УДК 004.415

А.Б. Райимкулов, raiimkulov.bekzat@gmail.com
М.Ж. Ашералиева, asheralievazhazgul@gmail.com
Кыргызско-Германский институт прикладной информатики
Ю.С. Корякина , jk169@list.ru
Институт машиноведения автоматики и геомеханики НАН КР

МЕТОДЫ И ТЕХНОЛОГИИ ИЗМЕРЕНИЯ ТОНАЛЬНОСТИ ТЕКСТА

В представленной статье проводится комплексный сравнительный анализ современных методов машинного обучения, применяемых для автоматизированного определения тональной принадлежности текстовых данных. Исследование охватывает четыре алгоритмических подхода, для каждого из которых осуществляется оценка функциональных характеристик, вычислительной сложности, а также устойчивости к лексико-синтаксическим вариациям входных данных.

Результаты экспериментальных испытаний показывают, что использование логистической регрессии в сочетании с предварительной фильтрацией стоп-слов обеспечивает наивысшие показатели точности и стабильности при классификации текстов. Данный метод демонстрирует оптимальное соотношение вычислительной эффективности и качества прогнозирования, что особенно актуально для задач анализа пользовательских отзывов в интернет-среде, где преобладают короткие и структурно простые высказывания.

Ключевыми преимуществами выбранного подхода являются низкая ресурсоёмкость, высокая интерпретируемость результатов и технологическая простота внедрения в прикладные системы. На основании полученных данных формулируются методические рекомендации по выбору оптимальных алгоритмов машинного обучения для задач семантической интерпретации текстов и выявления эмоционально-смысловой направленности сообщений. Полученные результаты используются при проектировании интеллектуальных сервисов, ориентированных на обработку естественного языка и создание адаптивных систем анализа текстового контента.

Ключевые слова: машинное обучение, логистическая регрессия, наивный байесовский классификатор, метод опорных векторов, TF-IDF, векторизация текста, лемматизация, обработка естественного языка, SpaCy, анализ тональности текста, классификация отзывов, моделирование данных, метрики точности, матрица ошибок, сравнительный анализ моделей, оптимизация обучения, анализ пользовательских отзывов, Flask, веб-приложение для анализа тональности.

Введение

Одной из актуальных проблем современной аналитики данных является недостаточная точность существующих систем анализа тональности русскоязычных текстов. Большинство разработанных моделей ориентировано преимущественно на английский язык, вследствие чего они не способны адекватно интерпретировать лингвистические особенности русской речи. Такие модели испытывают трудности при обработке разговорных выражений, сарказма, аббревиатур, орфографических ошибок и эмоционально окрашенной лексики. Это приводит к снижению точности классификации отзывов и искажению объективного представления о восприятии продукта покупателями.

Проблема точного анализа тональности имеет особое значение в контексте онлайн-коммерции, где качество товаров и услуг напрямую связано с уровнем удовлетворенности клиентов. Применение анализа тональности в данной сфере позволяет более глубоко изучить мнение покупателей, выявить эмоциональные и семантические особенности отзывов, а также своевременно реагировать на выявленные проблемы. В отличие от традиционных систем оценивания, основанных на числовых рейтингах, такой подход обеспечивает качественную интерпретацию пользовательских комментариев и способствует повышению эффективности контроля качества.

Решение данной задачи требует разработки и внедрения методов, способных учитывать структурные и стилистические особенности русского языка, а также обеспечивать устойчивые результаты при обработке текстов различного характера. Эффективность подобных моделей во многом определяется качеством используемых обучающих данных.

Именно корректный выбор и подготовка обучающей выборки оказывают решающее влияние на точность и стабильность работы системы. В то же время процесс обучения моделей машинного обучения характеризуется высокой ресурсоемкостью и предъявляет повышенные требования к вычислительным характеристикам оборудования.

В современном машинном обучении существует широкий спектр моделей, применяемых для классификации текстов по тональности. Каждая из них имеет свои преимущества и ограничения, обусловленные объемом данных, типом анализируемого текста, степенью детализации и требованиями к вычислительным ресурсам. Выбор конкретной модели определяется спецификой исследовательской задачи и условиями её реализации.

Цель настоящего исследования заключается в определении наиболее эффективной модели анализа тональности русскоязычных текстов, обеспечивающей оптимальный баланс между точностью, скоростью обработки и вычислительной сложностью. В рамках эксперимента рассматриваются четыре метода, традиционно применяемые при решении задач классификации текстов: логистическая регрессия, метод обработки естественного языка с использованием библиотеки SpaCy, наивный байесов классификатор и линейная машина опорных векторов. Указанные методы отличаются интерпретируемостью, надежностью и высокой применимостью к анализу текстовых данных. Все используемые модели опираются на векторизацию TF-IDF, позволяющую преобразовать текстовую информацию в числовое представление, оптимальное для машинного анализа.

Для проведения замеров производительности моделей было использовано два компьютера со следующими характеристиками:

1. acer nitro 5 процессор – intel core i5-8400h, видеокарта – nvidia geforce gtx 1050-TI 4 GB GDDR5, 24 GB оперативной памяти.
2. Asus TUF gaming F17 процессор – intel core i7-12650H, видеокарта – nvidia geforce rtx 3060 8 GB GDDR6, 16 GB оперативной памяти.

Логистическая регрессия и TF-IDF векторизация

Логистическая регрессия зачастую используется для «бинарной классификации» [1], то есть если есть всего два класса, на которые мы разбиваем наши данные. В модели, которая была разработана, используется именно подобного рода классификация. Данные делятся на негативные и позитивные классы. Эта модель известна как более простая техника классификации. Она может использоваться для примерного понимания, как модель должна работать, а после уже двигаться в сторону более комплексных моделей. Эта модель была выбрана для использования в программном обеспечении потому, что она показала лучшие результаты в замерах точности при собранных данных для обучения. Результаты оказались точнее, чем в других моделях потому, что классов было всего два. Так же важно учесть, что метод использует «предикторные переменные» [1] для того, чтобы сделать вывод, принадлежат данные к одному или другому классу, что в обученной модели и делается. Логистическая регрессия – это линейная модель, которая предсказывает вероятность положительного или отрицательного отзыва на основе взвешенной суммы его характеристик [4].

Матрица ошибок для этой модели показала сбалансированное распределение правильных предсказаний по обоим классам. Как и SVM, логистическая регрессия продемонстрировала небольшое количество ошибок классификации, в основном в случаях, когда тон отзыва был неоднозначным. Визуализация наиболее влиятельных слов продемонстрировала четкое разделение полярности: слова с высоким положительным весом были напрямую связаны с положительным настроением, в то время как отрицательные коэффициенты соответствовали словам, выражающим недовольство. Гистограмма различий TF-IDF между положительными и отрицательными отзывами подтвердила это наблюдение, визуально показав, как лингвистические характеристики по-разному влияют на выражение настроения [1,5,6,8].

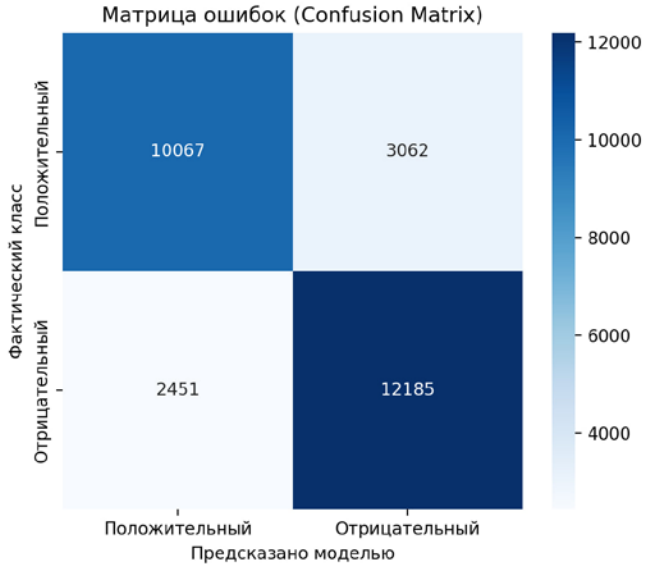

Рисунок 1 – Матрица ошибок логистической регрессии

При замерах производительности было выявлено, что на первом компьютере модель была обучена всего за 6 секунд, а на втором за 11.27 секунды. Точность оказалась 86 процентов, что может показаться не таким большим процентом, но на практике модель была точнее, чем все другие исследуемые модели.

```
C:\Users\osole\miniconda3\python.exe C:\Users\osole\PycharmProjects\JupyterProject\metod1.py
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\osole\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
Accuracy: 0.8615826078498128
Classification Report:
              precision    recall  f1-score   support

    negative       0.85      0.83      0.84      8938
    positive       0.87      0.89      0.88     11623

    accuracy                           0.86     20561
   macro avg       0.86      0.86      0.86     20561
weighted avg       0.86      0.86      0.86     20561

Прогноз: positive

Process finished with exit code 0
```

Рисунок 2 – Результаты логистической регрессии

Обработка натурального языка с использованием библиотеки SpaCy

Для обучения модели была использована логистическая регрессия и лемматизация с помощью библиотеки SpaCy. Лемматизация приводит все слова к лемме – ее нормальной(начальной) форме [2]. А также удаление стоп-слов и нормализации текста. Эти шаги помогают уменьшить размер словаря, повысить однородность данных и улучшить качество последующей классификации [4].

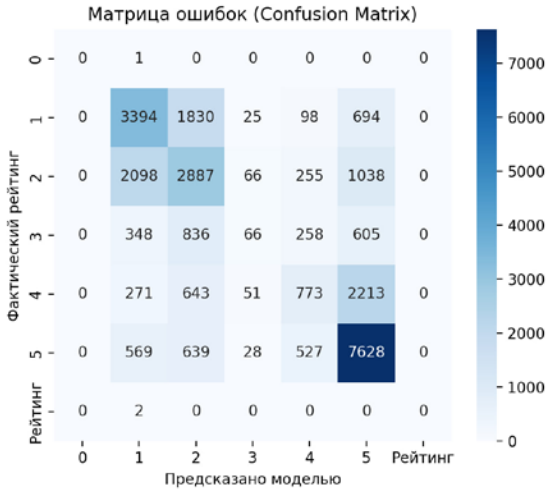

Рисунок 3 – Матрица ошибок метода с использование библиотеки SpaCy

Матрица ошибок для этого метода показала больше ошибочных классификаций, особенно среди пограничных случаев, когда язык был тонким или содержал сарказм. Также на рисунке 3 можно увидеть, что модель путала соседние категории оценок. Например, отзывы с фактическими оценками 1 и 2 часто неправильно классифицировали друг друга, и то же самое происходило между оценками 4 и 5. Это указывает на то, что модель испытывала трудности с обобщением и не могла четко разделить классы мнений [1].

При тестах на производительность и на точность было отмечено, что точность сильно упала и время завершения обучения увеличилось. На первом компьютере модель сработала за 16.82 секунды. На втором компьютере модель завершила работу за 9.2 секунды. Точность составила 52.98 процента.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(ASUS) PS C:\Users\ASUS\Desktop\3kurs\ML_kourswork> python metod2.py
Точность модели: 52.98%
Настроение для нового отзыва: Отрицательный
```

Рисунок 4 – Результаты с использованием SpaCy

Наивный Байес (MultinomialNB) и TF-IDF векторизация

Использование метода наивного Байеса может быть оправдано малым количеством данных для обучения модели. При условиях небольшого тренировочного набора данных этот метод имеет преимущество перед логистической регрессией. Наивный байесовский метод основан на предположении независимости признаков и вычислении апостериорной вероятности классов. При применении данной модели к корпусу русскоязычных отзывов наблюдалась высокая скорость обучения и предсказания, что делает её удобной для обработки больших потоков данных [5,9].

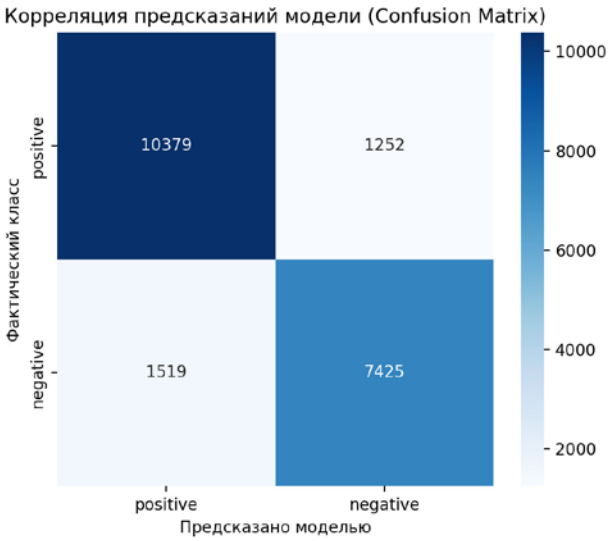

Рисунок 5 – Матрица ошибок наивного байесовского классификатора

На графике зависимости точности от размера словаря (max_features) отчётливо видно, что при увеличении числа признаков выше 2000 модель теряет устойчивость — точность колебалась в диапазоне от 0.79 до 0.83. Это связано с тем, что байесовский алгоритм плохо обрабатывает взаимосвязанные слова и многозначные выражения, часто встречающиеся в естественном языке.

Матрица ошибок показала смещение в сторону положительного класса: значительная часть нейтрально-негативных отзывов классифицировалась как положительные. На диаграмме распределения вероятностей предсказаний наблюдается выраженный пик в диапазоне вероятностей 0.6 – 0.8 для положительного класса, что указывает на чрезмерную уверенность модели [1]. Таким образом, несмотря на простоту реализации, байесовский классификатор показал ограниченную способность к захвату контекстных зависимостей в тексте.

Так же этот метод хорошо подходит для простых задач и, если характеристики компьютера невысоки, так как алгоритм, использованный в методе, очень простой. На тестовых компьютерах этот метод показал плохую производительность и не самую высокую точность.

На первом компьютере модель сработала за 64.0 секунды. На втором компьютере модель завершила работу за 35.59 секунды. Аккуратность метода составила около 80 процентов.

```
C:\Users\osole\miniconda3\python.exe C:\Users\osole\PycharmProjects\JupyterProject\metod3.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\osole\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\osole\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
Оценка модели:
              precision    recall  f1-score   support

           0       0.80      0.83      0.82     14529
           1       0.81      0.78      0.79     13221

    accuracy                           0.80     27750
   macro avg       0.80      0.80      0.80     27750
weighted avg       0.80      0.80      0.80     27750

Отрицательный отзыв

Process finished with exit code 0
```

Рисунок 6 – Результаты наивного Бейеса

Линейная поддержка векторных машин (LinearSVC)

Метод линейной поддержки векторных машин, как и логистическая регрессия, хорошо подходит для разбиения данных на два класса. Этот метод классифицирует данные, находя гиперплоскость, которая разбивает все данные на две части. Далее данные классифицируются исходя из того, в какой части гиперплоскости находятся данные. Этот метод известен как очень точный. Он не использует одни и те же данные для обучения. Эта модель очень быстрая и легко имплементируемая. Поэтому, если компьютер имеет мало памяти, то этот метод может быть подходящим. Данный метод так же хорошо работает со сложными данными. Но в то же линейная поддержка векторных машин должна быть хороша обучена, и это занимает много времени [4,8]. Линейная машина опорных векторов (LinearSVC) продемонстрировала наивысшую общую точность среди тестируемых моделей.

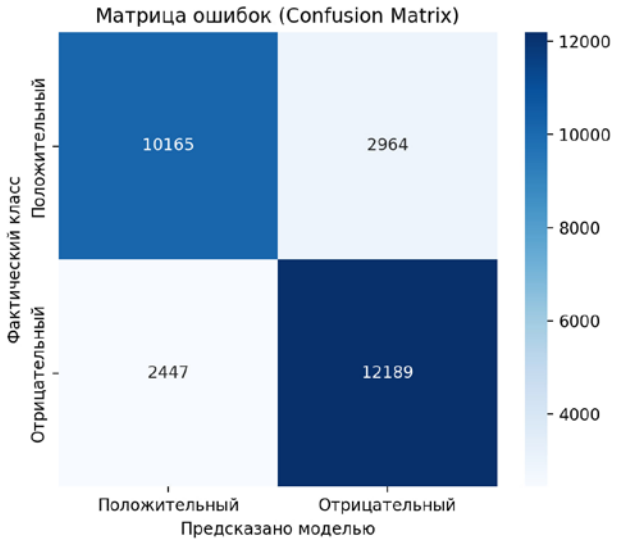

Рисунок 7 – Матрица ошибок метода опорных векторов

Матрица ошибок продемонстрировала четкое доминирование диагонали, что указывает на высокую способность различать два класса настроений. Было отмечено лишь несколько случаев неправильной классификации. Визуализация важности признаков показала, что эта модель эффективно идентифицировала дискриминационные слова, которые сильно влияли на решение о классификации. Положительные слова, такие как «качественный», «удобный» и «нравится», имели высокие положительные веса, в то время как отрицательные слова, такие как «некачественный», «плохой» и «ужасный», имели сильные отрицательные коэффициенты. Графики показали хорошо разделенные распределения столбцов, что свидетельствует о том, что SVM успешно уловила полярность признаков настроения с высокой точностью и стабильностью [1].

На первом компьютере модель сработала за 77.28 секунды. На втором компьютере модель завершила работу за 42.11 секунды. Точность составила 80 процентов.

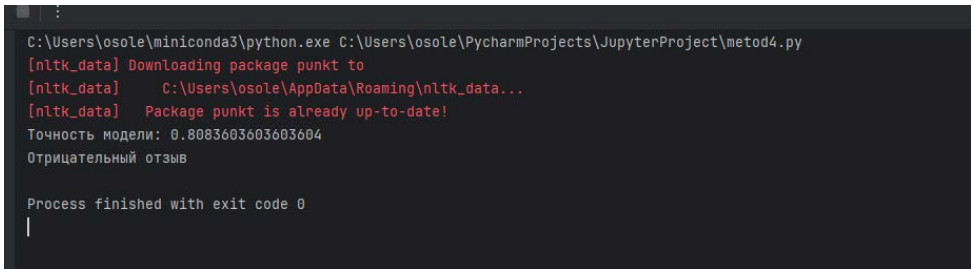

Рисунок 8 – Результаты линейной поддержки

Сравнение методов

Для выбора модели, подходящей для последующей интеграции в веб-приложение, были проведены сравнительные испытания всех четырёх методов. В таблице приведены результаты замеров производительности и точности каждой модели. В которой метод № 1 – логистическая регрессия, метод № 2 – SpaCy и логистическая регрессия, метод № 3 – Наивный Байес, метод № 4 – линейная поддержка векторных машин.

Таблица 1 – сравнение методов

Метод №:	Метод № 1	Метод № 2	Метод № 3	Метод № 4
Скорость На первом ПК	11.27 секунды	16.82 секунды	64 секунды	77.28 секунды
Скорость На втором Пк	6 секунд	9.2 секунды	35.59 секунды	42.11 секунды
Точность	86%	52.98%	80%	80%

Данные показывают, что лучшим выбором для классификации текстов по тональности является метод № 1 — логистическая регрессия. Она продемонстрировала наивысшую точность и лучшую производительность на обоих тестовых компьютерах. Именно этот метод был выбран для реализации в веб-приложении.

При сравнении всех четырех моделей линейная машина опорных векторов и логистическая регрессия показали лучшие результаты, продемонстрировав высокую точность и надежную генерализацию на невиданных данных. Модель многочленной наивной байесовской классификации, хотя и была немного менее точной, предоставила интерпретируемые вероятностные результаты и быстрое обучение, что делает ее практичным вариантом для крупномасштабных приложений. Модель с использованием библиотеки SpaCy показала самые слабые результаты из-за склонности к переобучению и ограниченной способности к генерализации за пределами обучающих примеров. Визуализации по всем моделям представили четкие доказательства этих различий в производительности: в то время как наивная байесовская и логистическая регрессионная модели отображали связные и интерпретируемые графики важности признаков, графики метода с библиотекой SpaCy показывали менее значимые паттерны, а графики SVM демонстрировали четкие, хорошо определенные различия в настроениях. В целом модель LinearSVC продемонстрировала наиболее стабильную и надежную производительность для анализа настроений в этом эксперименте [1,4].

Пользовательский интерфейс – это лучший способ дать пользователю, незнакомому с командной строкой, пользоваться программным продуктом. Для того, чтобы программа имела удобный интерфейс, для которого не придется скачивать и устанавливать дополнительные модули, было решено разработать веб-приложение. К такому приложению пользователь имеет доступ сразу в браузере. Такой интерфейс позволяет вводить свои данные для анализа без дополнительного обучения пользователя.

Разработка веб-приложения

Созданное веб-приложение представляет собой браузерный интерфейс с серверной частью, написанной на языке программирования Python с использованием фреймворка Flask. Интерфейс имеет современный и интуитивно понятный дизайн. Для удобства отдельные комментарии вводятся на новой строке. При создании дизайна были учтены лучшие решения аналогичных приложений с акцентом на удобство и читаемость [10].

Все модели обучались на одинаковом подмножестве данных, разделённом в пропорции 80/20 для обучающей и тестовой выборок. Для обеспечения воспроизводимости эксперимента использовался фиксированный параметр random_state = 42.

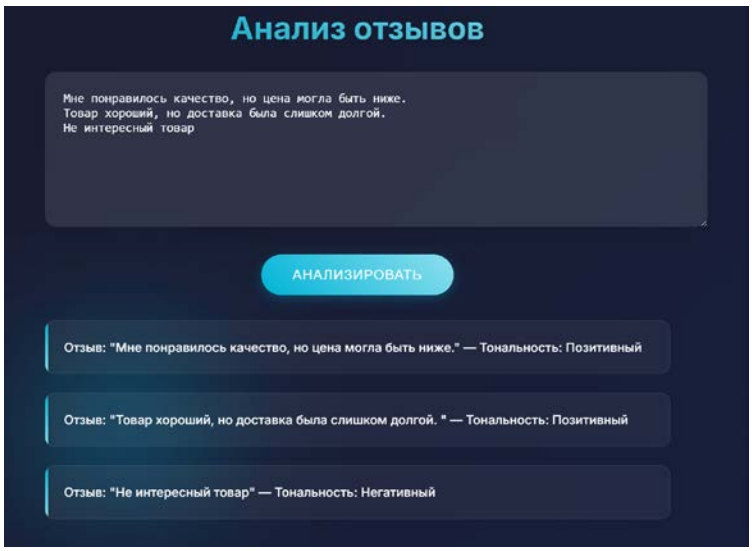

Рисунок 9 – Пример интерфейса

После проведения предварительных экспериментов и выбора наиболее подходящих алгоритмов для анализа тональности текста была выполнена углублённая оценка качества работы каждой из моделей. Для этого были построены матрицы ошибок и графики зависимости точности от параметров обучения. Такой подход позволил выявить сильные и слабые стороны каждой модели, а также определить оптимальные условия их применения на практике.

Заключение

В данном исследовании был проведен всесторонний обзор и сравнительный анализ современных методов машинного обучения для измерения тональности текста. Основной целью было определить наиболее эффективный и точный подход к классификации эмоциональной окраски в коротких текстах, созданных пользователями, таких как онлайн-отзывы о продуктах. В контролируемых экспериментальных условиях были изучены четыре различных метода: логистическая регрессия, наивный байесовский алгоритм, линейные машины поддержки векторов и обработка естественного языка с помощью SpaCy [6].

Научная новизна данного исследования заключается в систематической оценке производительности моделей в различных аппаратных средах и конфигурациях предварительной обработки, а также в определении оптимального баланса между вычислительной эффективностью и точностью классификации [4]. Исследование показало, что логистическая регрессия с векторизацией TF-IDF и удалением стоп-слов дает наиболее эффективные результаты, достигая 86% точности при минимальных вычислительных затратах. Было установлено, что эта комбинация превосходит более сложные модели при эквивалентных ограничениях данных и ресурсов [5].

С методологической точки зрения эта работа вносит вклад в область анализа тональности, показывая, что более простые линейные модели при правильной оптимизации могут достигать высокой надежности и интерпретируемости без необходимости использования обширных вычислительных ресурсов [6,7]. Это открытие бросает вызов преобладающей тенденции применять неоправданно сложные нейронные или гибридные модели к относительно простым задачам классификации.

С практической точки зрения разработанное веб-приложение, основанное на предложенной модели, представляет собой доступный и эффективный инструмент для автоматического анализа тональности. Оно позволяет пользователям и организациям отслеживать качество обратной связи, извлекать полезную информацию из текстовых отзывов и улучшать процессы принятия решений [11].

В заключение, исследование не только подтверждает жизнеспособность логистической регрессии как надежной базы для анализа тональности, но и представляет

усовершенствованный рабочий процесс, который оптимизирует предварительную обработку и выбор модели для легких, реальных реализаций [8]. Будущая работа может быть сосредоточена на расширении этой структуры на многоязычные наборы данных, изучении гибридных подходов, сочетающих лексические и контекстные вложения, и интеграции механизмов адаптивного обучения в реальном времени.

Литература

1. Choosing the Best Machine Learning Classification Model and Avoiding Overfitting
2. [Электронный ресурс]. – MathWorks, 2025. – URL:
3. https://www.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html (дата обращения: 27.09.2025 22:04).
4. Лемматизация [Электронный ресурс] // Wikipedia. – URL:
5. https://ru.wikipedia.org/wiki/Лемматизация (дата обращения: 28.09.2025 11:14).
6. Jurafsky D. Speech and Language Processing [Электронный ресурс]: 3rd ed., draft / D. Jurafsky, J. H. Martin. – Stanford University, 2023. – URL: https://web.stanford.edu/~jurafsky/slp3/ (дата обращения: 14.10.2025 16:20).
7. Pedregosa F. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel [и др.] // Journal of Machine Learning Research. – 2011. – Т. 12. – С. 2825–2830. – URL: https://jmlr.org/papers/v12/pedregosa11a.html (дата обращения: 14.10.2025 16:20).
8. Raschka S. Python Machine Learning: 4th ed. / S. Raschka, V. Mirjalili. – Packt Publishing, 2022. – URL: https://sebastianraschka.com/books.html (дата обращения: 14.10.2025 16:20).
9. Manning C. D. Introduction to Information Retrieval / C. D. Manning, P. Raghavan, H. Schütze. – Cambridge: Cambridge University Press, 2008. – DOI: 10.1017/CBO9780511809071. – URL: https://nlp.stanford.edu/IR-book/ (дата обращения: 14.10.2025 16:20).
10. Bird S. Natural Language Processing with Python / S. Bird, E. Klein, E. Loper. – O'Reilly Media, 2009. – URL: https://www.nltk.org/book/ (дата обращения: 14.10.2025 16:20).
11. Zhang L. Deep Learning for Sentiment Analysis: A Survey / L. Zhang, S. Wang, B. Liu // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. – 2018. – Vol. 8, No. 4. – e1253. – DOI: 10.1002/widm.1253 (дата обращения: 14.10.2025 16:20).
12. Kowsari K. Text Classification Algorithms: A Survey / K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. E. Brown // Information. – 2019. – Vol. 10, No. 4. – 150. – DOI: 10.3390/info10040150 (дата обращения: 14.10.2025 16:20).
13. Корякин, С. В. Разработка универсальной среды проектирования автоматизированных систем защищенного исполнения / С. В. Корякин // Проблемы автоматики и управления. – 2021. – № 2(41). – С. 40 – 55. – EDN EFUUDC.
14. Корякин, С. В. Разработка концепции построения программно-аппаратного ядра универсальной среды проектирования автоматизированных систем защищенного исполнения / С. В. Корякин // Проблемы автоматики и управления. – 2020. – № 1(38). – С. 60–69. – DOI 10.5281/zenodo.3904117. – EDN IWZQBP.