ՀՏԴ 004.8

Классификация литературных произведений по жанрам с использованием методов машинного обучения

Варданян Айарпи,
Оганян Гегине

DOI: https://doi.org/10.58726/27382923-2025.2-84

Ключевые слова: *метод векторизации TF-IDF, классификатор, алгоритм SGD Classifier, метрика, матрица несоответствий, цифровые библиотеки*

Введение

Современное развитие искусственного интеллекта и машинного обучения существенно расширяет спектр их применения, позволяя автоматизировать задачи, ранее требовавшие исключительно человеческого участия, и эффективно решать проблемы, ранее считавшиеся нерешаемыми. Одной из таких задач является жанровая классификация литературных произведений.

Стремительное развитие информационных технологий с конца XX века трансформировало формы восприятия текста: наряду с печатными изданиями широкое распространение получили электронные книги и аудиокниги, что в свою очередь способствовало увеличению числа пользователей, читающих книги в электронном формате, и сформировало новые требования к представлению и обработке литературного контента [14]. Рост объёмов цифрового контента, а также изменение потребительских привычек сформировали потребность в автоматизированной классификации литературных произведений по жанрам. Это особенно актуально для цифровых библиотек, онлайн-магазинов и образовательных платформ, где ручная обработка больших массивов данных затруднительна.

Большинство существующих исследований ориентированы на англоязычные тексты и часто используют ограниченные источники признаков – обложки, краткие описания или метаданные [8; 13; 16]. Подобные подходы игнорируют семантические и морфологические особенности текстов, что снижает точность классификации, особенно в языках с богатой морфологией, таких как русский и армянский. Это приводит к снижению точности при классификации жанров с близким содержательным контекстом (например, «ужасы/мистика» и «фэнтези»).

Настоящее исследование направлено на устранение этих ограничений путём:

1. применения TF-IDF-векторизации с адаптированными стоп-словами для многоязычного корпуса;
2. сравнительного анализа эффективности семи алгоритмов машинного обучения и выбора оптимального классификатора по совокупности метрик;
3. формирования репрезентативного корпуса объёмом свыше 8,8 тыс. произведений.

Реализация выполнена на Python с использованием библиотек pndas, NLTK, scikit-learn и др., что обеспечило полную цепочку обработки данных – от сбора и предобработки до построения и тестирования моделей. Эксперименты показали, что при корректной подготовке данных классические методы машинного обучения способны

эффективно решать задачу жанровой классификации многоязычных текстов. Перспективы дальнейшей работы включают внедрение нейросетевых архитектур (BERT, Transformers), ансамблевых методов и расширение корпуса за счёт дополнительных языковых подсистем.

1.Предлагаемая методология

Предлагаемый подход можно представить следующей схемой (рисунок 1) [5]:

Рисунок 1. Предлагаемая модель системы

1.1. Определение жанров (меток)

Эффективность классификации литературных произведений во многом зависит от чёткости и полноты заранее определённых жанровых меток.

Для обучения модели был сформирован корпус литературных произведений, включающий 8842 книг на русском языке. Источниками данных послужили электронные библиотеки и онлайн-платформы, содержащие книги разных жанров. Выранная классификация основывается на предложенных наиболее известными онлайн-платформами [1;4;6;10;17] классификациях, которые были оптимизированы на

основе нашего корпуса данных. В общей сложности было определено 16 жанров: бизнес, боевики, детектив, дом/дача, знания/навыки, история, классика, поэзия, психология/мотивация, религия, роман, спорт/здоровье/красота, ужас/мистика, фантастика, фэнтези, хобби/досуг. При сборе данных пришлось отказаться от некоторых жанров, так как количество книг этих жанров было ограничено (менее 100 наименований), что привело бы к неточностям при классификации. Например, были исключены комедии, компьютерная, культура/искусство, приключения и т.д.

Сбор и обработка данных

Прежде всего была создана собственная база данных: с помощью вспомогательной функции была автоматически собрана коллекция, состоящая из 8842 загруженных книг.

Затем была проведена предварительная обработка текстов собранных книг – удаление лишних символов, знаков и пробелов. Для упрощения работы набор данных был переведен из списка в DataFrame из библиотеки pandas.

Далее для работы с алгоритмами машинного обучения был использован пакет sklearn [21]. Данный пакет предоставляет широкий функционал для работы с наборами данных, в том числе разные алгоритмы машинного обучения.

Для работы с текстовыми данными из пакета sklearn использовались класс TfidfVectorizer и функция train_test_split(). В качестве вспомогательного аспекта для работы с текстом была так же использована библиотека NLTK [18].

Для преобразования исходных текстовых данных в формат, пригодный для машинного обучения, то есть, для векторизации текстовых данных, применялся метод TF-IDF (Term Frequency – Inverse Document Frequency), реализованный в Python посредством класса TfidfVectorizer библиотеки scikit-learn. Метод *TF-IDF (Term Frequency–Inverse Document Frequency)*, зарекомендовал себя как эффективный инструмент для извлечения признаков в задачах классификации текстов [14; 20].

Аббревиатуры в названии «TF» (Term Frequency) и «IDF» (Inverse Document Frequency) имеют следующие значения.

- TF (частота термина) показывает, как часто определённое слово встречается в данном документе. Другими словами, оно измеряет важность слова в рамках конкретного документа.
- IDF (обратная документная частота) измеряет насколько уникально слово в целом по всему набору документов. Слова, которые встречаются во многих документах, имеют низкое значение IDF, так как они не представляют особой информационной ценности.

В отличие от простых моделей "мешка слов", TF-IDF зарекомендовал себя как эффективный инструмент извлечения признаков, позволяющий учитывать как локальную частоту появления термина в документе (TF), так и его глобальную уникальность в корпусе (IDF), то есть, TF-IDF учитывает не только частоту термина в документе, но и его дискриминативную способность в масштабе корпуса, что позволяет уменьшить вес общеупотребительных слов и повысить значимость информативных признаков [12]. Данное свойство особенно важно для жанровой классификации многоязычных текстов, где необходимо корректно обрабатывать как частотные маркеры языка, так и специфические жанровые лексемы.

Базовая формула расчёта TF-IDF объединяет понятия TF и IDF для оценки важности каждого слова в каждом документе:

$$TF\text{-}IDF(t, d) = TF(t, d) * IDF(t)$$ ([14; 20])

Эта *концептуальная* формула, по которой значение TF-IDF получается как произведение:

- TF(t, d) – локальной частоты термина t в документе d;
- IDF(t) – глобальной меры уникальности термина в корпусе, то есть обратной документной частоты для термина t [3].

Развёрнутая (математическая) форма, то есть математическая модель метода описывается выражением:

$$TF\text{-}IDF(t, d) = TF(t, d) * log\frac{N}{1+DF(t)}$$ ([3; 15]),

где:

- N – общее число документов в корпусе;
- DF(t)) – количество документов, в которых встречается термин t;
- $log\frac{N}{1+DF(t)}$ – конкретная формула для вычисления *IDF(t)*, при этом добавление 1 в знаменателе используется для сглаживания (чтобы избежать деления на ноль, если термин встречается во всех документах).

В рамках проведённого исследования векторизация сопровождалась удалением языковых stop_words («стоп-слова»), адаптированных под каждый язык корпуса, а также применением параметров min_df (задаёт порог для редких слов) и max_df (задаёт порог для очень распространённых слов) для фильтрации редких и чрезмерно частотных токенов. stop_words – это слова, которые часто используются и не несут смысловой нагрузки для классификации машинного обучения, например:

- стоп-слова для русского языка – и, в, во, не, что, он, на, я, с, со, как, а, то, все, она, так, его, но, да, ты, к, у, же, вы, за, бы, по, только, ее, мне, было, вот, от, меня, еще, нет, о, из, ему, …
- стоп-слова для английского языка – I, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, him, his, himself, she, she's, her, hers, herself, it, it's, its, itself, …

Такой подход позволил минимизировать влияние высокочастотных функциональных слов, не несущих значимой жанровой нагрузки, и повысить дискриминативную способность признакового пространства.

Выбор TfidfVectorizer был обусловлен его устойчивостью к масштабированию, поддержкой работы с разреженными матрицами и встроенной интеграцией с алгоритмами scikit-learn, что обеспечило единый процесс обработки: от предобработки текста до построения классификаторов. В частности, использование разреженного формата хранения признаков позволило работать с корпусом объёмом более 8,8 тыс. произведений без значительных потерь по времени выполнения и объёму потребляемой памяти.

Функция train_test_split() используется для разделения набора данных на обучающую и тестовую выборки. В частности, в данной работе для обучения использовалось 80 % всего набора данных, а оставшиеся 20 % – для тестирования.

2. Алгоритмы классификации

Для решения задачи многоклассовой классификации текстов были протестированы семь широко используемых алгоритмов машинного обучения: *Complement Naïve Bayes (ComplementNB), Multinomial Naïve Bayes (MultinomialNB), Support Vector Classifier (SVC), Logistic Regression, Stochastic Gradient Descent Classifier (SGDClassifier),*

Ridge Classifier и *Random Forest Classifier*. Выбор данных методов обусловлен их доказанной эффективностью в задачах классификации текстов и умеренной вычислительной сложностью [9; 11; 15].

На основе предварительных экспериментов и оценки метрик *accuracy* и *macro-averaged precision/recall* было установлено, что наилучшие результаты демонстрируют линейные методы, в частности *SGDClassifier*, основанный на стохастическом градиентном спуске (SGD) [7]. Такой подход особенно эффективен при работе с разреженными векторными представлениями (например, TF-IDF), характерными для текстовых данных большого размера [5].

Сравнение результатов показало, что *SGDClassifier* достигает точности 0,67, превосходя методы Logistic Regression и Ridge Classifier (0,66), а также байесовские классификаторы, чувствительные к дисбалансу классов. Данный вывод согласуется с результатами Joachims [15], где линейные классификаторы на больших текстовых корпусах показали сопоставимую или более высокую производительность по сравнению с нелинейными методами (таблица 1).

Таблица 1

Результаты метрик выбранных методов

Методы	*Accuracy*	*Macro avg (precision)*	*Micro avg (recall)*
ComplementNB	0,63	0,65	0,61
MultinomialNB	0,53	0,62	0,50
SVC	0,65	0,67	0,64
LogisticRegression	0,66	0,67	0,66
SGDClassifier	**0,67**	**0,67**	**0,67**
RidgeClassifier	0,66	0,67	0,67
RandomForestClassifier	0,56	0,54	0,54

Кроме того, результаты экспериментов подтверждают, что ансамблевые модели, такие как *Random Forest*, уступают по качеству линейным методам при работе с высокоразмерными текстовыми признаками, что согласуется с ранее опубликованными исследованиями [19].

Таким образом, *SGDClassifier* был выбран как основной классификатор для предложенной системы ввиду его относительно высокой точности, масштабируемости и способности эффективно работать с большими корпусами текстов.

Несмотря на то, что классификатор SGDClassifier показал наилучшие результаты среди протестированных алгоритмов, данный показатель нельзя считать высоким в абсолютном смысле. Значение accuracy на уровне 67 % отражает лишь умеренную эффективность модели. Это объясняется рядом объективных факторов:

1. корпус данных содержит широкий спектр жанровых стилей и тематик, что приводит к значительной вариативности лексики и усложняет обучение модели;

2. жанровая разметка имеет семантические пересечения – многие тексты могут

одновременно относиться к нескольким жанрам (например, «ужасы/мистика» и «фэнтези», «роман» и «психология/мотивация»).

Представим код обучения и тестирования модели с применением SGDClassifier (рисунок 2).

```
sgd1 = SGDClassifier()
sgd1.fit(X_train_vec, y_train)
print(sgd1.score(X_test_vec, y_test))
y_pred_sgd1=sgd1.predict(X_test_vec)
accuracy = accuracy_score(y_test, y_pred_sgd1)
print("Accuracy: ", accuracy)
draw_Conf(y_pred_sgd1, sgd1)
print(classification_report(y_test, y_pred_sgd1))
```

Рисунок. 2. Обучение и тестирование модели

Сначала создаётся объект SGDClassifier. Затем проводится обучение на данных X_train_vec и y_train. Далее, с помощью метода predict() получаем прогнозы модели для тестовых данных, то есть предсказанные метки, после чего вычисляется *accuracy,* то есть сравниваются реальные и предсказанные метки, результаты выводятся на экран.

Также составляется матрица несоответствий (confusion matrix), которая показывает, насколько правильно и ошибочно модель классифицировала объекты по разным классам.

В конце с помощью функции classification_report() формируется подробный отчёт, включающий accuracy, precision, recall и другие метрики.

Полученные модели для последующего использования в приложении необходимо сначала сохранить в виде файлов, а затем интегрировать в приложение.

3. Анализ полученных результатов

Построенная матрица несоответствий (рисунок 2) демонстрирует ошибки модели, то есть сколько объектов она отнесла к неправильному классу – жанру.

Матрица несоответствий на рис. 3 показывает соответствие предсказанных значений классификатором SGDClassifier с реальными значениями. В целом, модель демонстрирует относительно небольшое количество ошибок. Как и ожидалось, значения по диагонали матрицы соответствуют тем жанрам, которые классификатор определил правильно. Однако имеются и некоторые ошибки.

Представим жанры, для которых были зафиксированы сравнительно большие ошибки классификации:

- из 100 тестовых образцов жанра ужасы/мистика 25 % (25 образцов) были классифицированы как фэнтези, а 10 % (10 образцов) – как фантастика;
- из 102 тестовых образцов жанра фэнтези около 23 % (23 образца) были классифицированы как ужасы/мистика;
- из 135 тестовых образцов жанра хобби/досуг около 15 % (20 образцов) были классифицированы как спорт/здоровье/красота, а из 132 образцов жанра спорт/здоровье/красота около 16 % (21 образец) были классифицированы как хобби/досуг;
- из 133 образцов жанра фантастика около 13 % (17 образцов) были классифициро-

ваны как боевики, а из 127 тестовых образцов жанра боевики около 15 % (19 образцов) были классифицированы как фантастика.

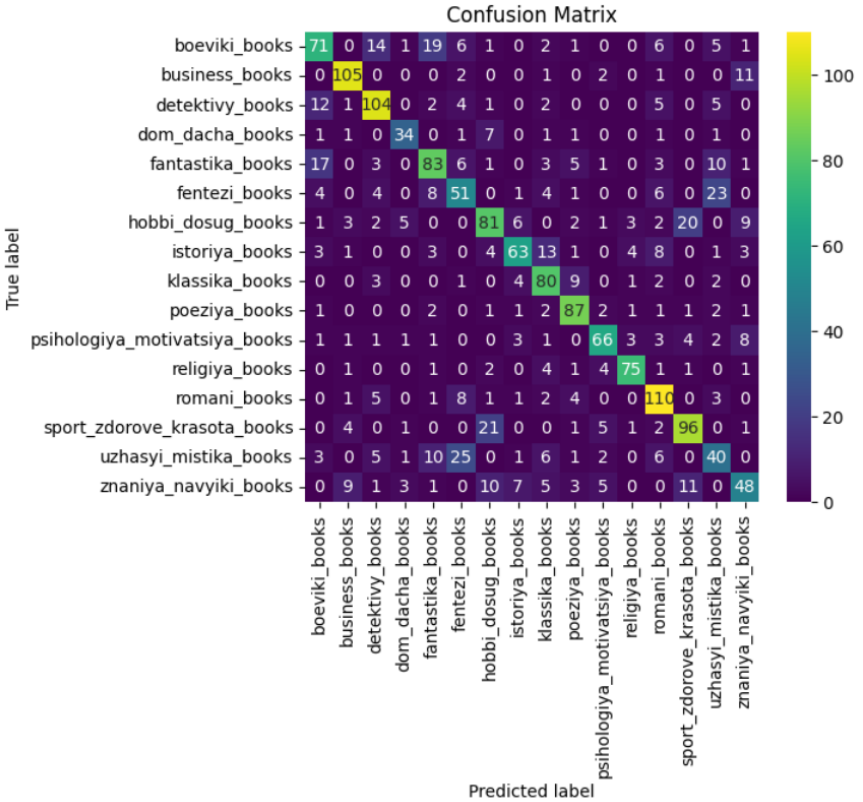

Рисунок 3. Матрица несоответствий

Представим жанры с умеренным количеством ошибок классификации:

- из 104 тестовых образцов жанра история 13 % (13 образцов) были классифицированы как классика;
- из 136 тестовых образцов жанра детектив 9 % (12 образцов) были классифицированы как боевики, а из 127 образцов жанра боевики 11 % (14 образцов) – как детектив;
- из 122 образцов жанра бизнес 9 % (11 образцов) были классифицированы как знания/навыки, а из 103 образцов жанра знания/навыки около 9 % (9 образцов) – как бизнес.

Наилучшие результаты были зафиксированы по следующим жанрам:

- из 122 образцов жанра бизнес 86 % (105 образцов) были правильно классифицированы как бизнес;

▪ из 136 образцов жанра детектив около 77 % (104 образца) – как детектив;

▪ из 136 тестовых образцов жанра роман 81 % (110 образцов) были классифицированы как роман.

Интересно, что некоторые жанры, несмотря на сравнительно небольшое количество образцов, также показали высокую аккуратность. Например:

▪ тестовый набор жанра «дом/дача» состоит из 48 образцов, из которых 71 % (34 образца) были классифицированы как «дом/дача»;

▪ из 91 тестового образца жанра религия 82 % (75 образцов) были классифицированы как религия;

▪ тестовый набор жанра «психология/мотивация» состоит из 95 образцов, из которых около 70 % (66 образцов) были классифицированы как «психология/мотивация».

Это предполагает, что некоторые классы, обладающие специализированной терминологией, легче различимы даже при ограниченном объёме данных.

Исследование демонстрирует эффективность классических алгоритмов при корректной подготовке корпуса и обозначает перспективы дальнейших исследований: внедрение нейросетевых архитектур (BERT, Transformers), применение ансамблевых методов и локализация моделей для различных языков. Представленный подход имеет практическую ценность для цифровых библиотек, онлайн-магазинов и образовательных ресурсов.

DOI: https://doi.org/10.58726/27382923-2025.2-84

Литература

1. Бесплатная библиотека электронных книг. https://fictionbook.ru
2. Доля рынка электронных книг, размер, тенденции и отраслевой анализ. https://www.mordorintelligence.com/ru/industry-reports/e-book-market
3. Извлечение признаков из текстовых данных с использованием TF-IDF. https://habr.com/ru/companies/otus/articles/755772/
4. Литрес https://www.litres.ru/
5. Al-Absi A. A., Sain M., Pattnaik P.K. Proceedings of the 3rd International Conference on Smart Computing and Cyber Security: Strategic Foresight, Security Challenges and Innovation. https://www.google.am/books/edition/Proceedings_of_3rd_International_Confere/p4wWEQAAQBAJ?hl=ru&gbpv=1
6. AvidReaders. https://avidreaders.ru/genre/
7. Bottou L. Stochastic gradient descent tricks in Neural Networks: Tricks of the Trade, Springer, 2012, pp. 421–436. [DOI: 10.1007/978-3-642-35289-8_25]
8. Chiang H., Ge Y., Wu C. Classification of Book Genres by Cover and Title. http://cs229.stanford.edu/proj2015/127_report.pdf
9. Cortes C. and Vapnik V. Support-vector networks, Machine Learning, vol. 20, pp. 273-297, 1995. [DOI: 10.1007/BF00994018]
10. Flibusta: https://flibusta.one/books-genres/
11. Hastie T., Tibshirani R., and Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009.

12. Joachims T. Text categorization with support vector machines: Learning with many relevant features. Springer, 1998, pp. 137–142. [DOI: 10.1007/BFb0026683]
13. Kundu C., Zheng L. Deep multi-modal networks for book genre classification based on its cover https://arxiv.org/abs/2011.07658v1
14. Manning C. D., Raghavan P., and Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008. Available: https://doi.org/10.1017/CBO9780511809071
15. McCallum A. and Nigam K. A comparison of event models for Naive Bayes text classification, in Proc. AAAI Workshop on Learning for Text Categorization, 1998.
16. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space https://arxiv.org/abs/1301.3781 (дата обращения 28.09.2025)
17. MyBook https://mybook.ru/
18. NLTK https://www.nltk.org/
19. Rokach L. and Maimon O. Data Mining with Decision Trees: Theory and Applications, 2nd ed. World Scientific, 2015.
20. Salton G. A. and Buckley C. Term-weighting approaches in automatic text retrieval. Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988. [DOI: 10.1016/0306-4573(88)90021-0]
21. Sklearn https://scikit-learn.ru/

Մեքենայական ուսուցման մեթոդների կիրառմամբ գրական ստեղծագործությունների ժանրային դասակարգում

Վարդանյան Հայարփի,
Օհանյան Հեղինե

Ամփոփում

Հանգուցային բառեր. *վեկտորիզացիայի TF-IDF մեթոդ, կլասսիֆիկատոր, SGDClassifier ալգորիթմ, մետրիկա, անհամապատասխանության մատրիցա, թվային գրադարաններ*

Հոդվածում դիտարկվում է գրական ստեղծագործությունների ավտոմատացված ժանրային դասակարգման խնդիրը՝ մեքենայական ուսուցման մեթոդների կիրառմամբ: Ուսումնասիրության արդիականությունը պայմանավորված է թվային գրադարանների և կրթական հարթակների աճով, որտեղ մեծ տեքստային զանգվածների ձեռքով մշակումը արդյունավետ չէ: Առաջարկվում է մոտեցում՝ հիմնված բազմալեզու կորպուսի համար հարմարեցված stop-բառերով տեքստերի TF-IDF վեկտորացման և յոթ դասակարգման ալգորիթմների համեմատական վերլուծության վրա: Լավագույն արդյունքներն ապահովել է SGDClassifier ալգորիթմը (ճշգրտություն՝ 0.67)՝ գերազանցելով թե՛ բայեսյան մեթոդները և թե՛ անսամբլային մոդելները:

Փորձարկված ալգորիթմների մեջ SGDClassifier-ը ցուցադրեց ամենաբարձր արդյունավետությունը, ինչը վկայում է արդյունավետության միջին մակարդակի մասին: Այս արդյունքը պայմանավորված է ժանրերի սեմանտիկ նմանությամբ և դասական մոդելների սահմանափակումներով:

Սխալների վերլուծությունը բացահայտել է դժվարություններ՝ կապված բովանդակորեն մոտ ժանրերի («սարսափ/միստիկա» և «ֆենթեզի») տարբերակման հետ, մինչդեռ մասնագիտացված ժանրերը («կրոն», «բիզնես», «վեպ») ապահովել են բարձր

ճշգրտություն: Իրականացումը կատարվել է Python լեզվով՝ pandas, scikit-learn և NLTK գրադարանների կիրառմամբ, ինչը թույլ է տվել կառուցել տվյալների մշակման ամբողջական շղթա՝ նախնական փուլից մինչև մոդելի թեստավորում:

Ուսումնասիրությունը ցուցադրում է դասական ալգորիթմների արդյունավետությունը տվյալների պատշաճ պատրաստման դեպքում և ուրվագծում է հետագա հետազոտությունների զարգացման ուղղությունները. նեյրոնային ցանցերի ճարտարապետությունների (BERT, Transformers) ներդրում, անսամբլային մեթոդների օգտագործում և մոդելների տեղայնացում տարբեր լեզուների համար: Առաջարկվող մոտեցումը գործնական արժեք ունի թվային գրադարանների, առցանց խանութների և կրթական ռեսուրսների համար:

Classification of Literary Works by Genre Using Machine Learning Methods

Vardanyan Hayarpi,
Ohanyan Heghine

Summary

Key words: *TF-IDF vectorization method, classifier, SGDClassifier algorithm, metric, confusion matrix, digital libraries*

This article examines the problem of automated genre classification of literary works using machine learning methods. The relevance of the study is determined by the expansion of digital libraries and educational platforms, where manual processing of large text collections is inefficient. The proposed approach relies on TF-IDF vectorization with adapted stop-words for a multilingual corpus and a comparative evaluation of seven classification algorithms. The SGDClassifier demonstrated the best performance (accuracy 0.67), surpassing both Bayesian methods and ensemble models.

SGDClassifier demonstrated the highest performance among the tested algorithms (0.67 accuracy), indicating a moderate level of effectiveness. The result obtained is due to the semantic similarity of the genres and the limitations of classical models.

Error analysis revealed difficulties in distinguishing semantically close genres (“horror/mysticism” and “fantasy”), whereas specialized genres (“religion,” “business,” “novel”) achieved high classification accuracy. The implementation was carried out in Python using pandas, scikit-learn, and NLTK, enabling a complete data processing pipeline from preprocessing to model evaluation.

The study highlights the effectiveness of classical algorithms when applied to well-prepared corpora and identifies promising directions for future research, including the use of deep learning architectures (BERT, Transformers), ensemble methods, and model localization for different linguistic subsystems. The proposed methodology offers practical value for digital libraries, online book markets, and educational platforms.

Ներկայացվել է 01. 10. 2025 թ.
Գրախոսվել է 07. 10. 2025 թ
Ընդունվել է տպագրության 26. 11. 2025 թ.