**УДК 004.852**
**МРНТИ 28.23.25**

https://doi.org/10.55452/1998-6688-2022-19-1-30-43

# СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ КЛАССИФИКАЦИИ ДАННЫХ ПРИ ПРОГНОЗИРОВАНИИ ЦЕН TRADE-IN АВТО

**АСУБАЕВА Е.М., АБДИАХМЕТОВА З.М.**

*Казахский национальный университет имени аль-Фараби,*
*050040, г. Алматы, Казахстан*

***Аннотация.*** *В статье реализованы и проанализированы алгоритмы машинного обучения для предсказания цен авто. Предсказание цен – одна из самых сложных, но интересных задач. В предсказании задействовано много факторов – год выпуска, состояние, пробег, объем двигателя и т.д. Эти аспекты в совокупности влияют на цены авто, делая их нестабильными и затрудняя прогнозирование с высокой степенью точности. Методы машинного обучения могут выявить закономерности и идеи, которые мы раньше не видели, и их можно использовать для безошибочно точных прогнозов и классификации данных. Выбор надлежащего алгоритма классификации данных, который подходил бы для отдельно взятой задачи, зависит от объема, качества и природы данных, от вычислительных ресурсов компьютера, а также от того, как вы планируете использовать результат. Каждый алгоритм классификации имеет свои особенности и основывается на определенных допущениях. В конечном счете качество классификатора, его вычислительная и предсказательная мощность зависят от базовых данных, предназначенных для тренировки алгоритма. Цель данной статьи – рассмотреть этапы предварительной обработки тренировочных данных и показать, как машинное обучение в частности и информационные технологии в целом преуспели в разработке инструментов для моделирования, проектирования, прогнозирования, планирования и поддержки принятия решений в области продажи авто. В данном исследовании предлагается гибридный подход к задачам прогнозирования, то есть к решению задач прогнозирования с применением методов статистического анализа и машинного обучения.*

***Ключевые слова:*** *машинное обучение, задача классификации, логистическая регрессия, случайный лес, дерево принятия решений, k-ближайших соседей, RESTAPI.*

# TRADE-IN АВТО БАҒАЛАРЫН БОЛЖАУ КЕЗІНДЕ ЖІКТЕУ ӘДІСТЕРІН САЛЫСТЫРМАЛЫ ТАЛДАУ

**АСУБАЕВА Е.М., АБДИАХМЕТОВА З.М.**

*әл-Фараби атындағы Қазақ ұлттық университеті,*
*050040, Алматы қ., Қазақстан*

***Аңдатпа.*** *Мақалада автокөлік бағасын болжау үшін машиналық оқыту алгоритмдері енгізілген және талданған. Бағаны болжау – күрделі, бірақ қызықты тапсырмалардың бірі. Болжауға көптеген факторлар қатысады – шығарылған жылы, жағдайы, жүрісі, қозғалтқыш көлемі және т.б. Барлық осы аспектілер автокөлік бағасын тұрақсыз етеді және жоғары дәлдікпен болжауды қиындатады. Машиналық оқыту әдістерін бұрын көрмеген үлгілер мен идеяларды ашып және оларды дәл болжау мен жіктеу үшін қолдануға болады. Берілген тапсырмаға сәйкес келетін деректерді жіктеу әдісін таңдау – деректердің көлеміне, сапасына және сипатына, компьютердің есептеу ресурстарына және нәтижені қалай пайдалану жоспарларына байланысты. Әрбір жіктеу алгоритмінің өзіндік ерекшеліктері бар және ол белгілі болжамдарға негізделген. Бұл мақаланың мақсаты – оқыту деректерін алдын ала өңдеу кезеңдерін қарастыру және атап айтқанда, машиналық оқыту және*

*тұтастай алғанда ақпараттық технологиялар автомобиль саласында модельдеу, жобалау, болжау, жоспарлау және шешімдерді қолдау құралдарын әзірлеуде қалай табысқа жеткенін көрсету. Бұл зерттеу есептерді болжаудың гибридті тәсілін ұсынады, яғни статистикалық талдау және машиналық оқыту әдістерін пайдалана отырып болжау мәселелерін шешу.*

***Түйінді сөздер:*** *машиналық оқыту, жіктеу мәселелері, логистикалық регрессия, кездейсоқ орман, шешім ағашы, k-жақын көршілер, RESTAPI.*

## COMPARATIVE ANALYSIS OF DATA CLASSIFICATION METHODS FOR PREDICTION OF TRADE-IN AUTO PRICES

**ASSUBAYEVA Y.M., ABDIAKHMETOVA Z.M.,**

*Al-Farabi Kazakh National university,050040, Almaty, Kazakhstan*

***Abstract.*** *This article implements and analyzes machine-learning algorithms, for predicting carsprices. Predicting prices is one of the most challenging but interesting tasks. There are so many factors involved in the prediction - year of manufacture, condition, mileage, engine size, etc. All these aspects combine to make auto prices volatile and very difficult to predict with a high degree of accuracy. Machine learning techniques can uncover patterns and ideas that we have not seen before, and can be used to predict and classify data accurately and accurately. The choice of the proper data classification algorithm, which would be suitable for a given task, depends on the volume, quality and nature of the data, on the computing resources of the computer, and how you plan to use the result. Each classification algorithm has its own characteristics and is based on certain assumptions. Also requires practical skills. In practice, it is always recommended to compare the quality of at least several different learning algorithms in order to choose the best model for a particular task, since the most experienced data scientists will not be able to tell which algorithm is more efficient. Algorithms can differ in the number of features or samples, the noise level in the dataset, and whether the classes are linearly separable or not. Ultimately, the quality of the classifier, its computational and predictive power, depends on the underlying data intended for training the algorithm. The purpose of this article is to consider the stages of pre-processing training data, and show how machine learning in particular and information technology in general have succeeded in developing tools for modeling, designing, predicting, planning and decision support in the field of auto sales. This study proposes a hybrid approach to forecasting problems, that is, solving forecasting problems using statistical analysis and machine learning methods.*

***Keywords:*** *machine learning, classification problems, logistic regression, random forest, decision tree, k-nearest neighbor, REST API.*

**Введение**

Машинное обучение (ML) стало одной из самых захватывающих и прорывных технологий современности [1, 2]. Такие крупные компании, как Google, Apple, Microsoft, Amazon и другие, вкладывают значительный капитал в разработку методов и приложений, в эту область исследования, открывая путь к новым возможностям. Например, когда приложение Kaspibank принимает решение по одобрению кредита или когда Netflix рекомендует фильм, который может вам понравиться, разговоры с речевыми ассистентами по смартфону происходят с помощью алгоритмов машинного обучения.

Работая в сфере продаж новых легковых и легких коммерческих автомобилей, мы столкнулись с такой глобальной проблемой, как спад производства, и новыми проблемами в логистике, связанными с разрывом цепочек поставок. Ключевой проблемой для автопрома с лета 2020 г. остается дефицит электронных компонентов, из-за чего автозаводы вынуждены сокращать выпуск машин и уходить в простои. Это привело к нехватке автомобилей и росту цен на новые легковые машины. В сравнении с октябрем 2020 г. в 2021 г. продажи упали на 18.1%. Аналитики утверждают, что автопрому предстоит еще пройти долгий путь, чтобы преодолеть сложившийся кризис. Поэтому руководство ООО «Р-Моторс ЛАДА» приняло решение компенсировать спад

продаж новых авто за счет выкупа вторичного авто для дальнейшей перепродажи.

Если в ценообразование на первичном рынке автомобилей входит логистика, налоги, желаемая прибыль дилера и зарплата цепочки его сотрудников, то факторы формирования стоимости цен на trade-in авто куда более обширные. Поэтому важно максимально объективно оценить состояние машины и в соответствии с этим выставить стоимость, принимая во внимание такие показатели, как:

– год выпуска авто;

– техническое состояние и состояние кузова;

– пробег;

– особенности комплектации;

– время продажи (даже сезон, в который авто выставляется на реализацию, оказывает влияние на спрос и, соответственно, стоимость);

– востребованность модели на рынке;

– сервисная история.

Традиционный подход к ценообразованию полностью опирается на слово эксперта, который принимает решение только на основе своего опыта.

Машинное обучение задействует сложные алгоритмы для того, чтобы учитывать множество факторов и устанавливать правильные цены для тысячи продуктов практически за секунды [4]. Модели ценообразования на базе машинного обучения определяют паттерны полученных данных, что дает возможность определять цены с учетом факторов, о которых менеджер по выкупу мог даже не догадываться.

На практике всегда рекомендуется сравнить качество нескольких разных алгоритмов обучения, чтобы выбрать наилучшую модель для отдельно взятой задачи, так как даже самые опытные специалисты по обработке и анализу данных не смогут сказать, какой алгоритм эффективнее [3]. Алгоритмы могут отличаться по числу признаков либо образцов, уровню шума в наборе данных и по тому, являются классы линейно разделимыми или нет. В рамках этой статьи будут рассмотрены такие методы классификации, как логистическая регрессия, случайный лес, дерево принятия решений, k-ближайших соседей, для прогнозирования цен на подержанные авто с использованием технологии машинного обучения.

**Материалы и методы исследования**

Задача классификации является подкатегорией методов машинного обучения с учителем, цель которой заключается в определении категориальных меток классов для следующих экземпляров на основе исторических наблюдений [5]. Здесь определение «с учителем» относится к коллекции образцов, в которых нужные метки принадлежности к классам уже известны. При обучении с учителем извлекается модель на основе алгоритмов классификации и из маркированных тренировочных данных, которая позволяет делать прогнозы о ранее не встречавшихся или будущих данных [6].

Другая подкатегория методов обучения с учителем представляет регрессия, где результат – непрерывная величина. Метки в классификации могут иметь двоичную природу, к примеру фильтрация почты на спам и не спам. Типичным примером многоклассовой классификации является рукописное распознавание символов.

Существует множество методов классификации с различными подходами при реализации. Каждый алгоритм имеет свои особенности и основывается на определенных допущениях. В конечном счете качество классификатора, процент точности предсказания зависят от тренировки алгоритма. Во время тренировки алгоритма задействуются такие шаги, как отбор признаков, выбор качественной метрики, выбор классификатора и алгоритмов оптимизации, оценка качества модели, тонкая настройка алгоритма.

Классификаторы на основе алгоритма деревьев принятия решений (DecisionTrees, DT) [14] представляют собой иерархическую древовидную структуру (подмножества), которая образовалась путем принятия решений, основываясь на постановке ряда вопросов [6]. Дерево содержит корень, откуда идет разбиение данных по признаку, тем самым генерируя правила, что ведет к приросту информации (Information Gain, IG). Процесс разбиения данных повторяется в каждом дочернем узле (Node) в зависимости от условия разветвления до тех пор, пока не получится результат прогнозирования (однородный лист). Для оценки качества разветвления можно использовать такие показатели, как коэффициент Джини или среднеквадратическая ошибка (MSE).У каждого узла столько ветвлений, сколько значений имеет выбранный признак. На практике результат может привести к образованию глубоких деревьев, что является признаком переобучения. Чтобы избежать этого, рекомендуется устанавливать пределы максимальной глубины. Существуют множества библиотек, где можно визуализировать результат таких деревьев принятия решений.

Целевая функция алгоритма на основе дерева определяется следующим образом:

$$IG(D_p, f) = I(D_p - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j), \quad (1)$$

где $f$ – это признак, по которому выполняется расщепление, $D_p$ и $D_j$ – набор данных родительского и j-го дочернего узла, I – мера неоднородности, $N_p$ – общее число образцов в родительском узле и $N_j$ – число образцов в j-м дочернем узле [1].

Деревья решений могут создавать сложные границы решения путем деления пространства признаков на прямоугольники. Чтобы избежать глубоких деревьев, в библиотеке scikit-learn предусмотрена возможность указывать максимальную глубину. Таким образом, можно легко натренировать дерево, обходя сложные границы решения.

Логистическая регрессия [17] – один из простых и одновременно мощных алгоритмов для задач линейной и бинарной классификации. Несмотря на название этого метода, логистическая регрессия – это модели задачи классификации, а не регрессии. Модель с динамичным обучением стохастического градиентного спуска позволяет прогнозировать вероятность отдельно взятого события [7]. Алгоритм статистическим методом предсказания событий максимизирует условные вероятности тренировочных данных, делая ее более подверженной выбросам. Практическая ценность заключается в том, что модель легче реализовать, чем модели на основе опорных векторов (SVM). Методы SVM главным образом сосредоточены на точках, ближайших к границе решения. Кроме того, модели логистической регрессии можно легко обновлять, упрощая работу с потоковой передачей данных. Однако модель не лишена метода регуляризации для обеспечения предотвращения переобучения, фильтрации шума из данных. В основе регуляризации лежит идея внесения дополнительной информации для наложения штрафа на экстремальные веса параметров. Стандартной формой регуляризации является L2-регуляризация весов, которую можно записать следующим образом [3]:

$$\frac{\Lambda}{2}\|\omega\|^2 = \frac{\Lambda}{2}\sum_{n}^{m} \omega_j^2 \quad (2)$$

Здесь $\Lambda$ – это параметр регуляризации лямбда. Регуляризация является еще одним аргументом в пользу важности масштабирования признаков, таких как стандартизация. Чтобы регуляризация работала должным образом, необходимо обеспечить сопоставимость весов.

Алгоритм случайного леса (randomforest) [14] – еще один пример классификатора с учителем, который используется также и для регрессии, приобрел популярность в ML в таких задачах, как механизмы рекомендаций, классификация изображений, за счет своей простоты использования, классификационной способности и меньшей восприимчивости к переобучению. Интуитивно лес принятия решения можно рассматривать как объединение нескольких деревьев решений для достижения единого результата. Основная идея заключается в том, чтобы объединить слабые деревья для создания более устойчивой модели к выбросу данных.

Для назначения метки класса агрегируется прогноз из каждого дерева на основе голосов, т.е. наиболее частая категориальная переменная даст предсказанный класс. Каждое дерево в лесу решения классификации выводит гистограмму ненормализованной частоты меток с помощью голосования [5]. В ходе статистической обработки суммируются эти гистограммы и нормализуется результат для получения вероятностных значений для каждой метки. Деревья с высокой достоверностью прогноза имеют больший вес в окончательном принятии решения ансамблей.

Большое преимущество леса принятия решений в том, что не приходится переживать о переобучении, так как модель устойчива к шуму из отдельных деревьев решений. Как правило, чем больше число деревьев, тем выше качество классификатора на основе леса, достигаемое за счет вычислительной емкости [12].

Последний алгоритм, рассматриваемый в данной статье, – это классификатор на основе k-ближайших соседей (k-nearestneighborclassifier, KNN) [15]. Алгоритм интересен тем, что является примером ленивого обучения [10]. Классификатор получил такое название из-за своей очевидной простоты – он не извлекает различающую функцию из тренировочных данных, а вместо этого запоминает тренировочный набор данных.

Число k – это количество соседних объектов в пространстве признаков, которые сравниваются с классифицируемым объектом путем измерения расстояния. Для этого необходимо определиться с метриками расстояния. Метрики расстояния подбираются в зависимости от признака набора данных. Для образцов с вещественными

значениями часто используется простая евклидова мера. Основываясь на метрике расстояния, алгоритм KNN находит в тренировочном наборе данных k образцы, которые являются самыми близкими к классифицируемой точке. Например, если k=6, то каждый объект сравнивается с шестью соседями. В ходе обучения алгоритм улавливает идею сходства (расстояние) и запоминает все векторы признаков и соответствующие им метки классов. При работе с наблюдениями для меток класса, которые алгоритм еще не видел, вычисляется расстояние между вектором нового наблюдения и ранее запомненными. Затем выбирается k ближайших к нему векторов, и новый объект относится к классу, которому принадлежит большинство из них [3]. Правильный выбор числа k крайне важен для нахождения хорошего равновесия между переобучением и недообучением.

Машинное обучение является мощным и эффективным инструментом при реализации алгоритмов классификации, однако определяющее значение в этих процессах имеет качество исходных данных [9], так как качество данных и объем полезной информации являются ключевыми факторами, которые определяют, как хорошо алгоритм сможет обучиться. Следовательно, крайне важно сначала набор данных подвергнуть предварительной обработке и только потом подавать его на вход обучаемого алгоритма. Реальные наборы данных могут содержать пропущенные значения из-за отсутствия данных, операторской ошибки при заполнении и т.д. В параметрах некоторых моделей есть возможность указать игнорировать пропуски (use_missing = false). Лучшей стратегией было бы заполнить недостающие значения, чем избавляться от наблюдений, в которых отсутствуют данные, но стоит учесть, что выбор неудачного метода заполнения пропущенных значений не всегда приводит к улучшению результата прогнозирования. Именно поэтому проведение подготовки исходных данных, их предварительная обработка позволяют значительно повысить точность результатов, получаемых в ходе применения машинного обучения. В данном эксперименте создание хороших тренировочных наборов резюмировалось в пяти шагах.

Шаг первый: исключение признаков, которые не несут смысловой нагрузки для поступающего анализа. В данном дата-сете это id, vin код, ссылки на сайт, где можно подробно увидеть авто, координаты.

Шаг второй: импутация и удаление данных [12].

Это процесс замещения пропущенных, некорректных значений другими значениями. Один из наиболее распространенных методов интерполяции является импутацией простым средним значением всего признакового столбца. Для категориальных данных удобно заменять пропущенные значения самыми частотными (mostfrequent). Довольно часто используемый подход при работе с отсутствующими данными – это исключение записей (строк) или полей (столбцов), в которых встречаются пропуски (NaN). В крупных дата-сетах, чтобы увидеть количество пропущенных данных, можно воспользоваться методом sum по каждому столбцу. Один из самых простых способов исключить все объекты, которые содержат значения NaN (т.е. notanumber, не число), – метод dropna. Это приводит к сокращению объема данных и повышению его смысловой ценности. Однако этот метод несет в себе определенные недостатки; например, можно в конечном счете удалить слишком много образцов, которые сделают надежный анализ невозможным.

Шаг третий: корреляционный анализ. Является основой анализа статистических данных, цель которого заключается в определении наличия каких-либо значимых связей, закономерностей или тенденций. Итог такого анализа – коэффициент корреляции, который показывает, насколько сильна связь между двумя переменными в наборе данных [8]. Положительный результат корреляции означает, что обе переменные увеличиваются по отношению друг к другу, в то время как отрицательная корреляция означает, что по мере того, как одна переменная уменьшается, другая увеличивается. Применение корреляционного анализа позволяет исследователям определить, какие аспекты и переменные зависят друг от друга, результат которых может дать полезные сведения или отправную точку для дальнейших исследований и более глубокого понимания. Наглядно данную связь можно увидеть, построив тепловую карту плотности с помощью различных библиотек визуализации данных (рисунок 1, стр. 35). Интерпретация полученного результата – коэффициент корреляции, колеблется от -1 до +1. Если значение близко к +1, значит, существует не так много положительной корреляции, при -1 означает, что существует сильная отрицательная корреляция. Когда он близок к нулю, это означает, что корреляции нет.

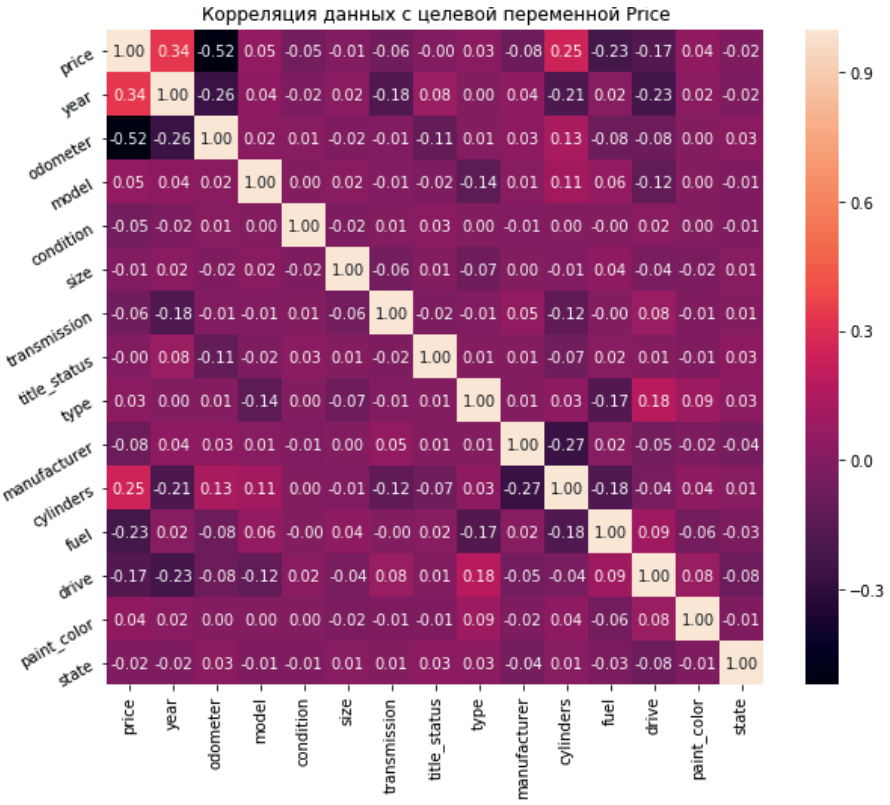


Рисунок 1 – Визуализация корреляционного анализа

Как видно на рисунке 1, коэффициент корреляции с целевой функцией низкий, что может привести к менее точному предсказанию. Тепловая карта будет более эффективной в представлении данных, если будут удалены избыточные данные, которые действуют на анализ данных как отвлекающий шум.

Шаг четвертый: избавиться от выбросов. Выбросы сильно отличаются от других наборов данных из-за изменчивости в измерениях или же в ходе ошибки ввода данных [8]. Если возможно, выбросы следует исключить из набора данных. Однако обнаружение этих аномальных экземпляров может быть трудным и не всегда возможным. Если признак численный, то можно построить гистограмму или коробчатую диаграмму (ящик с усами):

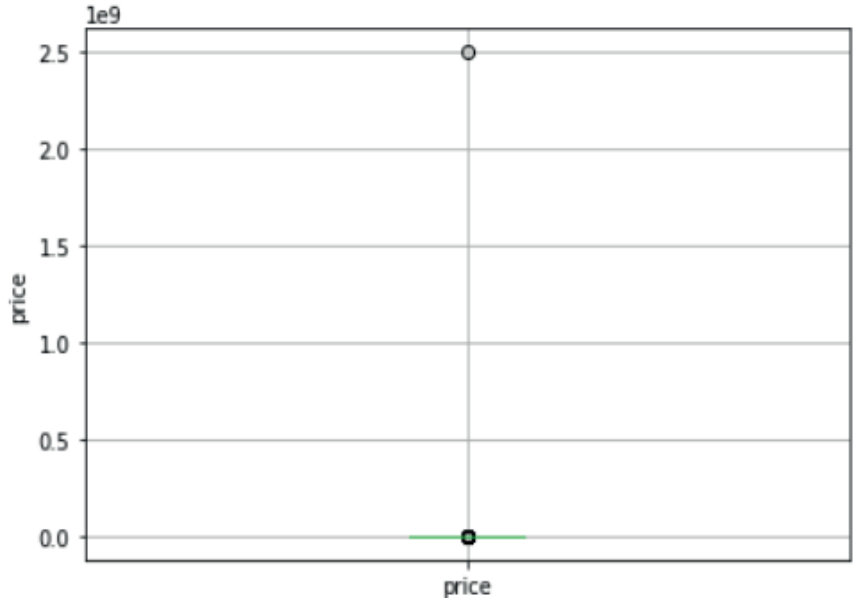


Рисунок 2 – Построение коробчатой диаграммы для определения выбросов в целевой переменной

Шаг пятый: обработка категориальных данных. Результат прогнозирования таких алгоритмов, как дерево решений, может быть получен непосредственно из категориальных данных без преобразования данных (это зависит от конкретной реализации). Когда алгоритмы как KNN не могут работать с категориальными данными напрямую, они требуют, чтобы все входные и выходные переменные были числовыми. Поэтому для кодирования меток классов использовался метод Label Encoder библиотеки scikit-learn, который однократно кодирует фиктивные переменные для категориальных данных. Затем можно применить словарь соответствий для преобразования меток классов в целые числа (таблица 1).

Таблица 1 – Словарь соответствия после присвоения меток

| № | Drive | Fuel | Color | Метка |
|---|---|---|---|---|
| 1 | FWD | Gas | Red | 0 |
| 2 | RWD | Diesel | White | 1 |
| 3 | AWD | Petrol | Black | 0 |
| 4 | 4WD | Electric | Gray | 1 |

При создании модели машинного обучения важно измерить результат работы модели. Обычно используемый метод измерения эффективности алгоритма классификации – это матрица неточностей (матрица истинности, confusionmatrix) [13]. Матрица неточностей отображает количество правильных прогнозов по сравнению с количеством неправильных прогнозов. В случае бинарного классификатора это будет количество истинных, ложных положительных, отрицательных результатов. Основываясь на этих числах, можно рассчитать некоторые значения, объясняющие производительность модели [12].

Точность (accuracy) – это мера того, сколько правильных прогнозов модель сделала для полного набора тестовых данных. Формула для вычисления точности выглядит следующим образом:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

где TN – истинно отрицательный, FP – ложно положительный, FN – ложно отрицательный, TP – истинно положительный результат.

Точность – хороший базовый показатель для измерения производительности модели. Обратной стороной простой точности является то, что точность хорошо работает в сбалансированных наборах данных. Однако в несбалансированных наборах данных точность становится худшим показателем.

AUC ROC (площадь под кривой ошибок) – это график, который суммирует характеристики модели двоичной классификации по положительному классу [9]. Ось X указывает частоту ложных положительных результатов, а ось Y показывает истинную положительную частоту. Оценивая истинно положительные и ложные срабатывания для различных пороговых значений, можно построить кривую, которая простирается от нижнего левого угла к верхнему правому и изгибается к верхнему левому углу. Эта кривая называется кривой ROC. В литературе иногда приводится следующая экспертная шкала для значений ROC AUC, по которой можно судить о качестве модели: 0.9-1.0 – отличное; 0.8-0.9 - очень хорошее; 0.7-0.8 – хорошее; 0.6-0.7 – среднее; 0.5-0.6 – неудовлетворительное.

**Научная новизна**

Научная новизна данной работы заключается в применении алгоритмов машинного обучения для расширения возможностей программного комплекса по перекупке trade-in авто. В частности, рассматриваются несколько этапов обработки данных, описаны результаты проведенных экспериментов и практическая значимость исследований. Обосновано использование разработанного метода для проведения оценки в вопросах ценообразования.

**Результаты и обсуждения**

Данные для обучения были взяты с сервиса Haraba [18]. Это база объявлений подержанных автомобилей со всей России с 2017 г. Обмен данными с сервисом Haraba осуществляется с помощью архитектуры RESTAPI [19]. Для этого написана служба Windows Service [20], задача которой состоит в том, чтобы каждые 10 минут отправлять запрос в Haraba для получения новых объявлений:

```
private void timer_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    Logger.Log.Info("timer_Elapsed");
    _timer.Stop();
    try
    {
        Logger.Log.Info("try to get data from xaraba");

        var httpRequest = (HttpWebRequest)WebRequest.Create(url);
        httpRequest.ContentType = "text/plain";
        httpRequest.Timeout = System.Threading.Timeout.Infinite;
        var httpResponse = (HttpWebResponse)httpRequest.GetResponse();
        var streamReader = new StreamReader(httpResponse.GetResponseStream());

        var result = streamReader.ReadToEnd();

        Data jresResponse = JsonConvert.DeserializeObject<Data>(result.Trim());
        Logger.Log.Info("получили " + jresResponse.TotalCars + " записей ");
        if (jresResponse.TotalCars > 0)
        {
            Logger.Log.Info("try to insert log");
            query = string.Format("insert into rrt_global_log.dbo.Data_Xaraba_log
                result );
            ExecuteQuery.SQL_Exec_Scalar(query, connectionStr);
            foreach (var item in jresResponse.Results)
            {
                InsertDataToLine(item);
            }
        }
    }
    catch (Exception ex)
    {
        Logger.Log.Error("insert exception: " + ex);
        Send(ex.Message);
    }
    _lastRun = DateTime.Now;
    _timer.Start();
}
```

Рисунок 3 – Windows Service для обмена данными с сервисом Haraba

Запросив все исторические данные из сервиса, получаем дата-сет, который прошел 5 этапов обработки данных, что описаны выше. Во время эксперимента использовалась пропорция 80:20, таким образом разделив набор на тренировочные и тестовые данные. Задача – обучить модели анализировать каждый фактор, который влияет на ценообразование, и выбрать самую оптимальную среди 4 рассматриваемых алгоритмов.

Язык программирования – Python, т.к. у него есть множество фреймворков, которые упрощают процесс написания кода и сокращают время на разработку и анализ данных.

```
#импорт библиотек
import matplotlib.pyplot as plt  #для графика
import pandas as pd #для работы с датасетом
from sklearn import preprocessing # для кодировки категориальных переменных
from sklearn.model_selection import train_test_split #для разбиения выборки на обучающие и тестовые
import seaborn as sns # для визуализации данных
######библиотеки для работы с моделями###############
from sklearn import tree
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import RandomForestClassifier
#####библиотеки для оценки качества моделей############
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn import datasets
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
```

Рисунок 4 – Необходимые библиотеки Python для создания программы

Воспользуемся матрицей ошибок (confusion matrix) для наглядного представления результата прогнозирования классификатора k-ближайших соседей [6]. Значения матрицы дают сводку правильных и неправильных прогнозов с разбивкой по каждой категории. Матрица показывает 0 + 2175 = 2175 правильных прогнозов и 257 + 5 = 262 неверных прогноза (рисунок 5, стр. 38).


Рисунок 5 – Оценка классификатора k-ближайших соседей с помощью матрицы ошибок

При классификации данных с помощью алгоритма k ближайших соседей точность модели составила 0,86 при k = 5. В ходе эксперимента были заданы 2, 3, 4, 5, 6, 7 соседей в модель KNN. При пяти и более соседях границы решения показали более гладкие границы, приняв оптимальное равновесие между переобучением и недообучением. Так как число голосов при реализации алгоритма KNN между 5 и 6 соседями одинаковые, предпочтительно выбрать соседей с наименьшим расстоянием до образца. Среднее время на обучение классификатора заняло 1115.65 мсек.

В логистической регрессии мы используем значение по умолчанию C = 1 (инверсионная сила регуляризации). Это обеспечивает хорошую производительность с точностью 0.89 как для обучения, так и для набора тестов. Результат, приведенный с помощью матрицы ошибок, показывает 1188 + 51 = 1249 правильных прогнозов и 38 + 0 = 38 неверных предсказаний (рисунок 6). Также результат приведем с помощью гистограммы вероятности (рисунок 7, стр. 39). Среди рассматриваемых классификаторов логистическая регрессия оказалась быстрее всех, показав результат 215.15 мсек.

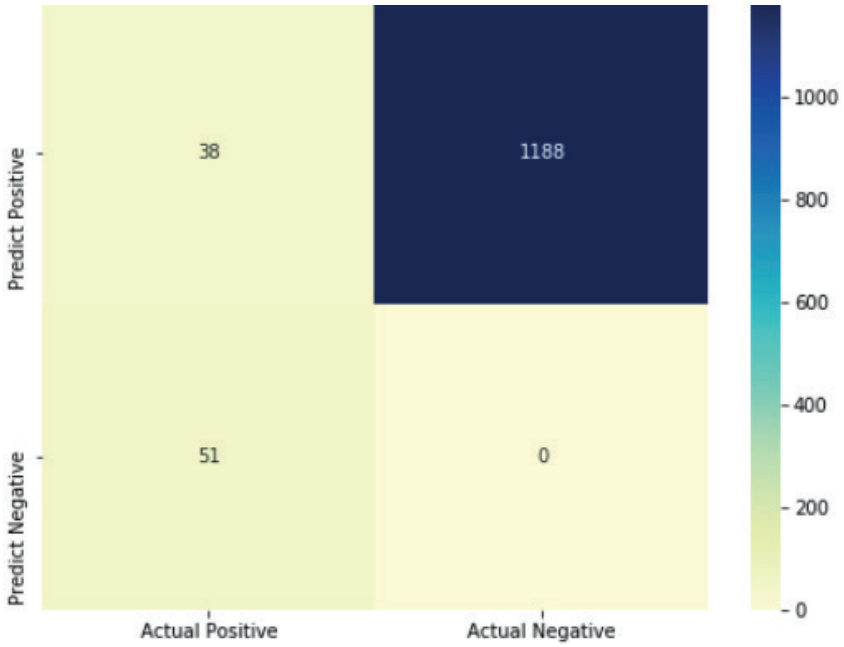


Рисунок 6 – Сопоставление предсказаний с фактическими данными с помощью алгоритма логистической регрессии

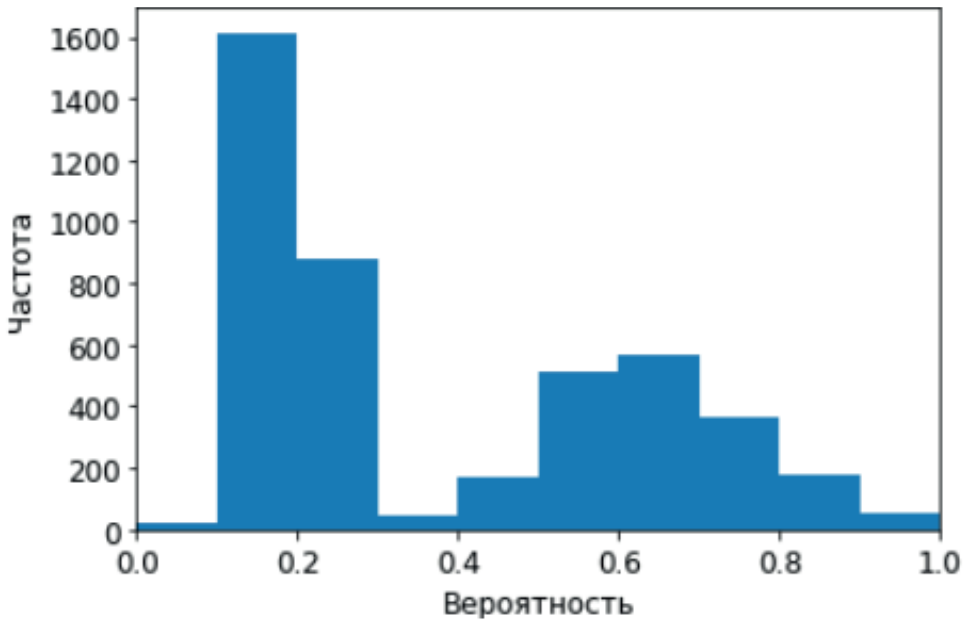


Рисунок 7 – Гистограмма вероятности определения цены методом логистической регрессии

Как видно на рисунке 7, гистограмма имеет положительный перекос. Второй столбец сообщает нам, что существует примерно 1600 наблюдений с вероятностью от 0, до 0,2. Есть небольшое количество наблюдений с вероятностью больше 0,5.

Результат прогнозирования с помощью метода дерева принятия решения с параметрами по умолчанию также показывает отличный результат с точностью 0,93. На обучение затрачено 412,07 мсек, уступая по скорости только модели на основе алгоритма логистическая регрессия. С помощью матрицы ошибок представлен результат прогнозирования (рисунок 8).


Рисунок 8 – Оценка классификатора дерево принятия решения с помощью матрицы ошибок

И последняя модель, рассматриваемая в данной статье, – случайный лес. Модель выявила больше закономерностей в данных, показав точность прогноза в 94% и затратив на обучение 541.03 мсек. Результат представлен в виде ROC-кривой (рисунок 9, стр. 40):

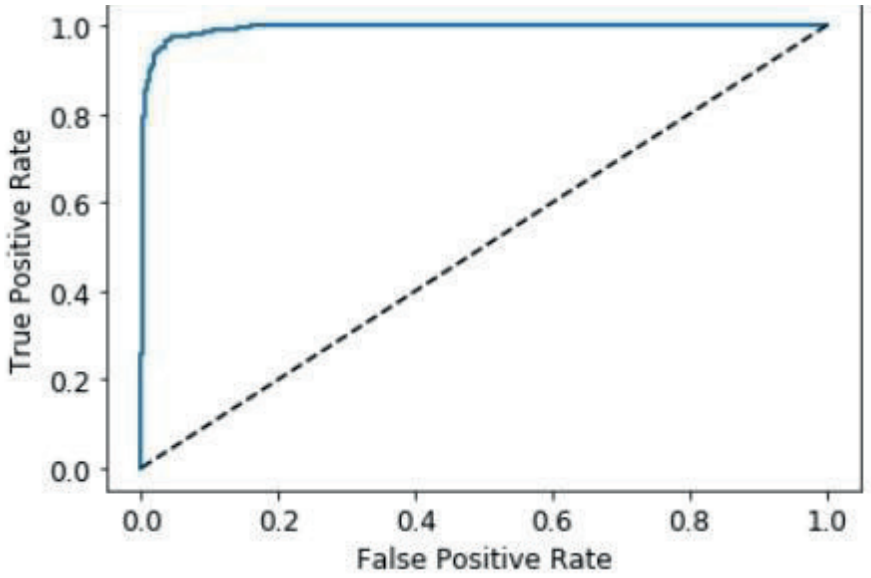


Рисунок 9 – ROC-кривая для случайного леса

Так как классификатор на основе алгоритма случайный лес показал высокие результаты по прогнозированию цены, принято решение использовать модель для выкупа вторичного авто.

Пользовательский интерфейс программного продукта показан на рисунке 10. Форма состоит из реестра объявлений, где зеленым выделяются объявления, где цена от продавца не является завышенной:

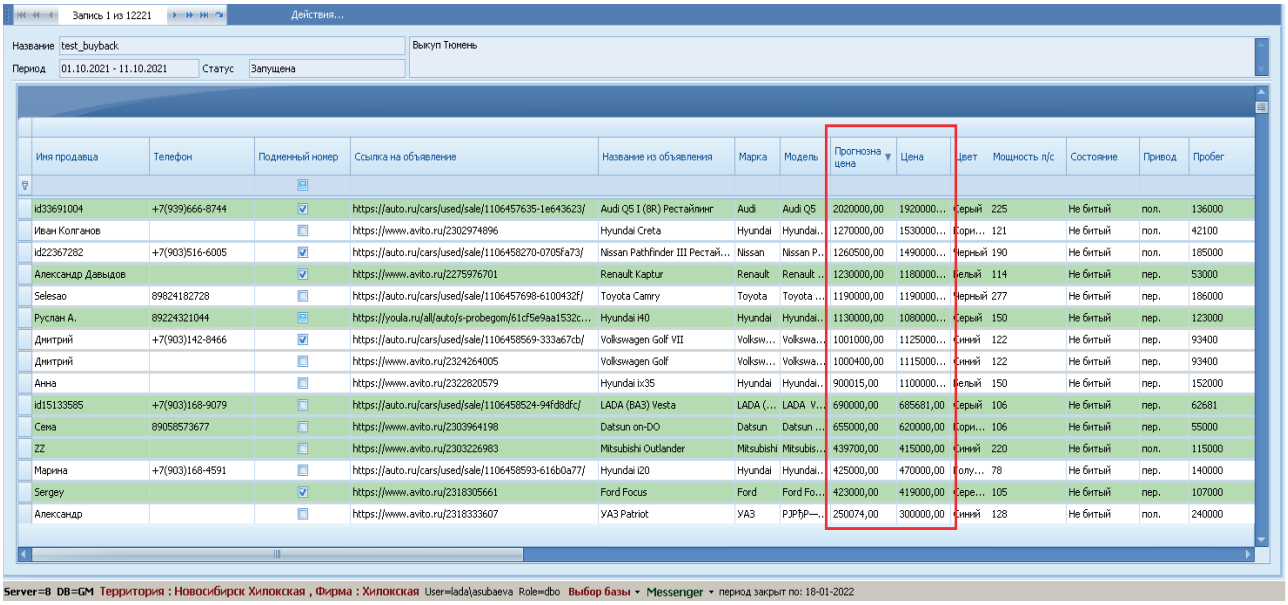


Рисунок 10 – Пользовательский интерфейс с ценой от продавца и предсказанной ценой с помощью классификатора на основе алгоритма случайный лес

**Заключение**

В этой статье рассказывается, как нынешние реалии дефицита электронных компонентов привели дилерские центры к перепродаже подержанных авто и как методы машинного обучения помогают выявить, не слишком ли завышена цена, и позволяют находить в этом сегменте оптимальное решение. В итоге мы получили систему, которая каждые 10 минут запрашивает с сервиса Haraba новые объявления, анализирует полученные данные и, находязакономерности, прогнозирует цену и будущий спрос на авто.

Благодаря введению машинного обучения в вопросах ценообразования компания оптимизирует операционную эффективность, использует алгоритмы для ценовых рекомендаций и прогноза продаж, позволяя менеджерам фокусироваться на стратегических задачах.

**ЛИТЕРАТУРА**

1 Narender Kumar, Dharmender K. Machine Learning based Heart Disease Diagnosis using Non-Invasive Methods, 2021, J. Phys.: Conf. Ser. 1950 012081.

2 Alarsan F.I., Younes M. Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. J Big Data 6, 81, 2019. URL: https://doi.org/10.1186/s40537-019-0244-x.

3 Рашка С.Python и машинное обучение. – Москва: ДМК Пресс, 2017. – 265 с.

4 Akhmed-Zaki D.Zh., Mukhambetzhanov S.T., Nurmakhanova Zh.M. and Abdiakhmetova Z.M. Using Wavelet Transform and Machine Learning to Predict Heart Fibrillation Disease on ECG 2021 // IEEE International Conference on Smart Information Systems and Technologies (Nur-Sultan, 28-30 April, 2021). URL: https://doi.org/ 10.1109/SIST50301.2021.9465990.

5 Bilbro R., Ojeda T., Bengfort B.,Language-Aware Data Products with Machine Learning, O'Reilly Media, 2018, 313 p.

6 Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. – Диалектика-Вильямс, Альфа-книга, 2018. – 465 с.

7 Плас Дж.В. Python для сложных задач. Наука о данных и машинное обучение. – Питер, 2018. – 265 с.

8 Бослав С. Статистика для всех. – Москва: ДМК Пресс, 2015. – 201 с.

9 Бенгфорд Б., Билбро Р. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. – Санкт-Петербург, 2019. –181 с.

10 Бринк Х., Ричардс Д., Феверолф М. Машинное обучение // Библиотека программиста. – Питер, 2019. – 304 с.

11 Chollet F. Deep learning with Python, Manning, 2020, 169 p.

12 Бурков А. Машинное обучение без лишних слов // Библиотека программиста. – Питер, 2020. – С. 60–69.

13 Shone N., Ngoc T.N., Phai V.D. and Shi Q. A deep learning approach tonetwork intrusion detection // IEEE Trans. Emerg. Topics Comput. Intell, vol. 2, no. 1, pp. 41–50.

14 Чистяков С.П. Случайные леса: обзор // Труды Карельского научного центра РАН.–2013.– No 1.– С.117–136.

15 Гришанов К.М., Белов Ю.С. Метод классификации K-NN и его применение в распознавании символов // Фундаментальные проблемы науки: Сборник статей Международной научно-практической конференции (15 мая 2016 г.) – Ч. 3. – Тюмень: НИЦ Аэтерна, 2016. – С. 30–33.

16 Jenhani I., Amor N. B., Eloued Z. Decision trees as possibilistic classifiers // International Journal of Approximate Reasoning, no. 48 (nov.2008), pp. 786–801. URL: https://doi.org/10.1016/j.ijar.2007.12.002.

17 Rymarczyk T., Kozłowski E. Logistic Regression for Machine Learning in Process Tomography // MDPI, no. 19(15) (2019), pp. 206–208. URL: https://doi.org/10.3390/s19153400.

18 URL: https://haraba.ru.

19 Sanjay P. Pro RESTful APIs Design, Build and Integrate with REST, JSON, XML and JAX-RSApress, Berkeley, CA, 2018. URL: https://doi.org/10.1007/978-1-4842-2665-0.

20 Stephen R.G. Fraser Windows Services. In: Pro Visual C++/CLI and the .NET 2.0 Platform, Apress, 2006. URL: https://doi.org/10.1007/978-1-4302-0109-0_14.

**REFERENCES**

1 Narender Kumar, Dharmender K. Machine Learning based Heart Disease Diagnosis using Non-Invasive Methods, 2021, J. Phys.: Conf. Ser. 1950 012081.

2 Alarsan F.I., Younes M. Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. J Big Data 6, 81, 2019. URL: https://doi.org/10.1186/s40537-019-0244-x.

3 Rashka S. (2017) Python i mashinnoe obuchenie. Moskva: DMK Press. 265 p.

4 Akhmed-Zaki D.Zh., Mukhambetzhanov S.T., Nurmakhanova Zh.M. and Abdiakhmetova Z.M. Using Wavelet Transform and Machine Learning to Predict Heart Fibrillation Disease on ECG 2021 // IEEE International Conference on Smart Information Systems and Technologies (Nur-Sultan, 28-30 April, 2021). URL: https://doi.org/ 10.1109/SIST50301.2021.9465990.

5 Bilbro R., Ojeda T., Bengfort B.,Language-Aware Data Products with Machine Learning, O'Reilly Media, 2018, 313 p.

6 Zheron O. (2018) Prikladnoe mashinnoe obuchenie s pomoshh'ju Scikit-Learn i TensorFlow. Dialektika-Vil'jams, Al'fa-kniga. 465 p.

7 Plas Dzh.V. (2018) Python dlja slozhnyh zadach. Nauka o dannyh i mashinnoe obuchenie. Piter. 265 p.

8 Boslav S. (2015) Statistika dlja vseh. Moskva: DMK Press. 201 p.

9 Bengford B., Bilbro R. (2019) Prikladnoj analiz tekstovyh dannyh na Python. Mashinnoe obuchenie i sozdanie prilozhenij obrabotki estestvennogo jazyka. Sankt-Peterburg. 181 p.

10 Brink H., Richards D., Feverolf M. (2019) Mashinnoe obuchenie. Biblioteka programmista. Piter. 304 p.

11 Chollet F. Deep learning with Python, Manning, 2020, 169 p.

12 Burkov A. (2020) Mashinnoe obuchenie bez lishnih slov. Biblioteka programmista. Piter. PP. 60–69.

13 Shone N., Ngoc T.N., Phai V.D. and Shi Q. A deep learning approach tonetwork intrusion detection // IEEE Trans. Emerg. Topics Comput. Intell, vol. 2, no. 1, pp. 41–50.

14 Chistjakov C.P. (2013) Sluchajnye lesa: obzor. Trudy Karel'skogo nauchnogo centra RAN. No 1. PP.117–136.

15 Grishanov K.M., Belov Ju.S. (2016) Metod klassifikacii K-NN i ego primenenie v raspoznavanii simvolov. Fundamental'nye problemy nauki: Sbornik statej Mezhdunarodnoj nauchno-prakticheskoj konferencii (15 maja 2016 g.). Ch. 3.Tjumen': NIC Ajeterna. – PP. 30–33.

16 Jenhani I., Amor N. B., Eloued Z. Decision trees as possibilistic classifiers // International Journal of Approximate Reasoning, no. 48 (nov.2008), pp. 786–801. URL: https://doi.org/10.1016/j.ijar.2007.12.002.

17 Rymarczyk T., Kozłowski E. Logistic Regression for Machine Learning in Process Tomography // MDPI, no. 19(15) (2019), pp. 206–208. URL: https://doi.org/10.3390/s19153400.

18 URL: https://haraba.ru.

19 Sanjay P. Pro RESTful APIs Design, Build and Integrate with REST, JSON, XML and JAX-RSApress, Berkeley, CA, 2018. URL: https://doi.org/10.1007/978-1-4842-2665-0.

20 Stephen R.G. Fraser Windows Services. In: Pro Visual C++/CLI and the .NET 2.0 Platform, Apress, 2006. URL: https://doi.org/10.1007/978-1-4302-0109-0 14.

**Сведения об авторах**

**1. Асубаева Еркежан Маратовна** (автор для корреспонденции)

Магистрант Казахского национального университета им. аль-Фараби, пр. Аль-Фараби, 71/27, 050040, г. Алматы, Казахстан;

ORCID ID: 0000-0001-7229-267X;

E-mail: erkezhanasubaeva@gmail.com.

**2. Абдиахметова Зухра Муратовна**

PhD, и.о. доцента Казахского национального университета им. аль-Фараби, факультет информационных технологий, пр. Аль-Фараби, 71/27, 050040, г. Алматы, Казахстан;

ORCID ID: 0142-5747-4;

E-mail: zukhra.abdiakhmetova@gmail.com.

**Авторлар туралы мәліметтер**

**1. Асубаева Еркежан Маратовна** (корреспонденция авторы)

Магистрант, әл-Фараби атындағы Қазақ ұлттық университеті, ақпараттық технологиялар факультеті, әл-Фараби даңғылы, 71/27, 050040, Алматы қ., Қазақстан;

ORCID ID:0000-0001-7229-267X;

E-mail: erkezhanasubaeva@gmail.com.

**2. Абдиахметова Зухра Муратовна**

PhD, доцент м.а., әл-Фараби атындағы Қазақ ұлттық университеті, Ақпараттық технологиялар факультеті, әл-Фараби даңғылы, 71/27, 050040, Алматы қ., Қазақстан*;*

ORCID ID: 0142-5747-4;

E-mail: zukhra.abdiakhmetova@gmail.com.

**Information about authors**

**1. Assubayeva Yerkezhan Maratovna** (corresponding author)

Master's student in Computer Engineering, Al-Farabi Kazakh National university, Faculty of Information Technologies, Al-Farabi Ave., 71/27, 050040, Almaty, Kazakhstan;

ORCID ID: 0000-0001-7229-267X;

E-mail: erkezhanasubaeva@gmail.com.

**2. Abdiakhmetova Zukhra Muratovna**

PhD, a.a. professor, Al-Farabi Kazakh National University, Faculty of Information Technologies, Al-Farabi Ave., 71/27, 050040, Almaty, Kazakhstan;

ORCID ID: 0142-5747-4;

E-mail: zukhra.abdiakhmetova@gmail.com.