О. Ю. Вакух, А. А. Радостев, Н. Е. Кузнецова, И. Г. Захарова
Тюменский государственный университет, г. Тюмень
УДК 004.67

ИССЛЕДОВАНИЕ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ДИАГНОСТИКИ ОТИТОВ

Аннотация. В работе представлен сравнительный анализ методов машинного обучения для прогнозирования отитов. Для обучения моделей использовалась выборка, полученная путем выгрузки 3995 записей электронных медицинских карт (ЭМК).

Показано, что наиболее перспективными являются "Gradient boosting" и "Random forest".

Ключевые слова: машинное обучение, прогнозные модели, диагностика отитов, ЭМК.

Введение. Совершенствование процессов организации медицинской помощи за счет внедрения информационных технологий является одним из приоритетных направлений развития здравоохранения России.

Все большее внимание уделяется не только внедрению информационных систем в медицинскую практику и ведению документации в электронном виде, но и применению технологий интеллектуального анализа больших данных с целью повышения качества оказываемой медицинской помощи [1].

Большое количество исследований в медицинской области опирается на анализ медицинских данных взрослого населения [2-4].

В то же время дети остаются без должного внимания и несвоевременное диагностирование отита в раннем возрасте может привести к потере слуха. Это определяет актуальность данной работы.

Проблема исследования. Выбрать наиболее перспективные модели машинного обучения для прогнозирования острого сезонного среднего отита (ОССО) и острого гнойного среднего отита (ОГСО) на данных ЭМК пациентов детского стационара.

Материалы и методы. В работе используются данные, представляющие собой сведения о медицинских случаях пациентов и истории их стационарного лечения. Набор состоит из 3995 случаев.

Таблица 1

Структура данных

Структура данных	*Тип данных*
Медицинская карта	Строка
Основной диагноз	Строка
Осложнения основного диагноза	Строка
Сопутствующие диагнозы	Строка
Регионы	Строка
Пациент	Строка
Дата рождения	Дата
Пол	Строка
Адрес проживания	Строка

Окончание табл. 1

Структура данных	*Тип данных*
Дата поступления	Дата
Дата выписки	Дата
Жалобы	Строка
Объективный статус (при поступлении)	Строка
Результаты исследований	Строка

Один и тот же пациент может быть встречен в записях несколько раз и иметь свою историю лечения, так как каждый медицинский случай рассматривается отдельно.

Для предварительного анализа, обработки данных, а также для обучения моделей классификации был выбран язык программирования Python и его библиотеки: numpy, pandas, matplotlib, seaborn, scikitlearn.

Основными методами классификации послужили Ridge Classifier, Perceptron, Passive-Aggressive, kNN, Random forest, L2 penalty LinearSVC, L1 penalty SGDClassifier, Elastic-Net penalty, NearestCentroid (aka Rocchio classifier), Gradient boosting [5-6].

Предварительная обработка данных. Для использования таблицы в обучении моделей классификации проведена предобработка записей медицинских случаев пациента.

1. Дополнение таблицы данными. Многие столбцы исходной таблицы не несут за собой полезной информации и смысловой нагрузки без обработки. Это касается таких столбцов как: «дата рождения», «дата поступления», «дата выписки», «адрес проживания», «Объективный статус», «Локальный статус», «Результаты исследований». Для этих столбцов были проведены дополнительные этапы и итоговые данные были внесены в таблицу в виде отдельных столбцов:

– Расчет количества лет на момент поступления;

– Расчет времени, проведенного в стационаре, как разность между «датой выписки» и «датой поступления»;

– Классификация регионов на основе столбца «Адрес проживания»: на основе регулярного выражения из текстовой строки выделялся текст, содержащий в себе область/регион;

– Классификация жалоб из колонки «Жалобы»: головная боль, боль в левом ухе, боль в правом ухе, снижение слуха на левое ухо, снижение слуха на правое ухо, гной в носу, повышенная температура были получены с помощью поиска синонимичных конструкций текста, имеющих общий смысл;

– Извлечение роста и веса из «Объективного статуса»;

– Классификация отеков из «Локального статуса»;

– Извлечение результатов общего анализа крови из «Результатов исследований»: эритроциты, лейкоциты, моноциты, лимфоциты, эозинофилы, базофилы, нейтрофилы, сегментоядерные, палочкоядерные, средний объем тромбоцитов, тромбоциты, гематокрит, гемоглобин.

2. Исключение нерелевантных столбцов. Из получившейся таблицы были удалены столбцы: «Медицинская карта», «Адрес проживания», «Жалобы», «Объективный статус», «Локальный статус», «Пациент». «Дата поступления», «Дата выписки», «Дата рождения» также были удалены, так как имелись числовые представления проведенного времени в стационаре и количества лет на момент поступления.

3. Обработка пропущенных значений. Для обработки пропущенных числовых значений был использован SimpleImputer из библиотеки scikitlearn. Для каждого столбца был вычислено средние значения по уже имеющимся значениям в каждом из столбцов и ими заполнены пустые значения. Пустые значения столбцов с категориальными признаками заполнены с использованием SimpleImputer. Использовалось заполнение наиболее часто встречающимися заполненными значениями по каждому из столбцов.

4. Нормализация числовых признаков. Нормализация числовых признаков — это процесс приведения всех числовых данных к единому масштабу, обычно в диапазоне от 0 до 1. Этот шаг чрезвычайно важен в машинном обучении, поскольку различия в масштабах могут привести к тому, что модели будут непропорционально воспринимать большие значения как более важные, что может исказить результаты обучения модели. Нормализация данных помогает улучшить процесс обучения, сокращая время, необходимое для нахождения оптимальных параметров [7-8].

5. Кодирование категориальных признаков. Для кодирования категориальных переменных был применен метод One-Hot Encoding [9]. При One-Hot Encoding каждая категория преобразуется в отдельный столбец/массив.

Таким образом были закодированы следующие столбцы: «Пол», «Осложнение», «Сопутствующий диагноз», «Регион», «головная боль», «боль в левом ухе», «снижение слуха на левое ухо», «гной в носу», «повышенная температура», «Возрастная группа», «Наличие осложнений», «Наличие сопутствующих диагнозов», «Оперативное лечение», «Консультации специалистов». В результате работы алгоритма общее количество столбцов итоговой таблицы составило 67 столбцов.

Результаты исследования моделей прогнозирования. Данные были разделены на 2 выборки согласно стандартному разделению: обучающую выборку 80% и тестовую выборку 20%.

Все модели были обучены на одной и той же выборке данных с одинаковым составом записей обучающей выборки. Для обучения всех моделей был реализован алгоритм, заранее определяющий параметры каждой из моделей и проводящий замер времени и вычисление точности accuracy метрики для сравнения моделей. Результаты времени обучения и точности accuracy метрики представлены в табл. 2.

Таблица 2

Результаты обучения моделей

Модели	*Время обучения (сек.)*	*Точность (accuracy)*
Ridge Classifier	0,050886	0,712140
Perceptron	0,051495	0,673342
Passive-Aggressive	0,058708	0,704631
kNN	0,005627	0,745932
Random forest	1,044671	0,793492
L2 penalty LinearSVC	0,078644	0,742178
L1 penalty LinearSVC	0,796063	0,745932
Gradient boosting	16,925341	0,799750

Исходя из результатов обучения можно заключить, что наиболее успешными оказались ансамблевые методы: "Gradient boosting" и "Random forest". Предполагается что количество показателей, используемых в обучающей выборке, избыточно и требуется провести дополнительный анализ для их исключения.

Наиболее точной оказалась модель "Gradient boosting" с точностью 0,799750.

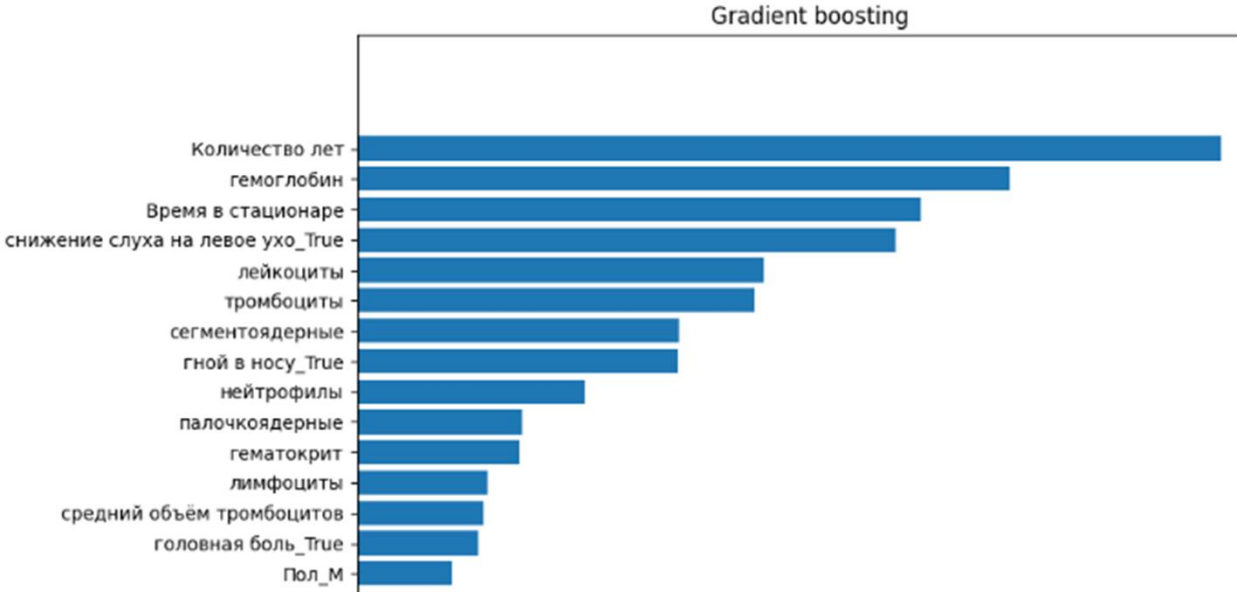

Рис. 1. Значимость признаков предикторов для модели "Gradient boosting"

Также хорошую точность 0,793492 показала модель машинного обучения "Random forest".

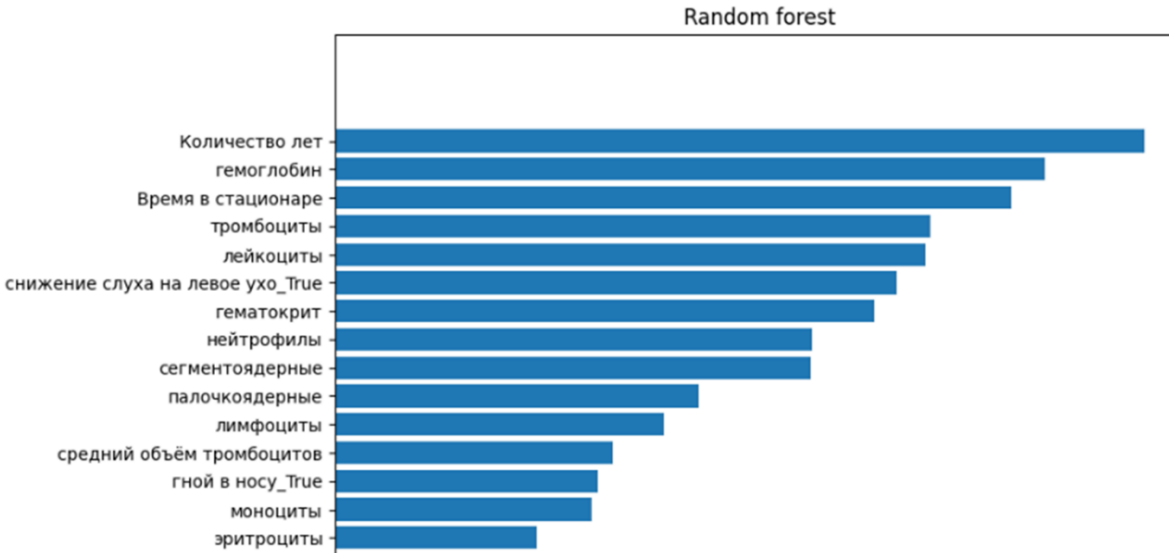

Рис. 2. Значимость признаков предикторов для модели "Random forest"

Первые три признака-предиктора одинаковые для обеих моделей, это количество лет, гемоглобин и время в стационаре. Большая часть признаков связана с показателями анализа крови как у первой модели, так и у второй.

Заключение. В данном исследовании была описана структура данных, полученных из ЭМК, проведен предварительный анализ данных. Построено семейство моделей для прогнозирования ОССО и ОГСО.

В качестве наиболее перспективных для практического использования выделены "Gradient boosting" и "Random forest". В дальнейшем предполагается оптимизировать число признаков-предикторов и интегрирование модели в сервис системы поддержки принятия врачебных решений (СППВР).

СПИСОК ЛИТЕРАТУРЫ

1. Указ Президента РФ от 21 июля 2020 г. № 474 «О национальных целях развития Российской Федерации на период до 2030 года»: [сайт]. — URL: https://base.garant.ru/74404210/ (дата обращения: 04.04.2024). — Текст: электронный.
2. Кузнецова Н.Е., Ястремский А.П., Извин А.И. и др, Прогнозирование острых средних отитов у детей с применением интегрального индекса крови // Folia Otorhinolaryngologiae et Pathologiae Respiratoriae. — 2023. — 29 (2). — С. 69-76.
3. Разработка модели машинного обучения для про-гнозирования числа впервые выявленных пациентов с ВИЧ инфекцией в субъектах Российской Федерации / М.Ю. Котловский [и др.]. — Текст: непосредственный // Врач и информационные технологии. — 2023. — Т. 3. — С. 16-29.
4. Wu W., Huang Q. Clinical analysis of complications of suppurative otitis media in children // — Text: electronic // Lin Chuang er bi yan hou tou Jing wai ke za zhi= Journal of Clinical Otorhinolaryngology, Head, and Neck Surgery. — 2020. — Т. 34, № 7. — С. 587-591.
5. Основные библиотеки для программирования на Python. — URL: https://habr.com/ru/articles/481432 (дата обращения: 06.05.2024). — Текст: электронный.
6. Методами классификации в sklearn.linear: [сайт]. — URL: https://scikit-learn.ru/user_guide/ (дата обращения: 06.05.2024). — Текст: электронный.
7. Исключение определенных столбцов из DataFrame в Pandas: [сайт]. — URL: https://sky.pro/media/isklyuchenie-opredelennyh-stolbczov-iz-dataframe-v-pandas (дата обращения: 06.05.2024). — Текст: электронный.
8. Нормализация данных в Python: [сайт]. — URL: https://pythonist.ru/normalizacziya-dannyh-v-python/ (дата обращения: 06.05.2024). — Текст: электронный.
9. Python: категориальные признаки: [сайт]. — URL: https://alexanderdyakonov.wordpress.com/2016/08/03/python-категориальные-признаки/ (дата обращения: 06.05.2024). — Текст: электронный.