DOI: 10.18137/RNU.V9187.23.02.P.141 УДК 004.02:004.89:338.1

А.В. Маликов, К.Ю. Медведев, Т.-А.А. Хаитов, С.Е. Вечерская

ПРИМЕНЕНИЕ МЕТОДА СЛУЧАЙНОГО ЛЕСА ДЛЯ АНАЛИЗА ПОКУПАТЕЛЬНОЙ СПОСОБНОСТИ

Аннотация. Основной задачей работы является применение метода машинного обучения в реальной бизнес-ситуации. Выбран такой метод машинного обучения, как случайный лес. Описаны преимущества данного алгоритма, а также подробно изложен процесс разработки – с нуля до корректно функционирующей модели, способной автоматически классифицировать новых клиентов магазина по их платежеспособности и покупательной способности. Итоговая точность модели составила ≈ 81 %. Внедрение предложенного инструмента позволит автоматизировать бизнес-процесс анализа покупателей по их платежеспособности и повысить эффективность использования временных и человеческих ресурсов.

Ключевые слова: машинное обучение, метод случайного леса, алгоритмы машинного обучения, задачи классификации, анализ данных, анализ платежеспособности покупателей.

A.V. Malikov, K.Yu. Medvedev, T.-A.A. Khaitov, S.E. Vecherskaya

APPLICATION OF THE RANDOM FOREST METHOD FOR PURCHASING POWER ANALYSIS

Abstract. The article considers the application of the random forest machine learning method in a real business situation. The aim of the work set the task: having a database of clients collected when registering them in a mega market, quantitatively estimated store marketers for each customer manually, to develop a system based on machine learning, which is able to automatically classify customers by their solvency. Achieving this aim assumed the solution of the following tasks: 1) on the basis of the collected customer database and their creditworthiness groups, to analyze and determine which criteria specified by clients at the time of registration most affect the perceived solvency; 2) normalize collected customer data to fit the selected machine learning method; 3) train the model and test its work on clients, already with the specified solvency groups; 4) improve the system's accuracy.

The article reveals the advantages of this algorithm and describes the entire development process: from zero to a correctly functioning model that can automatically classify new store customers according to their solvency. The final accuracy of the model was approx. 81 %. Implementation of the proposed model enables to automate the business process of buyers' solvency analysis and increase the time and human resources efficiency.

Keywords: machine learning, machine learning algorithms, random forest method, classification tasks, data analysis.

Введение

Обычной практикой для большинства крупных сетей и магазинов является сбор данных о покупательском поведении клиентов с целью улучшения качества товаров и предоставляемых услуг. С ростом количества клиентов возникает потребность в их сегментации и распределении поведенческих характеристик по выбранным паттернам для последующего проведения более структурированного анализа. Одна из наиболее часто оцениваемых групп показателей – покупательная способность клиента. Однако как же следует

© Маликов А.В., Медведев К.Ю., Хаитов Т.-А.А., Вечерская С.Е., 2023

Маликов Алексей Валерьевич
магистрант кафедры информационных систем в экономике и управлении, Институт информационных технологий и инженерно-компьютерных технологий, Российский новый университет, Москва. Сфера научных интересов: искусственный интеллект, применение информатики в управлении.
Электронный адрес: thrashor29@gmail.com

Медведев Кирилл Юрьевич
магистрант кафедры информационных систем в экономике и управлении, Институт информационных технологий и инженерно-компьютерных технологий, Российский новый университет, Москва. Сфера научных интересов: искусственный интеллект, применение информатики в управлении.
Электронный адрес: MedvedevKYu@stud.rosnou.ru

Хаитов Тимур-Антоний Алишерович
магистрант кафедры информационных систем в экономике и управлении, Институт информационных технологий и инженерно-компьютерных технологий, Российский новый университет, Москва. Сфера научных интересов: искусственный интеллект, применение информатики в управлении.
Электронный адрес: HaitovT@stud.rosnou.ru

Вечерская Светлана Евгеньевна
кандидат химических наук, доцент, доцент кафедры информационных систем в экономике и управлении, Российский новый университет, Москва. Сфера научных интересов: эффективность управления, эконометрика. Автор более 70 опубликованных работ. ORCID: 0000-0001-6721-1388, SPIN-код: 1343-7927. AuthorID: 48602.
Электронный адрес: s.vecherskaya@bk.ru

распределить тысячи клиентов в разные группы по их платежеспособности, какие данные следует принимать во внимание?

Постановка задачи

В работе поставлена задача: имея базу данных о клиентах, собранных при регистрации их в мегамаркете, количественно оцененной маркетологами магазина для каждого клиента вручную, разработать систему на основе машинного обучения, которая способна автоматически классифицировать всех оставшихся, а также новых клиентов по показателям их платежеспособности.

В рамках поставленной задачи необходимо было решить следующие подзадачи:

- на основе собранной базы данных клиентов с проставленными группами их платежеспособности провести анализ и выяснить, какие критерии, указанные клиентами при регистрации, наиболее влияют на предполагаемую платежеспособность;
- нормализовать собранные данные клиентов, чтобы они подходили под выбранный метод машинного обучения;
- обучить модель и протестировать ее работу на клиентах, уже с указанными группами платежеспособности;
- доработать систему для улучшения точности ее работы.

Понятие случайного леса (ансамбль) и его описание

В качестве инструмента классификации была выбрана модель случайного леса (Random Forest). Суть модели заключается в том, что обучающая выборка (множество признаков) разделяется на несколько подмножеств, причем эти подмножества формируются случайным образом, где значения в каждом наборе данных могут повторяться. Алгоритм формирования таких подмножеств называется бутстрэп (от англ. Bootstrap) [1]. На Рисунке 1 изображен принцип разбиения обучающей выборки по данному алгоритму.

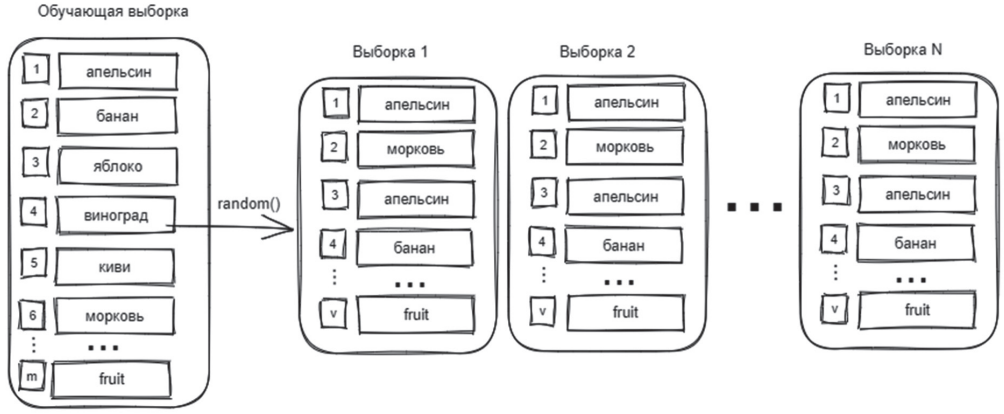

Рисунок 1. Алгоритм бутстрэп

Источник: здесь и далее рисунки и схемы выполнены авторами.

Как показано на Рисунке 1, обучающая выборка разделяется на определенное количество выборок N, причем должно выполняться условие v < m, где v – объем данных в новых сформированных выборках, а m – объем данных обучающей выборке.

Для каждого из сформированных подмножеств применяется алгоритм дерева решений, который строится независимо для каждого подмножества. Затем результат решения каждого дерева усредняется. При построении дерева решений для каждой вершины набор признаков также определяется случайным образом, причем набор признаков для каждой последующей вершины определяется на основе признаков той вершины, которая расположена выше. На Рисунке 2 показан принцип построения дерева решений для каждой сформированной выборки.

Как показано на Рисунке 2, первая вершина содержит набор случайно отобранных объектов (или подмножество) v и предикат $B_v(x)$, который определяет условие перехода на ту или иную ветвь. На следующем уровне для предиката $B_{v_1}(x)$ определяется подмножество v_1, которое представляет собой случайно отобранные объекты из подмножества v, причем $v_1 < v$, и, таким образом, идет формирование предиката для каждой вершины до последней ветви дерева.

Таким образом, для каждой выборки N строится независимое дерево $b(x)$, результаты выборок затем усредняются, что можно записать в виде

$$a_x = \frac{1}{N}\sum_{i=1}^{N} b_i(x),$$

где a_x – набор обучающих алгоритмов (в данном случае бинарных деревьев), который также называется ансамблем алгоритмов; N – количество деревьев; i – счетчик для деревьев; b – решающее дерево; x – сгенерированная выборка.

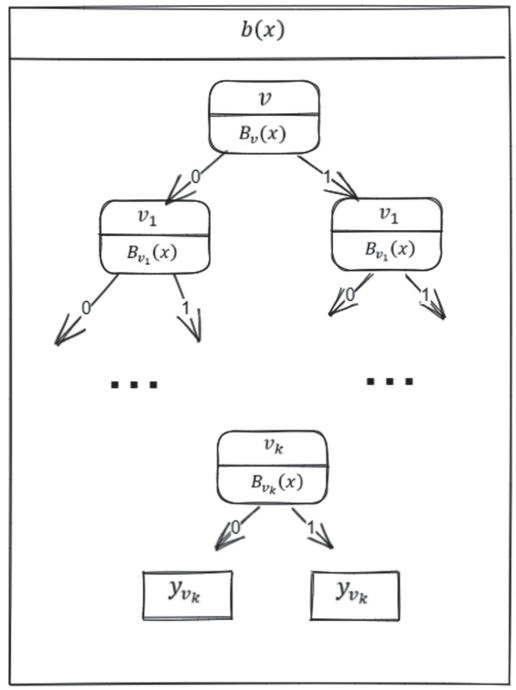

Рисунок 2. Построение решающего дерева

Важность (значимость) признаков

Исходя из вышесказанного, можно сделать вывод, что для построения случайного леса будут выбираться наилучшие признаки. Необходимо определить, какие признаки будут иметь большее значение для построения случайного леса в данной конкретной задаче. Для этого применяется метод Feature Importance [2].

Таким образом, на основе базы данных клиентов с уже проставленными группами их платежеспособности модель обучается, тестируется, и выделяются признаки, которые больше всего повлияли на построение случайного леса. На Рисунке 3 представлена гистограмма, показывающая наиболее важные признаки, которые выделила модель при обучении.

По приведенному графику можно отметить, что самую главную роль играет возраст, а на втором месте находится семейный статус клиента. Кроме того, существенное значение имеет размер семьи клиента. Полученные данные нетрудно обосновать логически: чем старше человек, тем больше его потребности, а если у человека есть семья, то он вынужден покупать больше товаров. Далее повторное обучение модели производится только по наиболее важным признакам, что позволяет улучшить точность результатов.

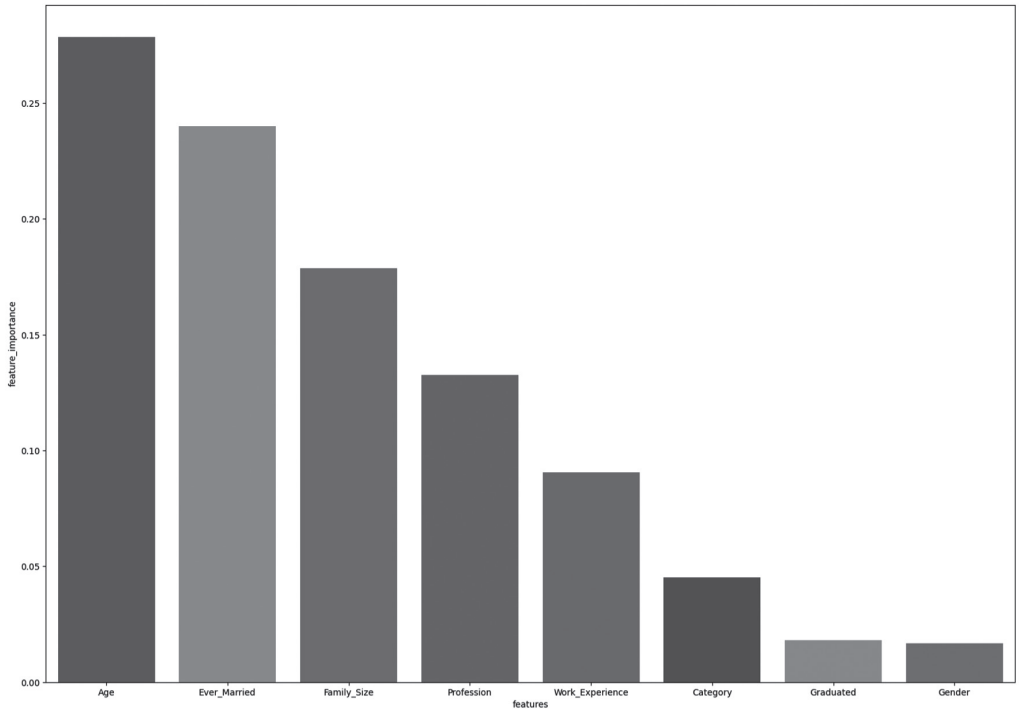

Рисунок 3. Наиболее важные признаки

Тестирование модели

Тестирование модели производилось в виртуальной среде Google Colab на языке программирования Python [3]. Работа с данными производилась с помощью библиотеки Pandas [4], а построение модели осуществлялось с помощью библиотеки Scikit-learn [5].

Имеющийся dataset насчитывает 10695 строк, собранных мегамаркетом. В нем присутствуют такие колонки, как:

ID – уникальный идентификатор клиента;
Gender – пол;
Ever married – семейное положение;
Age – возраст;
Graduated – является ли клиент выпускником;
Profession – профессия;
Work experience – опыт работы (в годах);
Spending score – потребительская способность;
Family size – количество членов семьи (включая самого клиента);
Category – анонимизированная категория для клиента.

Разработка системы

На Рисунке 4 представлены данные без предобработки из исходного датасета.

Необходимо провести первоначальную обработку этих данных, а именно избавиться от дублирующих записей, заполнить пустые значения или удалить их. Это необходимо сделать для корректной работы модели, в противном случае качество обучения алгоритма

будет низким. На Рисунке 5 представлена сумма пропущенных значений по всем колонкам в исходной выборке.

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Category
0	462809	Male	No	22	No	Healthcare	1.0	Low	4.0	Cat_4
1	462643	Female	Yes	38	Yes	Engineer	NaN	Average	3.0	Cat_4
2	466315	Female	Yes	67	Yes	Engineer	1.0	Low	1.0	Cat_6
3	461735	Male	Yes	67	Yes	Lawyer	0.0	High	2.0	Cat_6
4	462669	Female	Yes	40	Yes	Entertainment	NaN	High	6.0	Cat_6
...	...	...	...	...	...	...	...	...	...	...
2622	467954	Male	No	29	No	Healthcare	9.0	Low	4.0	Cat_6
2623	467958	Female	No	35	Yes	Doctor	1.0	Low	1.0	Cat_6
2624	467960	Female	No	53	Yes	Entertainment	NaN	Low	2.0	Cat_6
2625	467961	Male	Yes	47	Yes	Executive	1.0	High	5.0	Cat_4
2626	467968	Female	No	43	Yes	Healthcare	9.0	Low	3.0	Cat_7

10695 rows × 10 columns

Рисунок 4. Данные без предобработки

```
df.isna().sum()

Gender              0
Ever_Married      190
Age                 0
Graduated         102
Profession        162
Work_Experience  1098
Spending_Score      0
Family_Size       448
Category          108
dtype: int64
```

Рисунок 5. Сумма пропущенных значений

На Рисунке 6 показан код заполнения некоторых пропущенных значений, удаление дублирующих и пустых записей.

Признаки Profession и Ever_Married заполняются самым популярным значением – Artist, остальные заполняются средним по значениям в колонке. На следующем этапе необходимо преобразовать все категориальные переменные в числовые. Для этого нужно воспользоваться классом Label Encoder библиотеки sklearn.preprocessing. Код перевода строковых значений числа показан на Рисунке 7.

На Рисунке 7 можно видеть, что после предобработки данных размерность датасета уменьшилась до 10335 записей, что не является критичным по отношению к исходному объему данных.

Далее предобработанные данные разбиваются на две выборки – train и test – в соотношении 80 на 20 (Рисунок 8).

```
df['Profession'] = df['Profession'].fillna(df['Profession'].mode()[0])
df['Work_Experience'] = df['Work_Experience'].fillna(df['Work_Experience'].mean())
df['Family_Size'] = df['Family_Size'].fillna(df['Family_Size'].mean())
df['Ever_Married'] = df['Ever_Married'].fillna(df['Ever_Married'].mode()[0])
```

```
df.duplicated().sum()
```

```
136
```

```
df = df.drop_duplicates()
```

```
df.dropna(axis=0)
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Category
0	462809	Male	No	22	No	Healthcare	1.000000	Low	4.0	Cat_4
1	462643	Female	Yes	38	Yes	Engineer	2.619777	Average	3.0	Cat_4
2	466315	Female	Yes	67	Yes	Engineer	1.000000	Low	1.0	Cat_6
3	461735	Male	Yes	67	Yes	Lawyer	0.000000	High	2.0	Cat_6
4	462669	Female	Yes	40	Yes	Entertainment	2.619777	High	6.0	Cat_6
...	...	...	...	...	...	...	...	...	...	...
2622	467954	Male	No	29	No	Healthcare	9.000000	Low	4.0	Cat_6
2623	467958	Female	No	35	Yes	Doctor	1.000000	Low	1.0	Cat_6
2624	467960	Female	No	53	Yes	Entertainment	2.619777	Low	2.0	Cat_6
2625	467961	Male	Yes	47	Yes	Executive	1.000000	High	5.0	Cat_4
2626	467968	Female	No	43	Yes	Healthcare	9.000000	Low	3.0	Cat_7

10355 rows × 10 columns

Рисунок 6. Заполнение пропусков в колонках

```
from sklearn.preprocessing import LabelEncoder, normalize
le = LabelEncoder()

cols_e = ['Profession', 'Category','Gender','Ever_Married','Graduated','Spending_Score']

for col in cols_e:
    df_copy[col] = le.fit_transform(df_copy[col])
```

```
df_copy = pd.get_dummies(df_copy, columns=['Gender', 'Ever_Married', 'Graduated'])
```

```
df_copy = df_copy.rename(columns={'Gender_0': 'Gender_Female',
                                  'Gender_1': 'Gender_male',
                                  'Ever_Married_0': 'Ever_Married_No',
                                  'Ever_Married_1': 'Ever_Married_Yes',
                                  'Graduated_0': 'Graduated_No',
                                  'Graduated_1': 'Graduated_Yes'})
```

```
df_copy
```

	Age	Profession	Work_Experience	Spending_Score	Family_Size	Category	Gender_Female	Gender_male	Ever_Married_No	Ever_Married_Yes	Graduated_No	Graduated_Yes
0	22	5	1.000000	2	4.0	3	0	1	1	0	1	0
1	38	2	2.619777	0	3.0	3	1	0	0	1	0	1
2	67	2	1.000000	2	1.0	5	1	0	0	1	0	1
3	67	7	0.000000	1	2.0	5	0	1	0	1	0	1
4	40	3	2.619777	1	6.0	5	1	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
2622	29	5	9.000000	2	4.0	5	0	1	1	0	1	0
2623	35	1	1.000000	2	1.0	5	1	0	1	0	0	1
2624	53	3	2.619777	2	2.0	5	1	0	1	0	0	1
2625	47	4	1.000000	1	5.0	3	0	1	0	1	0	1
2626	43	5	9.000000	2	3.0	6	1	0	1	0	0	1

10355 rows × 12 columns

Рисунок 7. Кодировка признаков

```
X = df_copy.drop('Spending_Score', axis=1)
y = df_copy['Spending_Score']

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Рисунок 8. Выборки train, test

Обучение модели происходит в два этапа. Сначала обучается модель со стандартными параметрами, находятся наиболее важные признаки. Затем модель обучается еще раз при настраиваемых параметрах. Процесс обучения модели представлен на Рисунке 9.

```
[39] clf_rf = RandomForestClassifier()

[40] clf_rf.fit(x_train, y_train)

     ▾ RandomForestClassifier
     RandomForestClassifier()

[41] print("Training Accuracy :", clf_rf.score(x_train, y_train))
     print("Testing Accuaracy :", clf_rf.score(x_test, y_test))

     Training Accuracy : 0.9758958701590873
     Testing Accuaracy : 0.6780205655526992
```

Рисунок 9. Первичное обучение модели

Как можно заметить, точность данных на train-выборке составила 97 %; в свою очередь, точность на тестовой выборке составила 67 %, что является неудовлетворительным результатом. Предположительно, для достижения лучшей точности на тестовой выборке необходимо провести обучение модели с настраиваемыми параметрами. Обучение модели производится с помощью сетки, в которой перебираются все заданные параметры. Иными словами, обучение производится последовательным перебором различных параметров. На Рисунке 10 представлен итоговый результат обучения.

В ходе тестирования изменялись такие параметры, как количество деревьев в лесу, максимальная глубина дерева решений, а также соотношение данных в обучающей и тестовой выборках. В итоге наилучший результат дали параметры:

n_estimators = 250;

max_depth = 10.

Несмотря на небольшой прирост точности на тестовой выборке, качество обучения всё еще оставляет желать лучшего. Это может быть связано непосредственно с данными в датасете.

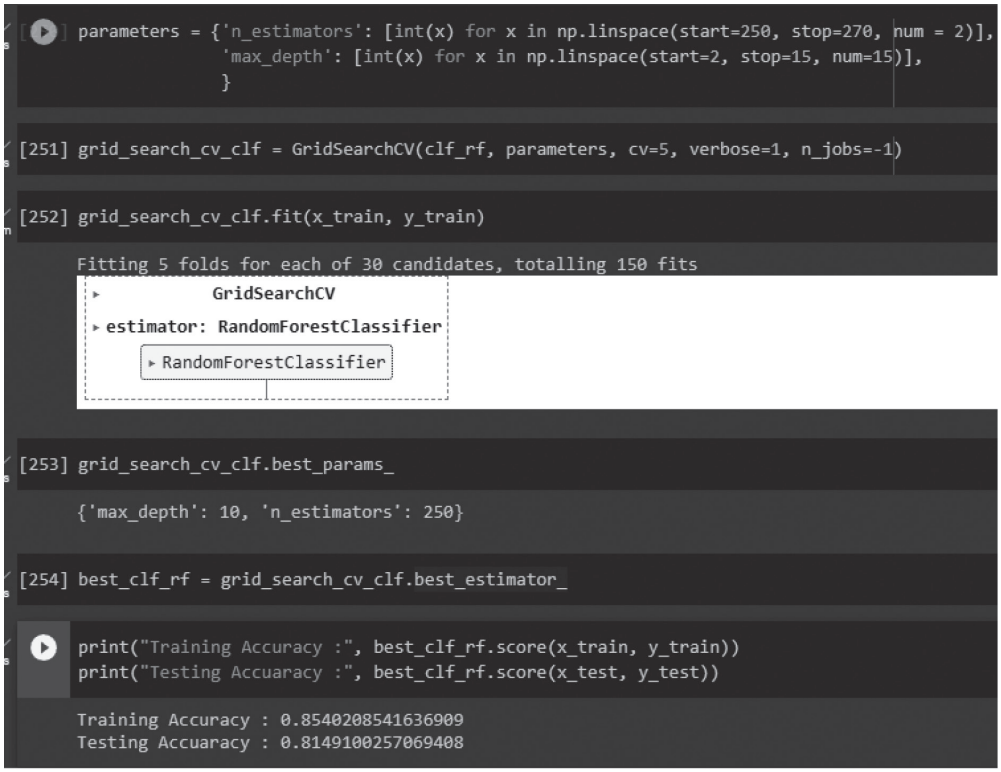

Рисунок 10. Финальный результат обучения модели

Оценка модели

Для оценки качества работы модели была выбрана метрика Precision (точность в пределах класса). Precision – доля объектов, названных классификатором положительными и при этом действительно являющихся положительными. На Рисунке 11 показан результат оценки качества модели.

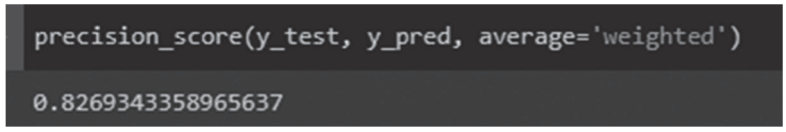

Рисунок 11. Precision score

На Рисунке 12 представлен финальный результат работы программы в виде датасета с колонкой предсказанных групп платежеспособности клиентов моделью машинного обучения.

```
df['Prediction'] = clf_rf.predict(X)

df['Prediction'] = le.inverse_transform(df['Prediction'])

df
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Category	Prediction
0	462809	Male	No	22	No	Healthcare	1.000000	Low	4.0	Cat_4	Low
1	462643	Female	Yes	38	Yes	Engineer	2.619777	Average	3.0	Cat_4	Average
2	466315	Female	Yes	67	Yes	Engineer	1.000000	Low	1.0	Cat_6	Low
3	461735	Male	Yes	67	Yes	Lawyer	0.000000	High	2.0	Cat_6	Low
4	462669	Female	Yes	40	Yes	Entertainment	2.619777	High	6.0	Cat_6	High
...	...	...	...	...	...	...	...	...	...	...	...
2622	467954	Male	No	29	No	Healthcare	9.000000	Low	4.0	Cat_6	Low
2623	467958	Female	No	35	Yes	Doctor	1.000000	Low	1.0	Cat_6	Low
2624	467960	Female	No	53	Yes	Entertainment	2.619777	Low	2.0	Cat_6	Low
2625	467961	Male	Yes	47	Yes	Executive	1.000000	High	5.0	Cat_4	High
2626	467968	Female	No	43	Yes	Healthcare	9.000000	Low	3.0	Cat_7	Low

10355 rows × 11 columns

Рисунок 12. Результат предсказаний

Заключение

Результатом проведенной работы является система на основе машинного обучения, которая способна автоматически классифицировать всех клиентов магазина по группам их платежеспособности. Изучена и проанализирована собранная база данных клиентов с проставленными группами их платежеспособности. Благодаря применению метода случайного леса были выделены критерии из базы данных, оказывающие наибольшее влияние на платежеспособность клиента. Затем была проведена нормализация используемых для обучения и тестирования данных. Также была успешно обучена и протестирована модель на основе машинного обучения, способная прогнозировать платежеспособность клиента по заданным критериям. Кроме того, модель была настроена для получения более высокой точности, при этом итоговая точность составила ≈ 81 %.

Данное значение довольно высоко, однако не исключает возможной доработки модели; в частности, можно провести более глубокую нормализацию данных.

Тем не менее уже в предложенном виде данный алгоритм получает возможность внедрения в бизнес-процесс анализа покупателей по их платежеспособности. Данная система будет использоваться в CRM-системе компании, при внесении клиента в базу данных будет проставлять ему предполагаемую группы платежеспособности. При отсутствии подобной системы маркетологам пришлось бы вручную рассчитывать платежеспособность для отдельного клиента. Напротив, внедрение предложенного инструмента позволит автоматизировать процесс и повысить эффективность использования временных и человеческих ресурсов.

Основная же ценность предложенного подхода заключается в возможности его масштабирования как на различные подразделения и уровни центров затрат в пределах одного предприятия или сети, так и в довольно широких пределах B2C бизнес-сегментов, для которых поведенческие профили клиентов имеют ключевое значение.

Литература

1. *Воронцов К.В.* Машинное обучение: курс лекций // MachineLearning.ru. URL: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций,_К.В.Воронцов)# (дата обращения: 17.06.2023).
2. Feature importance // Scikit learn. URL: https://inria.github.io/scikit-learn-mooc/python_scripts/dev_features_importance.html (дата обращения: 17.06.2023).
3. Python. URL: https://www.python.org/ (дата обращения: 17.06.2023).
4. Pandas. URL: https://pandas.pydata.org/ (дата обращения: 17.06.2023).
5. Scikit-learn. Machine Learning in Python // Scikit learn. URL: https://scikit-learn.org/stable/index.html (дата обращения: 17.06.2023).
6. Технологии интеллектуального анализа данных: Методическое пособие. URL: http://ftp.csdep.mephi.ru/kiselev/BD%26DM/Module04/BD%26DM04L.pdf (дата обращения: 17.06.2023).

References

1. Vorontsov K.V. Machine Learning: course of lectures. *MachineLearning.ru*. URL: http://www.machinelearning.ru/wiki/index.php?title=Mashinnoe_obuchenie_(kurs_lektsii,_K.V.Vorontsov)# (accessed 17.06.2023). (In Russian).
2. Feature importance. *Scikit learn.* URL: https://inria.github.io/scikit-learn-mooc/python_scripts/dev_features_importance.html (accessed 17.06.2023).
3. *Python*. URL: https://www.python.org/ (accessed 17.06.2023).
4. *Pandas*. URL: https://pandas.pydata.org (accessed 17.06.2023).
5. Scikit-learn. Machine Learning in Python. *Scikit learn.* URL: https://scikit-learn.org/stable/index.html (accessed 17.06.2023).
6. *Tekhnologii intellektual'nogo analiza dannykh* [Data Mining Technologies: Methodical Guide]. URL: http://ftp.csdep.mephi.ru/kiselev/BD%26DM/Module04/BD%26DM04L.pdf (accessed 17.06.2023).