Использование машинного обучения для определения контировок, исходя из экономического смысла закупочной документации

М.С. Межов

***Аннотация*—Крупные промышленные предприятия в ежедневных процессах своей деятельности генерируют, обрабатывают и хранят огромное количество документации, в том числе закупочной, которая требуется при взаимодействии с различными поставщиками необходимых товаров и услуг. В данной статье описывается подход к решению задачи автоматизации процесса определения бухгалтерской контировки исходя из экономического смысла закупочных документов с применением машинного обучения. При построении математической модели использовались реальные данные, собранные в экономическом отделе внешнеторговой компании промышленного сектора, в объёме 1020 документов, содержащих 183 различных вида контировки (класса). Приведено обоснование выбора метрики качества для оценки полученных в процессе работы моделей в условиях множественности классов и дисбаланса выборки данных.**

В рамках исследования было рассмотрено 14 различных алгоритмов машинного обучения. Наилучший результат показал алгоритм Ridge Classifier, который показал точность определения контировок на уровне 81%.

Изложенное решение может быть применено для задач автоматизации процесса определения контировок закупочной документации. Преимущество изложенного подхода заключается в развернутом описании решения задачи определения контировок, основываясь на экономическом смысле текста закупочных документов.

***Ключевые слова*—машинное обучение, обработка естественного языка, классификация документов, контировка, закупочная документация.**

I. Введение

Любая организация осуществляет закупочную деятельность, в которой все договорённости и обязательства фиксируется в закупочной документации: договорах, технических заданиях, счетах и др. В соответствии с Федеральным законом от 06.12.2011 № 402-ФЗ «О бухгалтерском учёте» любая хозяйственная операция должна быть отражена в бухгалтерском учёте [1]. Для отражения хозяйственной операции в бухгалтерском учёте используются различные числовые обозначения (коды) для классификации и группировки закупок. Одним из таких кодов является так называемый номер контировки. Его определяют исходя из экономического смысла документов, отражающих сделку. Контировка - принятые в бухгалтерском учёте обозначения дебетуемого и кредитуемого счетов и сумм в расчётных документах [2].

Важность автоматизации различных процессов в закупочной деятельности обусловлена их трудоёмкостью и значительным объёмом документации, которую необходимо создавать в соответствии с требованиями законодательства [3]. Автоматизация процессов позволяет снизить объём рутинных операций, повышает скорость протекания процессов и, как следствие, повышает эффективность деятельности.

Современные алгоритмы машинного обучения позволяют решать задачи анализа текстов, написанных на естественном языке, которые ранее невозможно было описать чёткими правилами алгоритма компьютерной программы, и были под силу только человеку [4], [5]. В ряде работ [6]-[10] рассматриваются схожие методы классификации текстов, однако применительно к закупочным документам и специфичным для них контировкам работ обнаружить не удалось. Кроме того, в дополнение к результатам упомянутых работ в данной статье раскрыт одни из важных вопросов при подготовке данных – извлечение чистого текста из исходных файлов документов различных форматов, решена задача многоклассовой, а не бинарной классификации в отношении закупочных документов с дисбалансом классов.

Нередко для классификации текстов применяются нейронные сети, однако частое требование бизнес-заказчиков – интерпретируемость моделей, что достигается за счёт использования более простых моделей машинного обучения: линейных моделей, моделей на основе решающих деревьев и их ансамблей, моделей на основе наивного байесовского подхода и др.

В данной работе подтверждается гипотеза о применимости алгоритмов машинного обучения в решении задачи определения контировок по содержанию закупочных документов. Построена математическая модель на основе выборки закупочных документов за год, состоящей непосредственно из файлов документов и определённой для каждого

Статья получена 18 июля 2022 г.

Межов Максим Сергеевич ведущий эксперт ООО «Цифровые технологии и платформы», Москва, Российская Федерация (e-mail: maksim.mezhov@outlook.com)

документа контировки. Данный набор данных был сформирован экономической службой одного из предприятий атомной отрасли в процессе своей деятельности.

II. Постановка задачи и характеристика набора данных

С точки зрения бизнес-постановки задачу можно сформулировать следующим образом: разработать математическую модель, которая даёт рекомендации по выбору контировки исходя из экономического смысла текста документа: технического задания, договора или счёта на приобретение товарно-материальных ценностей или услуг. В терминах машинного обучения задачу можно сформулировать следующим образом: разработать модель классификации текстов с дисбалансом и множественностью классов. На практике часто встречаются именно такие ситуации, когда присутствует более двух классов и количество примеров для экземпляров представленных классов сильно различается, имеются доминирующие и мало представленные классы.

Краткая характеристика данных:

- данные структурированы в соответствии с номерами контировок. Для каждой контировки представлен один или несколько файлов закупочной документации в различных форматах: *.doc, *.docx, *.pdf, *.jpeg;
- в наборе данных содержится 1020 документов по 183 различным классам, каждый класс соответствует одной конкретной контировке;
- из них 154 контировки мало представлены с точки зрения документов-примеров закупочной документации. Для каждой контировки из 154 имеется менее 10 документов-примеров;
- 29 контировок имеют более чем 10 документов-примеров (рис. 1). Количество таких документов составляет 650 шт.

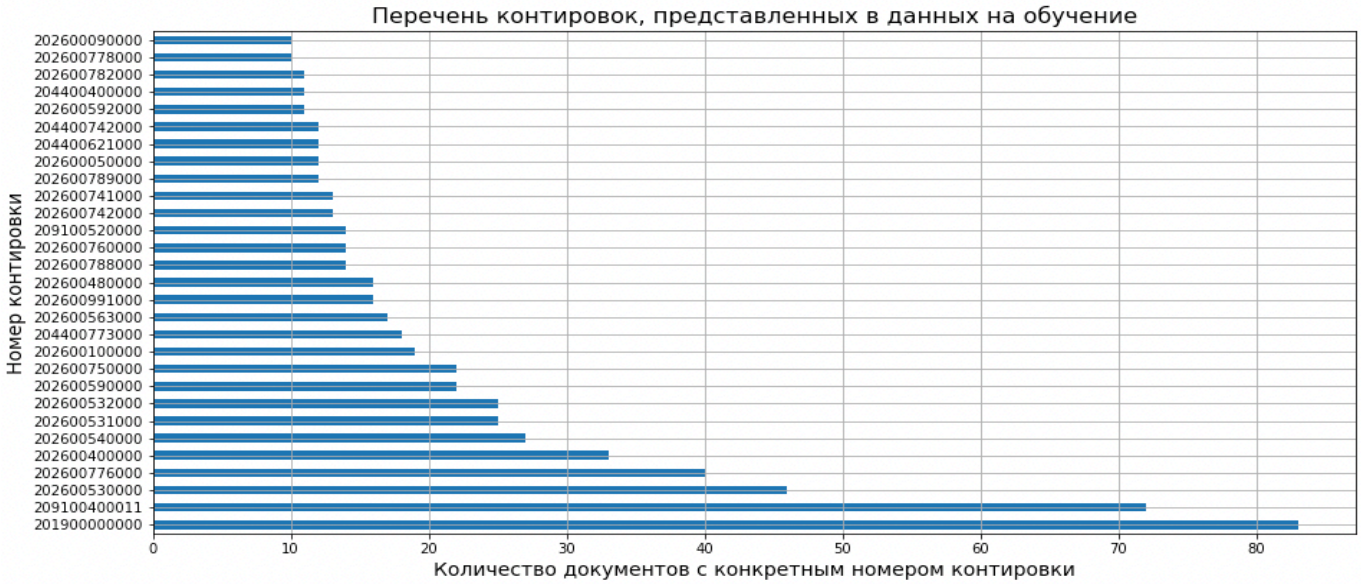

Рисунок 1. Перечень контировок, достаточно представленных в наборе данных

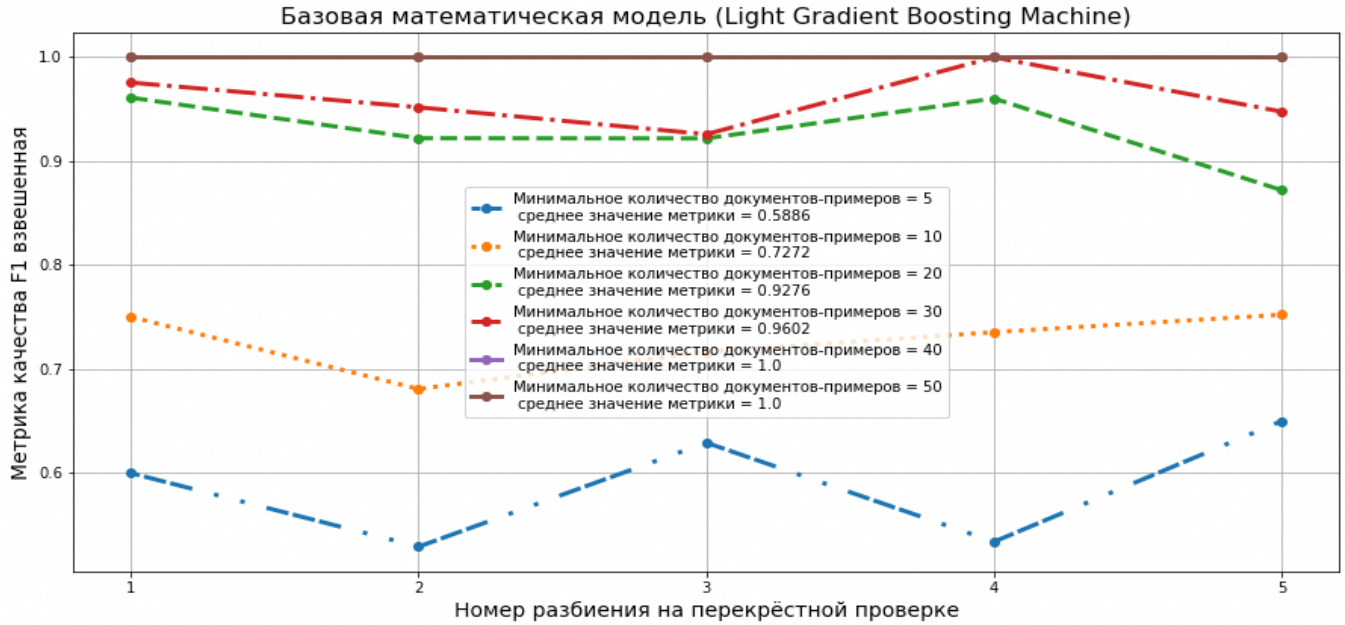

Рисунок 2. Оценка минимально необходимого количества документов-примеров

Для достижения приемлемого качества классификации каждой конкретной контировки необходимо иметь достаточное количество документов-примеров. Чтобы определить порог минимально необходимого количества таких примеров, был проведён дополнительный эксперимент с помощью базовой математической модели, построенной с использованием библиотеки LightGBM [11]. Порог по количеству документов-примеров был определён путём перекрёстной оценки качества классификации документов (рис. 2). При исключении из набора документов по контировкам, представленных менее чем десятью документами-примерами, получаем охват в 29 различных контировок и качество классификации 73%. При снижении границы до пяти – охват по контировкам возрастает до 46, но качество снижается до 59%. При увеличении границы до 30 – охват по контировкам существенно снижается до 3 и выглядит мало полезным на практике, однако качество классификации возрастает до 96%.

III. Векторное представление текста

Работа с текстом на естественном языке в машинном обучении связана с его векторным представлением. Для выявления похожих по смыслу документов необходимо применять статистические методы, благодаря котором можно представить имеющийся в документе текст в виде вектора чисел. Близкие по смыслу документы будут находиться рядом - будут иметь близкие по значению «координаты». Для того чтобы получить такую «координату» для каждого документа - векторное представление документа, необходимо выполнить следующие подготовительные шаги:

1) извлечь текст из всех имеющихся документов, несмотря на различие форматов и необходимости распознавания в случае со сканированными вариантами документов в таких форматах как *.pdf или *.jpeg. Для этой цели в данной работе использовалось программное обеспечение с открытым исходным кодом Tesseract OCR[1]. Для документов в формате *.doc и *.docx использовалась python-библиотека textract[2];

2) очистить извлечённый текст от пунктуации, чисел, предлогов и других часто встречающихся слов, т.к. они не помогают в конечном итоге разделять отличающиеся по смыслу документы, а лишь вносят «шум». Такие слова также называют стоп-словами. Перечень таких слов для русского языка можно заимствовать из библиотеки NLTK (Natural Language Toolkit)[3] языка программирования Python;

3) привести все слова к начальной форме, в которой они записываются в словарях, т.е. провести лемматизацию. Она важна ввиду того, что в тексте встречаются одни и те же слова, но в разных формах. Это обстоятельство увеличивает размерность векторного пространства документов, и снижает точность классификации. Для лемматизации использовалась программа MyStem[4], разработанная компанией Яндекс.

После прохождения всех подготовительных шагов, из всей коллекции документов получаем подготовленный корпус текстов, в который был помещён очищенный текст каждого документа из коллекции. На базе подготовленного корпуса необходимо подготовить матрицу (далее – матрица TFIDF, рис. 3), в которой по столбцам располагаются ключевые слова (далее – признаки), позволяющие наилучшим образом отделить разные по смыслу документы, а каждая строка представляет собой отдельный документ из коллекции. В ячейках данной матрицы записываются числовые значения, посчитанные на основе частоты вхождения каждого признака в конкретный документ (англ. – term-frequency, сокращённо - TF) и обратной частоты вхождения признака в коллекцию документов в целом (англ. – inverse document-frequency, сокращённо - IDF). TF отражает насколько часто слово встречается в каждом конкретном документе, а IDF – насколько редко слово встречается во всей коллекции документов. Другими словами, в данном подходе поощряется выбор тех слов, которые достаточно часто встречаются в тексте конкретных документов, но редко сразу во всей коллекции. Числовое значение для каждого конкретного слова, выбранного в качестве признака, и каждого документа вычисляется путём умножения TF на IDF. Данный метод имеет название TF-IDF [12]:

$$TFIDF(w, d, D) = TF(w, d) \times IDF(w, D) \tag{1}$$

где w – конкретное слово, d – конкретный документ, D – коллекция всех документов в корпусе.

автогражданский	автодор	автодорога	автомат	автоматизация	...	яркость	ярлык	яроо	ярославль	ярославский
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
0.0	0.0	0.0	0.0	0.004078	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.000000	...	0.0	0.016769	0.0	0.0	0.0

Рисунок 3. Пример матрицы TFIDF

[1] https://tesseract-ocr.github.io
[2] https://pypi.org/project/textract/
[3] https://www.nltk.org
[4] https://yandex.ru/dev/mystem/

Как видно из рис. 3 матрица TFIDF разреженная. Это возникает вследствие того, что не все ключевые слова присутствуют в документах, что логично.

Рассмотрим, как вычисляются компоненты *TF* и *IDF* из (1):

$$TF(w,d) = \log\left(1 + \frac{\text{количество слов } w \text{ в документе } d}{\text{общее количество слов в документе } d}\right),$$

$$IDF(w,D) = \log\left(\frac{\text{общее количество документов в корпусе } D}{\text{количество документов, в которых встречается слово } w}\right).$$

Автоматизировать процесс подготовки матрицы TFIDF можно с помощью библиотеки Scikit-learn[5] языка Python.

IV. Описание решения

Имея подготовленную матрицу TFIDF – векторное представление каждого документа из коллекции, можно использовать алгоритмы машинного обучения для выявления взаимосвязей, характеризующих принадлежность документов к конкретным контировкам. С этой целью необходимо дополнить матрицу дополнительным столбцом – номером контировки, которая соответствует каждому конкретному документу. Номер контировки мы можем извлечь из наименования папки, в которую были помещены документы, относящиеся к ней, на этапе подготовки набора данных. Таким образом, мы получаем подготовленный набор данных, содержащий все документы коллекции, представленные в векторном виде, и сопоставленные им номера контировок. Такой набор данных подходит для использования в алгоритмах машинного обучения.

Несмотря на то, что номера контировок – это целое число вида 201600012900 мы будем их воспринимать как наименование (метку) класса.

В итоговом наборе данных имеются документы, представляющие 29 различных классов - номеров контировок, а также имеется дисбаланс, т.к. не все классы равно представлены по количеству документов-примеров (рис. 1). В задачах многоклассовой классификации необходимо оценивать качество прогнозирования каждого класса. В этой связи часто используемая метрика качества Accuracy не годится, т.к. оценивает качество прогнозирования всего набора в целом и может быть применена для множественной классификации только в случае когда все классы имеют равное количество примеров (отсутствует дисбаланс).

$$Accuracy = \frac{\text{Количество верно определённых контировок}}{\text{Общее количество оцениваемых документов}}$$

В задачах множественной классификации с дисбалансом разумно использовать гармоническое среднее между точностью и полнотой – F1 меру, взвешенную на количество представленных документов-примеров для каждого класса.

$$F1 = 2 \times \frac{\text{точность} \times \text{полнота}}{(\text{точность} + \text{полнота})}$$

где точность – это доля документов, по которым были верно определены контировки на всём наборе данных;
полнота – это доля контировок, определённых верно, из общего числа контировок различных классов.

$$F1_{\text{взвешенная}} = \frac{\sum_{i=1}^{K}(F1_i \times N_i)}{N_{\text{общ}}}$$

где N_i – количество документов-примеров класса *i*,
$F1_i$ – $F1$ мера для класса *i*,
$N_{общ}$ – общее количество документов-примеров в наборе данных,
K – количество классов.

Для поиска алгоритма, который даст наилучший результат по метрике качества, было рассмотрено несколько алгоритмов и оценена их взвешенная метрика F1 на одном и том же наборе данных. Результаты приведены в табл. 1

Таблица 1. Оценка метрики качества моделей

Алгоритм, на базе которого построена математическая модель	Значение метрики качества F1 взвешенная
Ridge Classifier	0,8077
Light Gradient Boosting Machine	0,7844
Extra Trees Classifier	0,7813
Gradient Boosting Classifier	0,7739
Extreme Gradient Boosting	0,7702
Naive Bayes	0,7693
Random Forest Classifier	0,7659
Decision Tree Classifier	0,7228
Logistic Regression	0,6024
SVM-Linear Kernel	0,5753
K Neighbors Classifier	0,5559
Linear Discriminant Analysis	0,5341
Quadratic Discriminant Analysis	0,2539
Ada Boost Classifier	0,2037

Наилучший результат показала модель, построенная с использованием алгоритма Ridge Classifier [13]. Результат перекрёстной проверки данной модели представлен на рис. 4. При перекрёстной проверке весь объём данных был разделён на 10 равных частей с сохранением баланса классов, т.е. стратифицированно. Необходимо чтобы баланс классов в каждой отдельной

[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

части был такой же, как во всём наборе данных. На 9 частях проходило обучение модели, а на 1/10 части оценивалась метрика качества. Таким образом, мы проверяем качество модели на всех имеющихся данных, не обучая и тестируя модель на одной и той же части данных одновременно.

Рисунок 4. График оценки модели Ridge Classifier на кросс-валидации

На рис. 4 видно, что нижняя граница качества модели находится на уровне 70% на разбиении под № 5, в остальных вариантах разбиения видим, что качество находится выше 75%. Средние значения метрики отражает более адекватную оценку качества по всем разбиениям: $F1_{mean}$ = 0.8077

$$F1_{mean} = \frac{\begin{pmatrix} 0.8335 + 0.8778 + \\ 0.8898 + 0.7813 + \\ 0.7026 + 0.7623 + \\ 0.7796 + 0.7935 + \\ 0.7667 + 0.8904 \end{pmatrix}}{10} = 0.8077$$

Таким образом, можно говорить о 81% точности модели, понимая, что данная оценка была сделана с учётом каждого номера контировки и дисбаланса в данных по количеству представленных документов на каждый класс. Достигнутое значение метрики качества превосходит установленный на старте проекта критерий успешности в 75% точности определения контировки по тексту закупочных документов.

V. Заключение

Описанное решение иллюстрирует возможность применения машинного обучения для автоматизации процесса определения контировки закупочной документации по экономическому смыслу документа. Иллюстрирует подход к предварительной обработке документов в исходных форматах. Описывает подходящий для подобных задач способ векторизации текста. Производится сравнение нескольких алгоритмов машинного обучения, а также аргументируется выбор наилучшего на основе метрики качества, учитывающую специфику задачи: множественность и дисбаланс классов.

Разработанное решение позволило повысить более чем на 40% скорость процесса формирования аналитик для отражения первичных документов в системе учёта ресурсов в одной из внешнеторговых компаний промышленного сектора. Имеет потенциал к тиражированию. Является импортонезависимым, т.к. для его создания были использованы программные средства с открытым исходным кодом. Решение может использоваться как самостоятельный программный продукт или же быть интегрировано в имеющуюся ИТ-инфраструктуру предприятия.

Благодарности

Выражаю благодарность коллегам, которые внесли свой вклад в реализацию данного исследования:

- Ивану Максимову и Юрию Кацеру – за консультации в области машинного обучения,
- Максиму Лукичёву – за помощь в обработке исходного массива данных,
- Олегу Березину – за реализацию веб-приложения для проведения опытной эксплуатации разработанной математической модели,
- Нине Брусенцовой – за организационную поддержку исследования.

Библиография

[1] Федеральный закон «О бухгалтерском учёте» от 06.12.2011 № 402-ФЗ, статья 5. «Объекты бухгалтерского учета». Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_122855/191340d29485de342c21500874ade8ee79843bef/

[2] Райзберг Б. А. Современный экономический словарь / Б. А. Райзберг, Л. Ш. Лозовский, Е. Б. Стародубцева. – 2-е изд., испр. М.: ИНФРА-М, 1999. – 479 с.

[3] Федеральный закон "О закупках товаров, работ, услуг отдельными видами юридических лиц" от 18.07.2011 № 223-ФЗ». Режим доступа: https://www.consultant.ru/document/cons_doc_LAW_116964/
[4] Рудак Л. В. Анализ подходов к решению проблемы понимания и обработки естественного языка методами машинного обучения / Л. В. Рудак, О. И. Федяев // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2021) : Материалы VII Международной научно-технической конференции, Донецк, 23 ноября 2021 года / Под общей редакцией В. Н. Павлыша. – Донецк: Донецкий национальный технический университет, 2021. – С. 216-224
[5] Batrinca B., Treleaven P. Social media analytics: a survey of techniques, tools and platforms // Department of Computer Science, University College London. 2014
[6] Желябин Д. В. Применение методов машинного обучения для решения задачи NLP классификации текста на основе анализа семантики естественного языка / Д. В. Желябин // Вестник Алтайской академии экономики и права. – 2020. – № 6-2. – С. 229-235. – DOI 10.17513/vaael.1187
[7] Туманова А. Д. Методы машинного обучения в задачах автоматической обработки текстов на естественном языке / А. Д. Туманова, Н. С. Лагутина // Заметки по информатике и математике : Сборник научных статей. – Ярославль : Ярославский государственный университет им. П.Г. Демидова, 2018. – С. 174-181
[8] Иванова А. В. Исследование методов обработки текстовой информации и обзор этапов создания модели искусственного интеллекта при создании чат-ботов / А. В. Иванова, А. А. Кузьменко, Р. А. Филиппов [и др.] // Автоматизация и моделирование в проектировании и управлении. – 2021. – № 2(12). – С. 19-23. – DOI 10.30987/2658-6436-2021-2-19-23
[9] Никулин В. В. Применение методов машинного обучения для автоматизированной классификации и маршрутизации в библиотеке ITIL / В. В. Никулин, С. Д. Шибайкин, М. С. Соколова // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2022. – № 1. – С. 42-52. – DOI 10.24143/2073-5529-2022-1-42-52
[10] Евченко И. В. Управление процессом разработки мобильных игр на основе показателя проблем производства и применения обработки естественного языка / И. В. Евченко, Е. Д. Розинко, Е. А. Моргачева // International Journal of Open Information Technologies. – 2022. – Т. 10. – № 4. – С. 84-88
[11] LightGBM: A Highly Efficient Gradient Boosting Decision Tree. – Режим доступа: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/11/lightgbm.pdf
[12] Rajaraman, A. "Data Mining". Mining of Massive Datasets / A. Rajaraman, J. Ullman: Cambridge University Press, 2011. – pp. 1–17
[13] Ridge regression and classification. Режим доступа: https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression

М. С. Межов родился 13 марта 1987 г. В 2009 году окончил Ивановский Государственный энергетический университет по специальности «Прикладная математика и информатика».
12 лет работал в атомной отрасли: Инженером - программистом, Руководителем отдела, Исследователем данных (Data Scientist). С июня 2022 г. работает исследователем данных в компании по цифровизации химической промышленности (ул. Дубининская, д. 53, стр. 6, 115054, Москва, Российская Федерация). Интересы исследований: предиктивная аналитика, обработка естественного языка, компьютерное зрение.
М. Межов является финалистом Всероссийского соревнования для профессионалов в сфере цифровой экономики «Цифровой прорыв» 2021 года, занял первое место на III отраслевом чемпионате в сфере информационных технологий по стандартам WorldSkills «DigitalSkills 2021» в компетенции «Машинное обучение и большие данные».

https://www.linkedin.com/in/maksimmezhov

Using machine learning to determine accounting codes based on the economic meaning of procurement documentation

Maksim M. Mezhov

***Annotation*—Large industrial enterprises in the daily processes of their activities generate, process, and store a huge amount of documentation, including procurement, which is required when interacting with various suppliers of necessary goods and services. This article describes an approach to solving the problem of automating the process of determining accounting codes based on the economic meaning of procurement documents using machine learning. The real data collected in the economic department of a trade industrial sector company were used in the volume of 1020 documents containing 183 different types of accounting code (class). The rationale of a quality metric for estimate developed models is given.**

The study examined 14 different machine learning algorithms. The best result was shown by the Ridge Classifier algorithm, which showed an accuracy of 81% in determining the accounting codes.

The advantage of the above approach is a detailed description of the solution to the problem of determining the accounting code, based on the economic meaning of the text of the procurement documents.

***Keywords*—machine learning, natural language processing, document classification, accounting code, procurement documentation.**

REFERENCES

[1] Federal'nyi zakon «O bukhgalterskom uchete» ot 06.12.2011 № 402-FZ, stat'ya 5. «Ob"ekty bukhgalterskogo ucheta». Available: https://www.consultant.ru/document/cons_doc_LAW_122855/191340d29485de342c21500874ade8ee79843bef/

[2] Raizberg B. A. Sovremennyi ekonomicheskii slovar' / B. A. Raizberg, L. Sh. Lozovskii, E. B. Starodubtseva. – 2-e izd., ispr. M.: INFRA-M, 1999. – p. 479.

[3] Federal'nyi zakon "O zakupkakh tovarov, rabot, uslug otdel'nymi vidami yuridicheskikh lits" ot 18.07.2011 № 223-FZ". Available: https://www.consultant.ru/document/cons_doc_LAW_116964/

[4] Rudak L. V. Analiz podkhodov k resheniyu problemy ponimaniya i obrabotki estestvennogo yazyka metodami mashinnogo obucheniya / L. V. Rudak, O. I. Fedyaev // Sovremennye informatsionnye tekhnologii v obrazovanii i nauchnykh issledovaniyakh (SITONI-2021) : Materialy VII Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii, Donetsk, 23 noyabrya 2021 goda / Pod obshchei redaktsiei V. N. Pavlysha. – Donetsk: Donetskii natsional'nyi tekhnicheskii universitet, 2021. – pp. 216-224

[5] Batrinca B., Treleaven P. Social media analytics: a survey of techniques, tools and platforms // Department of Computer Science, University College London. 2014

[6] Zhelyabin D. V. Primenenie metodov mashinnogo obucheniya dlya resheniya zadachi NLP klassifikatsii teksta na osnove analiza semantiki estestvennogo yazyka / D. V. Zhelyabin // Vestnik Altaiskoi akademii ekonomiki i prava. – 2020. – № 6-2. – pp. 229-235. – DOI 10.17513/vaael.1187

[7] Tumanova A. D. Metody mashinnogo obucheniya v zadachakh avtomaticheskoi obrabotki tekstov na estestvennom yazyke / A. D. Tumanova, N. S. Lagutina // Zametki po informatike i matematike : Sbornik nauchnykh statei. – Yaroslavl' : Yaroslavskii gosudarstvennyi universitet im. P.G. Demidova, 2018. – pp. 174-181

[8] Ivanova A. V. Issledovanie metodov obrabotki tekstovoi informatsii i obzor etapov sozdaniya modeli iskusstvennogo intellekta pri sozdanii chat-botov / A. V. Ivanova, A. A. Kuz'menko, R. A. Filippov [i dr.] // Avtomatizatsiya i modelirovanie v proektirovanii i upravlenii. – 2021. – № 2(12). – pp. 19-23. – DOI 10.30987/2658-6436-2021-2-19-23

[9] Nikulin V. V. Primenenie metodov mashinnogo obucheniya dlya avtomatizirovannoi klassifikatsii i marshrutizatsii v biblioteke ITIL / V. V. Nikulin, S. D. Shibaikin, M. S. Sokolova // Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika. – 2022. – № 1. – pp. 42-52. – DOI 10.24143/2073-5529-2022-1-42-52

[10] Evchenko I. V. Upravlenie protsessom razrabotki mobil'nykh igr na osnove pokazatelya problem proizvodstva i primeneniya obrabotki estestvennogo yazyka / I. V. Evchenko, E. D. Rozinko, E. A. Morgacheva // International Journal of Open Information Technologies. – 2022. – T. 10. – № 4. – pp. 84-88

[11] LightGBM: A Highly Efficient Gradient Boosting Decision Tree. – Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/11/lightgbm.pdf

[12] Rajaraman, A. "Data Mining". Mining of Massive Datasets / A. Rajaraman, J. Ullman: Cambridge University Press, 2011. – pp. 1–17

[13] Ridge regression and classification. Available: https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression