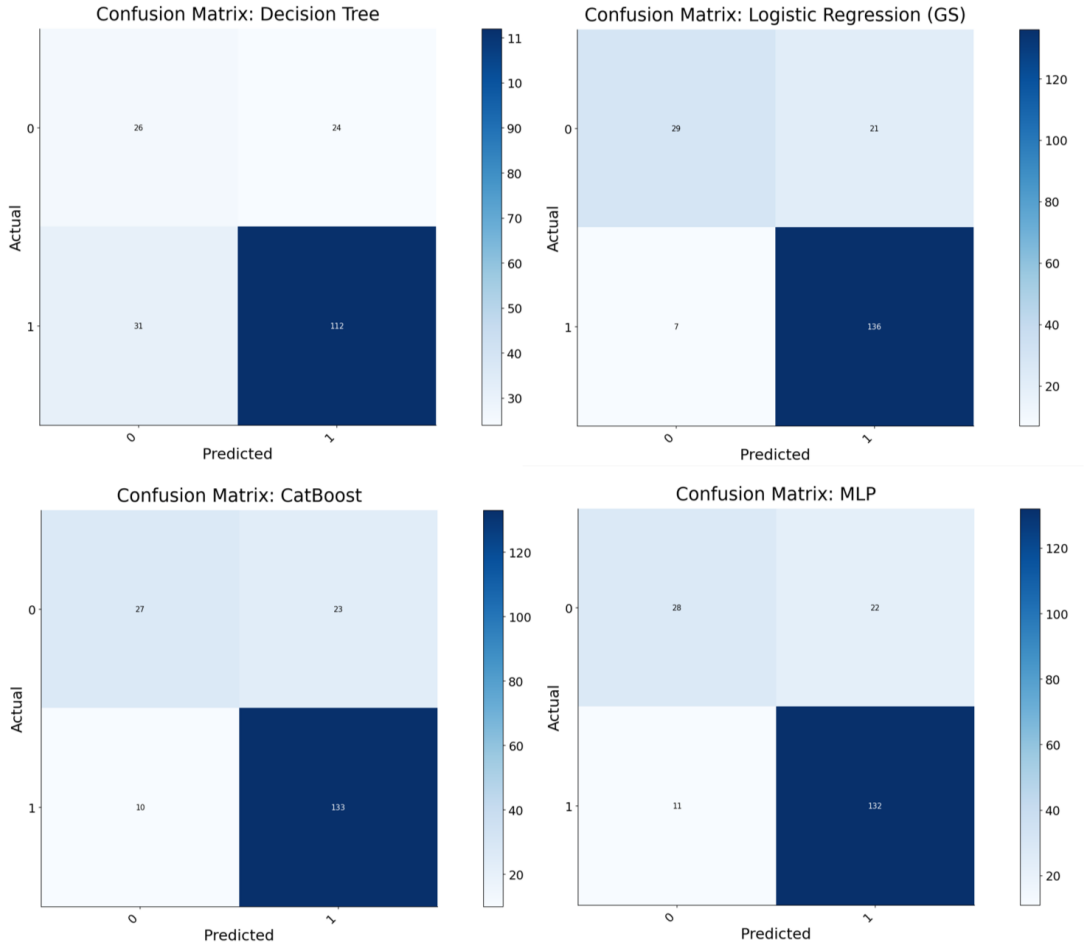

Рис. 8. Матрицы ошибок основных моделей и модели стекинга

В таблице 2 приведены итоговые интегральные метрики стекинга. Значение Recall, равное 0,93, практически компенсирует еще более высокий показатель Precision, равный 0,881, давая в итоге F1, равный 0,905. Площади под ROC и PR-кривыми составляют 0,887 и 0,952 соответственно, что подтверждает высокое качество вероятностной классификации.

Проведенные эксперименты показали, что комбинированный подход, сочетающий возможности деревоподобных бустингов (CatBoost, XGBoost, Random Forest) с линейной логистической регрессией (в качестве мета-модели), способен заметно повысить итоговое качество.

3. Анализ комбинации базовых моделей в стекинге

Рассмотренная в предыдущем разделе композиция (CatBoost, XGBoost, Random Forest и Logistic Regression) не является случайным набором алгоритмов. Каждая из

Таблица 2

Оценка качества выявления исследуемых информативных текстов на основе обученной модели языка

Метрика	Значение
Accuracy	0,855
Precision	0,881
Recall	0,930
F1-score	0,905
ROC AUC	0,887
PR AUC	0,952

указанных базовых моделей по-разному реагирует на особенности обучающей выборки, и итоговое объединение позволяет компенсировать их индивидуальные уязвимости. Случайный лес, используя ансамбль относительно неглубоких деревьев, хорошо обобщает общие закономерности и дает стабильные предсказания при нерезко выраженных признаках рассуждения. CatBoost и XGBoost, более агрессивные boosting-методы, глубже адаптируются к сложным аспектам логических связок; они способны захватывать комбинации признаков, связанные с причинно-следственными оборотами и специфическими лексическими единицами, но при этом иногда переоценивают неочевидные паттерны, что может приводить к ложным срабатываниям в менее формальных фрагментах текста. Линейная логистическая регрессия, хотя и часто уступает деревоподобным методам, дает ценный вклад за счет своей способности строго «наказывать» чрезмерно большие веса и удерживать правильный «баланс» между классами [1; 2].

Когда речь заходит о стекинге, формируется линейный классификатор, который «смешивает» вероятностные оценки базовых моделей. Такой подход оправдан, если систематические ошибки отдельных моделей различаются по знаку и проявляются в разных интервалах признаков. Например, CatBoost может ошибаться при доминировании повторяющихся лексем, тогда как Random Forest — при малом числе уникальных токенов, а Logistic Regression (GS) — при коллинеарности некоторых признаков. При объединении подобных сигналов мета-классификатор отбраковывает некорректные вбросы и усиливает верные предсказания. Важную роль в эффективности выбранной конфигурации играет также корреляция выходов базовых моделей. Если они дают однотипные вероятности, то стекинг не добавит принципиальных преимуществ. В рассматриваемом случае различия в реализации boosting-методов (CatBoost и XGBoost) и устойчивость Random Forest обеспечивают достаточную разнородность, а Logistic Regression, даже на базовом уровне, способна выделять характерные маркеры причинно-следственных конструкций. Таким образом, каждая из четырех составляющих стекинга формирует свой узнающий контур, и их линейная смесь позволяет устранить условно независимые ошибки. Выбор Logistic Regression в качестве мета-классификатора обусловлен тем, что при работе с небольшим числом входных признаков (вероятностных оценок) линейная модель с L_2-регуляризацией оптимально балансирует гибкость и защиту от переобучения, а ее коэффициенты легко интерпретируются. Таким образом, предлагаемая конфигурация (три деревоподобных алгоритма и одна линейная модель, объединенные логистической регрессией) обеспечивает выигрыш по основным метрикам — Recall достигает 0,93, а F1-score — 0,905. Анализ матриц ошибок подтверждает, что стекинг эффективно снижает случаи, когда текст с рассуждением ошибочно классифицируется как не содержащий рассуждений.

4. Результаты и обсуждение

Эксперименты с одиночными моделями, такими как Logistic Regression, CatBoost, SVC, XGBoost и т. д., показали, что на данных с достаточным объемом признаков (TF-IDF, 5 000 главных токенов) существенно выигрывают методы бустинга и оптимизированная логистическая регрессия. При этом CatBoost продемонстрировал высокое значение Recall, но не всегда удерживал Precision на одном уровне; XGBoost, напротив, имел более сбалансированные характеристики и достигал лучшего PR AUC (около 0,95). Нейросетевой подход (MLP) не превосходил бустинг, но в ряде случаев опережал классические деревья (Decision Tree) и KNN. Наиболее важный вывод заключается в том, что результирующий стекинг (Stacking) на основе CatBoost, XGBoost, Random Forest и Logistic Regression (в качестве мета-классификатора) дает прирост по основным метрикам. Значение F1-score в тестировании достигает 0,905, что выше, чем у любой одиночной модели, а Recall, равный 0,93, указывает на способность ансамбля «отлавливать» большинство текстов с рассуждениями. Precision (0,881) остается на высоком уровне, что подтверждает низкий уровень ложноположительных срабатываний. Анализ ROC- и PR-кривых, а также матриц ошибок демонстрирует, что стекинг эффективно сглаживает индивидуальные ошибки базовых моделей, обеспечивая оптимальный баланс между Recall и Precision.

Для оценки эффективности нашего метода было проведено сравнение с моделью Open AI GPT-4, используемой для автоматической разметки текстов. Как видно из таблицы 1, модель GPT-4 демонстрирует очень высокое значение Precision (около 0,9583), однако ее F1-score составляет всего 0,275, а Recall — лишь 0,3475. Такой результат может быть объяснен тем, что GPT-4 изначально нацелена на генерацию текста и обладает ограниченной адаптивностью к специфике классификации логических связей в заданном корпусе. В отличие от нее, предложенный нами метод стекинга, объединяющий алгоритмы бустинга и оптимизированную логистическую регрессию, обеспечивает гораздо более сбалансированное соотношение Recall и Precision, что подтверждается значениями F1-score (0,9067) и ROC AUC (0,8796). Таким образом, для задачи классификации рассудительных текстов наш подход оказывается значительно эффективнее, демонстрируя высокую точность и устойчивость к систематическим ошибкам.

Заключение

В данной работе рассмотрено применение современных алгоритмов машинного обучения для определения наличия рассуждений в русскоязычных текстах. Для эксперимента сформирован и размечен корпус из 963 фрагментов, которые после предобработки и векторизации (TF-IDF) использовались для сравнительного анализа различных моделей. Экспериментальные результаты показали, что оптимизированная логистическая регрессия (с подбором параметра C) и методы бустинга (XGBoost, CatBoost) демонстрируют наивысшие значения F1-score среди одиночных классификаторов. Интеграция этих моделей посредством стекинга (с использованием Random Forest и Logistic Regression в качестве мета-классификатора) позволила повысить итоговые показатели: F1-score составил 0,905, а Recall достиг 0,93. Анализ ROC-, PR-кривых и матриц ошибок подтверждает, что ансамблевый подход эффективно снижает число ложноотрицательных срабатываний, обеспечивая высокий уровень точности классификации.

В качестве дополнительных выводов можно отметить, что предложенный метод

успешно различает тексты, содержащие сложные логические связи и аргументацию, от текстов, в которых такие рассуждения отсутствуют. Например, модель корректно классифицировала следующие тестовые примеры:

Текст с рассуждениями: «Поскольку рынок нефти нестабилен, экономисты прогнозируют снижение цен. Следовательно, инвесторы должны быть осторожны при принятии решений.»

Текст без рассуждений: «Сегодня на улице солнечно. Температура воздуха составляет 25 градусов.»

В дальнейшем рекомендуется расширить подход за счет использования предобученных языковых моделей (таких как BERT, RuBERT), увеличения объема корпуса и применения специализированных методов детектирования сложных логических связей, что позволит еще более точно решать поставленную задачу.

СПИСОК ЛИТЕРАТУРЫ

1. Клячин, В. А. Атрибуция медийных текстов на основе обученной модели естественного языка и лингвистическая оценка качества идентификации / В. А. Клячин, Е. В. Хижнякова // Вестник Волгоградского государственного университета. Серия 2. Языкознание. — 2024. — Т. 23, № 5. — С. 31–46. — DOI: https://doi.org/10.15688/jvolsu2.2024.5.3

2. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations / Zh. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut // arXiv preprint arXiv:1909.11942. — 2020. — P. 1–17. — DOI: https://doi.org/10.48550/arXiv.1909.11942

3. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding / J. Devlin, M.-W. Chang, K. Lee, K. Toutanova // arXiv preprint arXiv:1810.04805. — 2019. — P. 1–16. — DOI: https://doi.org/10.48550/arXiv.1810.04805

4. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention / P. He, X. Liu, J. Gao, W. Chen // arXiv preprint arXiv:2006.03654. — 2021. — P. 1–23. — DOI: https://doi.org/10.48550/arXiv.2006.03654

5. RoBERTa: A Robustly Optimized BERT Pretraining Approach / Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov // arXiv preprint arXiv:1907.11692. — 2019. — P. 1–13. — DOI: https://doi.org/10.48550/arXiv.1907.11692

6. XLNet: Generalized Autoregressive Pretraining for Language Understanding / Zh. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le // arXiv preprint arXiv:1906.08237. — 2020. — P. 1–18. — DOI: https://doi.org/10.48550/arXiv.1906.08237

REFERENCES

1. Klyachin V.A., Khizhnyakova E.V. Atributsiya mediynykh tekstov na osnove obuchennoy modeli estestvennogo yazyka i lingvisticheskaya otsenka kachestva identifikatsii [Attribution of Media Texts Based on a Trained Natural Language Model and Linguistic Assessment of Identification Quality]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 2. Yazykoznanie* [Science Journal of Volgograd State University. Linguistics], 2024, vol. 23, no. 5, pp. 31-46. DOI: https://doi.org/10.15688/jvolsu2.2024.5.3

2. Lan Zh., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*, 2020, pp. 1-17. DOI: https://doi.org/10.48550/arXiv.1909.11942

3. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2019, pp. 1-16. DOI: https://doi.org/10.48550/arXiv.1810.04805

4. He P., Liu X., Gao J., Chen W. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. *arXiv preprint arXiv:2006.03654*, 2021, pp. 1-23. DOI: https://doi.org/10.48550/arXiv.2006.03654

5. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019, pp. 1-13. DOI: https://doi.org/10.48550/arXiv.1907.11692

6. Yang Zh., Dai Z., Yang Y., Carbonell J., Salakhutdinov R., Le Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*, 2020, pp. 1-18. DOI: https://doi.org/10.48550/arXiv.1906.08237

CONSTRUCTION OF A MODEL FOR THE TASK OF REASONING TEXT CLASSIFICATION

Alexander V. Kanygin

Postgraduate Student, Department of Computer Sciences
and Experimental Mathematics,
Volgograd State University
a.kanygin99@gmail.com
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract. The article addresses the task of classifying texts for the presence of reasoning (logical links, argumentation, cause-and-effect relationships). The aim of the study is to develop a method that allows for highly accurate determination of the "reasoning" nature of a text fragment using modern machine learning algorithms. Particular attention is paid to an ensemble approach based on stacking: strong models (XGBoost, CatBoost, Random Forest, etc.) are considered as base classifiers, while logistic regression serves as the meta-model. To justify the choice of stacking, we present the results of a comparative analysis of more than ten popular algorithms (Logistic Regression, SVC, Random Forest, CatBoost, XGBoost, etc.) by Accuracy, Precision, Recall, F1-score, ROC AUC, and PR AUC. The main stages of the study include the generation and annotation of the training dataset, preliminary text processing (tokenization, lemmatization, stop-word removal), feature vectorization (TF-IDF), and experimental comparison of the models on a control sample. The proposed stacking model showed the best overall performance across all metrics, enabling us to increase the accuracy of reasoning text classification to F1 equal to 0.905 at ROC AUC equal to 0.887.

Key words: machine learning, ensemble methods, stacking, TF-IDF, argumentation, text processing.