Список источников

1. Викулов Е.О., Леонов Е.А., Денисова Л.А. Автоматизированное распределение больших объемов данных высоконагруженных систем Динамика систем, механизмов и машин. 2014. № 3. С. 146-149.
2. Викулов Е.О., Распределение больших объемов данных // Известия Тульского государственного университета, технические науки. Тула, 2023. Вып. 12, С. 457-461.
3. Server load balancing | Akamai [Электронный ресурс] URL: https://www.akamai.com/uk/en/resources/server-load-balancing.jsp (дата обращения: 23.09.2018).
4. Basic Load Balancing – IBM cloud [Электронный ресурс] URL: https://console.bluemix.net/docs/infrastructure/loadbalancer-service/basic-load-balancing.html (дата обращения: 23.09.2018).
5. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия –Телеком, 2007. 288 с.
6. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. М.: Мир, 1992.
7. Каллан Р. Основные концепции нейронных сетей – The Essence of Neural Networks First Edition. М.: Вильямс, 2001. 288 с.
8. Vikulov E.O., Denisov O.V., Denisova L.A. Data distribution system preparation of server stations data // Journal of Physics: Conference Series Сер. "Mechanical Science and Technology Update, MSTU 2018" 2018. С. 012097.
9. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия – Телеком, 2001. С. 382.
10. MATLAB Makers of Matlab and Simulink. [Электронный ресурс] URL: http://www.mathworks.com (дата обращения:12.12.2023).

Викулов Егор Олегович, старший преподаватель, vikuloveo@gmail.com, Россия, Омск, Омский государственный технический университет,

Денисова Людмила Альбертовна, д-р техн. наук, профессор, denisova@asoiu.com, Россия, Омск, Омский государственный технический университет

RESEARCH OF HIGH-LOAD WEB APPLICATIONS DATA DISTRIBUTION USING INTELLECTUAL TECHNOLOGIES.

E.O. Vikulov, L.A. Denisova

The article discusses the issues of data distribution in a server complex, taking into account the state of the servers based on fuzzy logical inference. A simulation model has been developed to determine the number of requests served by each server, as well as the time and cost of data processing. The studies have confirmed that the use of the proposed approach allows you to select a server based on data about the state of the server complex and the user, which reduces the amount of stored and transmitted data, without reducing the speed when loading and downloading data.

Key words: data distribution, highly loaded web applications, balancer server, pattern clustering, data analysis.

Vikulov Egor Olegovich, senior lecturer, vikuloveo@gmail.com Russia, Omsk, Omsk State Technical University,

Denisova Lyudmila Albertovna, doctor of technical, professor, denisova@asoiu.com, Russia, Omsk, Omsk State Technical University

УДК 004.67; 004.912
DOI: 10.24412/2071-6168-2024-3-8-9

МЕТОДЫ ПРЕОБРАЗОВАНИЯ ДАННЫХ ДЛЯ АНАЛИТИКИ

Е.В. Нурматова

Результаты работы алгоритмов машинного обучения сильно зависят от качества подготовки данных, определяемого спецификой решаемой задачи. В данной работе рассматривается поэтапное описание решения проблем качества данных, предназначенных для аналитики. Каждый этап процесса преобразования данных, включающий обработку дубликатов, противоречий, аномальных и отсутствующих значений, сглаживание выбросов, поддерживается соответствующим программным решением с применением специализированных библиотек Python. Для более полного понимания зависимостей между признаками, анализа выбросов, распределения частоты категориальных признаков показаны примеры возможных визуализаций. В итоге получается создать более устойчивые, интерпретируемые данные, ценность которых определяется не столько их объёмами, сколько качеством.

Ключевые слова: предподготовка данных, исследовательский анализ данных, очистка, проблемы качества данных, машинное обучение.

Обеспечение качества данных позволяет уменьшить потенциальные ошибки, обеспечить качество и объективность результатов и управленческих решений. Этими вопросами занимались с момента развития компьютерных наук. Задача систематизации и стандартизации подходов к структурированию и качеству данных постоянно требует решения по мере роста объёмов данных, их разнородности.

Цикл обработки данных CRISP-DM (*cross-industry data mining process)* [2] начинается с формулирования цели аналитики, сбора и подготовки данных для исследования целевых показателей. При наличии нескольких источников важно продумать как будут интегрироваться данные.

Отчёт о первоначальном сборе данных включает описание формата и объёма выборки показателей, при подготовке которого, используя методы статистического анализа, распределяются ключевые признаки и отношения между ними.

Отчёт о качестве данных содержит описание полноты охвата всех необходимых признаков, правильности их выбора, наличия «выбросов», отсутствующих значений в данных. Другими словами, как данные представлены, где встречаются и насколько распространены [1,3].

Далее выполняется разделение на обучающую и тестовую выборки, используемые для создания обученной модели и её проверки. Обученная модель применяется к тестовым данным для оценки, насколько точно они предсказывают соответствующие метки классов. В дополнение, каждая из моделей проходит n-кратную перекрёстную проверку (обучение и тестирование с рандомизированным разделением данных).

На этапе оценки модели результаты генерируются с помощью нескольких различных методов, в зависимости от которых могут быть пересмотрены настройки параметров модели для следующей итерации, пока не будет получена наилучшая модель [6,7].

Объединение этапов исследовательского анализа данных (EDA, *exploratory data analysis*), моделирования, разработки модели и её оценки представлено на рис.1.

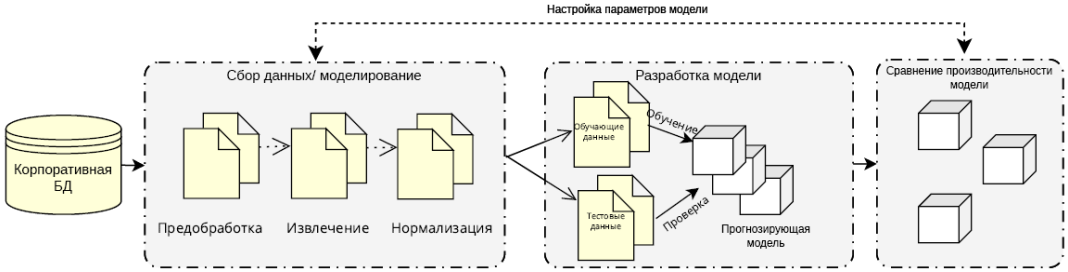

Рис.1. Процесс обработки данных и разработки модели

Этап предподготовки данных включает все операции по подготовке результирующего набора данных из исходных, возможно требующих объединения, данных для анализа теми инструментами, которые были выбраны для решения поставленной задачи [9, 11]. В него входят операции очистки и контроля, изменения структуры данных, агрегирования до необходимого уровня представления, объединение данных из различных источников и таблиц.

Очистка данных (скраббинг) связана с обнаружением и устранением ошибок и несоответствий в данных с целью повышения их качества [6]. Для выявления и устранения проблем с качеством данных, а также возможных их решений, используется контрольный список, представленный в таблице.

Для понимания данных, выявления возможных проблем с качеством данных применяем специализированные библиотеки *Python*, выполняя этапы предподготовки данных [10].

Создаётся описательная статистика, которая суммирует медиану (*mean*), дисперсию (*std*) и форму распределения набора данных, исключая отсутствующие значения. Результат показан на рисунке 1.

```
df.columns.sort_values()
nd = df.describe().T
ndt = num_describe.loc[:,['mean', 'std','25%', '50%', '75%']]
print(ndt)
odt = df.describe(include=object)
print('\n',odt)
```

```
            mean        std    25%   50%    75%
BMI    28.313822   5.458155  24.37  27.6  31.75
AC      2.506803   3.781993   0.00   0.0   4.00
FC     18.343433  10.935008   8.00  16.0  30.00
GVC    11.865191   9.641609   4.00   8.0  16.00

                GH      HD Depression Diabetes Arthritis Age_Category      SH
count       199608  199608     199608   199608    199608       199608  199608
unique           5       2          2        4         2           13       2
top      Very Good      No         No       No        No        65-69      No
freq         69999  182597     159406   166142    135285        20897  118026
```

Рис.1. Результаты описательной статистики выбранных признаков

Выполняется проверка ограничений данных по типу, диапазону и уникальности. При помощи функций библиотеки *pandas*, возвращаем подмножество столбцов датафрейма *df* с указанием типов столбцов. Выявляем признаки, имеющий текстовый формат данных, а не числовой, и наоборот, вычислив общую их численность:

```
num = df.select_dtypes(include=['float64']).columns.sort_values()
categ = df.select_dtypes(include=['object']).columns.sort_values()
print(f'В наборе {len(categ)} категориальных переменных')
print(f'В наборе {len(num)} числовых переменных')
```

И, при необходимости, изменяем формат данных:

```
s.BMI = s.BMI.str.replace(',', '.').str.replace(' ', '')
pd.to_numeric(s.BMI)   #преобразование типа object  в float64
```

Аналогично проверяем временные данные:

```
# Преобразование строковой даты в объект datetime с разделением на год, месяц и день
s['Timestamp'] = pd.to_datetime(s['Timestamp'])
s['year'] = s['Timestamp'].dt.year
s['month'] = s['Timestamp'].dt.month
s['day'] = s['Timestamp'].dt.day
s = s.drop("Timestamp", axis=1)
```

Категории проблем с качеством данных и их решений

Параметры проверки	Потенциальные решения
Проблемы ограничений данных	
1 Ограничения типа данных При импорте данных, либо при их формировании убедимся, что столбцы (признаки) имеют правильный тип данных	• Преобразовать в правильный тип данных
2 Ограничения диапазона данных Обеспечиваем, чтобы различные столбцы имели правильный диапазон. Особенно для столбцов, имеющих ограничения.	• Проверить, нет ли опечаток, например, десятичной точки в неправильном месте или неверном её обозначении (запятая вместо точки и наоборот) • Удалить строки, в которых точки данных нарушают ограничения диапазона, установить точку данных, которая нарушает ограничения диапазона, на максимум, или минимум
3 Ограничения уникальности Обеспечиваем отсутствие дубликатов в строк набора данных.	• Сохранить только одну из дублирующихся строк • Объединить строки, в которых нет дубликаты
Проблемы с текстовыми и категориальными данными	
4 Ограничения для категориальных данных Обеспечиваем, чтобы категориальные столбцы имели правильные и последовательные категории	• Удалить строки, на которые влияют несовместимые категории • Переименовать несоответствующие категории в правильные значения • Сделать выводы о категориях на основе других • Ввести категории на основе других точек данных, если неясно, как они должны быть изменены
5 Нарушение длины текстовых данных Обеспечиваем, чтобы текстовые колонки, соответствующие определенному признаку, имели одинаковую длину строки	• Удалить строки, на которые повлияло нарушение длины • Установить для затронутых наблюдений значение NaN
6 Единое форматирование текстовых данных Обеспечиваем, чтобы текстовые столбцы, соответствующие определенному признаку, имели одинаковое форматирование строк	• Стандартизировать форматирование затронутых наблюдений • Удалить строки, затронутые несоответствием
Проблемы однородности данных	
7 Однородность единиц измерения для числовых столбцов, для столбцов типа date Обеспечиваем, чтобы числовые столбцы имели одинаковые единицы измерения/ формат (особенно актуально при объединении наборов данных из разных источников).	• Удалить строки, которые выпадают из контекста и не проходят проверку на правильность представления • Стандартизировать единицы измерения/ формат datetime-данных, где возможно
8 Валидация для числовых столбцов, для столбцов типа date Используем несколько признаков в наборе данных для обеспечения достоверности и целостности другого признака, в т.ч. datetime-признаки проходят проверку на правильность	• Удалить строки, в которых проверка на правильность не работает • Применить правила предметной области на основе знания данных
Проблемы с отсутствующими данными	
9 Полностью отсутствующие случайные данные (не наблюдается никакой связи между отсутствующими значениями и другими значениями в наборе данных) **Пропущенные неслучайные данные** (когда существует систематическая связь между отсутствующими данными и другими ненаблюдаемыми величинами)	• Удалить отсутствующие строки • Заменить недостающие строки с помощью статистических показателей для измерения центра данных, например, медианы или среднее • Заменить пропущенные строки с помощью алгоритмов машинного обучения • Собрать новые данные и признаки

Проверяем возможное наличие аномальных значений, выходящих за пределы допустимого диапазона. Например, ставим фильтр по возрасту:

```
age_min = s['Age']<0
age_max = s['Age']>100
anomaly_age= age_min|age_max
s[anomaly_age].head()
```

При их наличии можно либо удалить выявленные точки данных, либо заменить значения в наборе:

```
s['Age'] = sd['Age']. replace([99999999, -29, -1726,-1], [33, 32, 29, 43])
```

И, завершая данный этап очистки, обеспечиваем отсутствие дубликатов в наборе данных:

```
dupl_rows= s[s.duplicated()]
print("№№ повторяющихся строк: ", dupl_rows.shape)
s = s.drop_duplicates()
```

В процессе преобразования данных используются методы нормализации, стандартизации данных для масштабирования значений признаков в диапазоне от 0 до 1. Это важно для переменных с разными единицами измерения и масштабами:

```
# Нормализация обучающих данных
from sklearn import preprocessing
```

```
min_mascaler =  preprocessing.MinMaxScaler()
scaled_minmax = min_mascaler.fit_transform(X_train)
scaled_minmadf = pd.DataFrame(scaled_minmax, columns = feature_names)
```

Более симметричным распределение данных делает стандартизация, преобразуя значения так, чтобы их среднее значение (mean) стало нулевым, а стандартное отклонение (std) единичным:

```
# Стандартизация обучающих данных
scaler =  preprocessing.StandartScaler()
st_data = scaler.fit_transform(X_train)
scaled_df = pd.DataFrame(st_data, columns = feature_names)
```

Используя в дальнейшем инструменты EDA, продолжается исследование данных, обобщение их основных характеристик, часто визуализируя полученные результаты, согласно следующим этапам:

— импорт необходимых библиотек для EDA. Как правило, это библиотеки pandas, numpy для работы с табличными данными, временными рядами, библиотеки для создания графиков и визуализации данных seaborn, matplotlib и прочие;

— загрузка данных в датафрейм, в зависимости от форматов исходных данных (csv, xlsx, json);

— проверка типов данных (checking the types of data), например, с использованием методов dtypes:

```
df.dtypes
```

— удаление нерелевантных столбцов (dropping irrelevant columns):

```
df = cvd.drop(columns=["Skin_Cancer","Other_Cancer", "Height_(cm)", "Weight_(kg)", "Checkup"])
```

— переименование столбцов (renaming the columns) , например:

```
df = df.rename(columns={"General_Health":"GH", "Heart_Disease": "HD"})
```

— удаление повторяющихся строк (dropping the duplicate rows):

```
dupl_rows= df[df.duplicated()]
df = df.drop_duplicates()
```

— удаление или замена отсутствующих или нулевых значений (dropping the missing or null values), для начала проверив их наличие:

```
print(df.isnull().sum())
df["Sleep Disorder"].fillna("No",inplace=True)
df["comments"].fillna("No comments",inplace=True)
```

— преобразования категориальных данных, а именно порядковое кодирование возможно реализовать при помощи классов LabelEncoder, OrdinalEncode, OneHotEncoder модуля sklearn.preprocessing

```
#enc = OrdinalEncoder()
le = LabelEncoder()
#ob=['Gender', 'NObeyesdad', 'CALC', 'MTRANS', 'CAEC','family_history_with_overweight, 'FAVC', 'SMOKE','SCC']
ob=df.select_dtypes(include=['object'])
for colsn in ob:
   df[colsn] = le.fit_transform(df[colsn].astype(str))
```

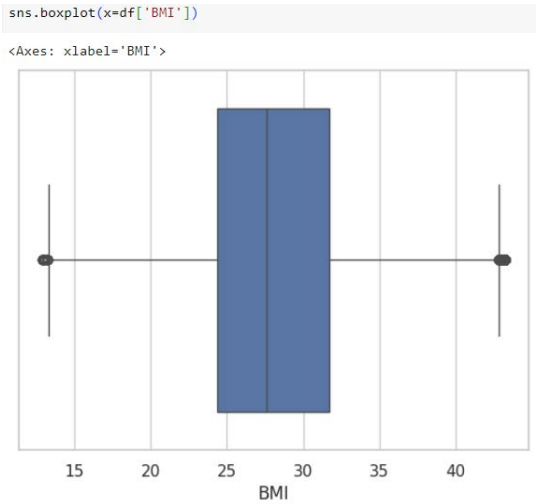

Рис.2. Результат визуализации выбросов для выбранного признака

— обнаружение выбросов (detecting outliers), например, при помощи межквартильного диапазона IRQ, представляющего разность между 25-ым процентилем Q1 и 75-ым процентилем Q3 в наборе данных. IRQ измеряет разброс средних 50%:

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print("Значения IQR для выбранных критериев\n",IQR)
```

Выбросом объявляется точка данных со значением в 1,5 раза выше/ ниже IRQ, отфильтровываем оставшиеся данные:

```
df = df[~((df < (Q1 - 1.5 * IQR))|(df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

Например, для признака BMI (индекс массы тела) визуализация выброса позволяет оценить симметрию данных, выявить аномалии данных с помощью boxplot («ящика с усами»), как показано на рисунке 2:

— построение графика зависимости различных характеристик друг от друга (разброс) и частоты (гистограмма), например, при помощи тепловой карты (heatmap). На рисунке 3 приведена таблица и визуализация, показывающей степень взаимосвязей между переменными:

```
plt.figure(figsize=(10,5))
c= df.corr()
sns.heatmap(c,cmap="YlGnBu",annot=True)
plt.title('Тепловая карта признаков набора данных')
```

Это позволяет выявить зависимости в наборе данных:

	BMI	AC	FC	GVC
BMI	1.000000	-0.056931	-0.055799	-0.042559
AC	-0.056931	1.000000	0.022916	0.070886
FC	-0.055799	0.022916	1.000000	0.260409
GVC	-0.042559	0.070886	0.260409	1.000000

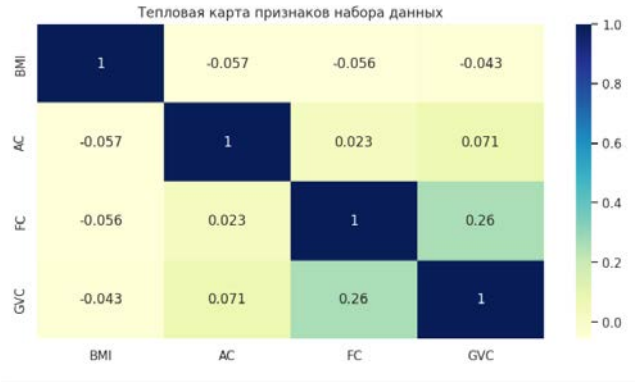

Рис.3. heatmap выбранных признаков

Для наиболее связанных признаков можно построить диаграмму рассеяния и корреляции для лучшего понимания структуры и зависимости данных, для принятия более обоснованных аналитических решений.

В работе были классифицированы проблемы качества данных, которые решаются с помощью очистки данных, занимающей большую часть рабочего процесса аналитики, представлен обзор основных подходов к решению.

Описанные выше варианты решения проблем не единственные. Есть достаточно много других методов обработки, способных помочь повысить качество данных, начиная от экспертных систем и заканчивая нейросетями [9, 12]. При этом нужно учитывать то, что методы очистки и преобразования данных могут быть сильно привязаны к предметной области.

Список литературы

1. What is the CRISP-DM metodology? // [Электронный ресурс] URL: https://www.sv-europe.com/crisp-dm-methodology (дата обращения: 27.02.2023).

2. Груздев А.В. Предварительная подготовка данных в Python: Том 1. Инструменты и валидация. М.: ДМК Пресс, 2023. 816 с.

3. Груздев А.В. Предварительная подготовка данных в Python: Том 2. План, примеры и метрики качества. М.: ДМК Пресс, 2023. 814 с.

4. Копырин А.С., Макарова И.Л. Алгоритм препроцессинга и унификации временных рядов на основе машинного обучения для структурирования данных // Программные системы и вычислительные методы. 2020. № 3.

5. Кэти Танимура SQL для анализа данных. Расширенные методы преобразования данных для аналитики. СПб: bhv-СПб, 2024. 384 с.

6. Макаров А.В., Намиот Д.Е. Обзор методов очистки данных для машинного обучения. International Journal of Open Information Technologies, 2023. Vol. 11. No. 10. P.70-78.

7. Rahm Erhard, Do Hong. Data Cleaning: Problems and Current Approaches. IEEE Data Eng. Bull.23. 2000. P. 3-13.

8. Sun W, Cai Z, Li Y, Liu F, Fang S, Wang G. Data Processing and Text Mining Technologies on Electronic Medical Records: A Review. J Healthc Eng. 2018.

9. Hernandez M.A., Stolfo S.J. Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge Discovery, 1998. 2 (1). P. 9-37.

10. Quass D.A Framework for Research in Data Cleaning. Unpublished Manuscript. Brigham Young Univ., 1999.

11. M. Bharathi, D. Abhiram, & I.V. Dwaraka Srihith From Raw to Refined: Python's Touch on Data Cleaning. Advanced Innovations in Computer Programming Languages, 2024. 6(1). P. 27–32.

12. Purbasari Ayi, Rinawan Fedri, Zulianto Arief Susanti, Ari Komara, Hendra. CRISP-DM for Data Quality Improvement to Support Machine Learning of Stunting Prediction in Infants and Toddlers, 1-6. DOI: 10.1109/ICAICTA53211.2021 9640294.

Нурматова Елена Вячеславовна. канд. техн. наук, доцент, nurmatova@mirea.ru, Россия, Москва, Российский технологический университет - РТУ МИРЭА

DATA TRANSFORMATION METHODS FOR ANALYTICS

E.V. Nurmatova

The results of machine learning algorithms strongly depend on the quality of data preparation, which is determined by the specifics of the problem to be solved. In this paper, we consider a step-by-step description of solving data quality problems for analytics. Each stage of the data transformation process, including processing of duplicates, inconsistencies, anomalous and missing values, and outlier smoothing, is supported by an appropriate software solution using specialized Python libraries. Examples of possible visualizations are shown for a better understanding of dependencies between features, outlier analysis, frequency distribution of categorical features. As a result, it is possible to create more stable, interpretable data, the value of which is determined not so much by its volume as by its quality.

Key words: data preconditioning, exploratory data analysis, cleaning, data quality issues, machine learning.

Nurmatova Elena Vyacheslavovna, candidate of technical sciences, docent, nurmatova@mirea.ru, Russia, Moscow, Russian Technological University – RTU MIREA

УДК 004.8
DOI: 10.24412/2071-6168-2024-3-13-14

ИНТЕЛЛЕКТУАЛЬНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ПОДБОРА БУДУЩЕЙ ПРОФЕССИИ ДЛЯ АБИТУРИЕНТОВ

Ю.А. Мустахитдинова, Р.С. Зарипова

Современный уровень развития цифровых технологий позволяет эффективно использовать их в области образования, в том числе для ориентира в выборе будущей профессии. Одним из таких инструментов является рекомендательная система, которая поможет абитуриентам принять более осознанное решение по выбору своего профессионального будущего. В статье рассмотрен вопрос реализации и применения интеллектуальной рекомендательной системы по подбору профессий, которая предлагает пользователям персонализированные рекомендации в выборе подходящей профессии на основе их интересов, целей и предпочтений, а также дополнительную информацию, которая включает требуемые навыки, перспективы карьерного роста и список университетов, в которых они смогут получить ту или иную профессию. Предлагается внедрение данной системы в школах республики Татарстан для того, чтобы школьники поступали в ВУЗы именно в своем регионе. В статье обозначена роль рекомендательной системы в процессе обучения, а также анализируются её преимущества и важность для будущей карьеры обучающихся.

Ключевые слова: образование, профессия, рекомендательная система, подбор профессий, абитуриент, карьера, нейронные сети, обучение.

Введение. В условиях цифровизации всех сфер жизни возникает большая необходимость использования новых технологий в сфере образования [1, 2]. В настоящее время на рынке труда имеется большое количество профессий. Абитуриентам сложно сориентироваться в огромном разнообразии направлений профессиональной деятельности, вследствие чего возникает затруднение при выборе подходящей профессии. Кроме того, в последнее время наблюдается проблема миграции абитуриентов республики Татарстан в другие регионы, которая может рассматриваться как одна из форм «утечки мозгов». Школьники Татарстана видят больше перспектив в таких городах, как Москва и Санкт-Петербург, вследствие этого покидают свой регион, чтобы продолжить образование в другом месте и часто не возвращаются. Миграция абитуриентов приводит к потере высококвалифицированных специалистов, что затрудняет экономический и социальный рост субъекта.

Для решения данных проблем было решено разработать интеллектуальную рекомендательную систему по подбору профессий, которая будет сотрудничать с ведущими ВУЗами Татарстана и продвигать их с помощью различных интерактивов (например, ролики от партнеров с примером работ).

Объектом исследования являются абитуриенты, которым нужна помощь при выборе будущей профессии.

Новизна исследования заключается в разработке и использовании единой системы для абитуриентов в учреждениях общего образования на уровне субъекта России.

В рамках данного исследования была разработана и протестирована система, которая собирает и анализирует информацию о предпочтениях школьников, а также использует машинное обучение для более точного подбора направления деятельности.

Методы исследования. Для исследований применялись статистические и логические методы, аналитические методы, методы сравнительного анализа, общенаучные и социологические методы: анализ научной литературы и источников по теме исследования, анкетный опрос абитуриентов и студентов-выпускников.

Результаты исследования. Разработанная интеллектуальная рекомендательная система по подбору профессий поможет школьникам принять более осознанное решение относительно своей будущей карьеры, так как в последнее время абитуриентам тяжело сориентироваться в огромном количестве информации и принять более информированное решение с учетом своих интересов и предпочтений.