DOI: 10.24412/2413-2527-2023-234-15-26

Использование методов автоматизированного машинного обучения для классификации дорожно-транспортных происшествий

К. К. Мосин
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)
Санкт-Петербург, Россия
msnknstntn@gmail.com

магистр В. Э. Ковалевский, д.т.н. Н. А. Жукова
Санкт-Петербургский Федеральный исследовательский центр Российской академии наук
Санкт-Петербург, Россия
darkeol@mail.ru, nazhukova@mail.ru

Аннотация. **Автоматическое машинное обучение — метод автоматизации процесса машинного обучения, реализующий автоматический выбор подходящих алгоритмов машинного обучения и настройку их гиперпараметров для создания моделей машинного обучения. Применение AutoML для прогнозирования тяжести ДТП может помочь строить модели для оценки степени тяжести ДТП, которые учитывают различные факторы, такие как метеорологические условия, виды дорог и транспортных средств, а также поведение водителя. Использование AutoML может позволить значительно сократить время, необходимое для создания и настройки моделей, а также повысить точность прогнозирования, что, в свою очередь, позволит повысить эффективность организации дорожного движения и уменьшить число ДТП.**

Ключевые слова: **AutoML, машинное обучение, оптимизация гиперпараметров, CASH-проблема.**

Введение

Дорожно-транспортные происшествия (ДТП) — непредвиденные ситуации на дорогах, которые могут привести к травмам, материальным потерям и гибели людей. В основном они происходят из-за нарушения правил дорожного движения, неисправности автомобилей, плохих погодных условий и других негативно влияющих факторов. Статистика [1] показывает, что ДТП являются одной из основных причин смерти и инвалидности во всем мире. Для уменьшения количества ДТП в разных странах принимаются различные меры, такие как обязательное использование ремней безопасности и шлемов, наложение штрафов за нарушение правил дорожного движения, строгая техническая проверка автомобилей и многое другое [2, 3].

Перспективным направлением повышения безопасности на дорогах является прогнозирование тяжести ДТП [4–6]. Такое прогнозирование может помочь выявлять причины ДТП, разрабатывать меры по их предотвращению и повышению уровня безопасности на дорогах. Если иметь информацию о том, какие аварии наиболее вероятны, то можно принимать меры для уменьшения вероятности их возникновения, например улучшая дорожное покрытие или устанавливая дополнительные дорожные знаки. Кроме того, прогнозирование тяжести ДТП может помочь повысить эффективность оказания медицинской помощи. Если заранее известна тяжесть травм, которые могут быть у пострадавших, то можно подготовить необходимое оборудование и обеспечить специализированную медицинскую помощь. Также данная информация может быть полезна для страховых компаний. Зная вероятность того, что водитель попадет в аварию, а также уровень тяжести возможной аварии, страховые компании могут определять стоимость страховки и предоставлять более точные расчеты выплат в случае ДТП [7]. Наконец, прогнозирование тяжести ДТП может быть полезным для правительственных организаций для планирования и управления дорожным движением, предотвращения возможных пробок и оптимизации пути следования скорой помощи.

Методы анализа и прогнозирования ДТП на основе статистических данных включают методы машинного обучения и различные статистические методы.

Машинное обучение

Машинное обучение (Machine Learning) [8, 9] — раздел искусственного интеллекта, изучающий алгоритмы и методы, которые позволяют компьютеру извлекать полезную информацию из данных и строить модели прогнозирования и модели для принятия решения на основе этой информации. Машинное обучение применяется во многих областях, где необходимо обрабатывать большие объемы данных и извлекать из них полезную информацию. Задачи машинного обучения можно классифицировать по нескольким критериям, таким как тип обучения, тип задачи и тип данных. Ниже приводятся несколько основных задач машинного обучения.

1. Классификация (Classification) — задача, когда модель обучается определять, к какому классу принадлежит объект на основе его признаков.

2. Регрессия (Regression) — задача, когда модель обучается предсказывать значения непрерывных признаков для данного объекта.

3. Кластеризация (Clustering) — задача, когда модель обучается определять группы похожих объектов в данных на основе их признаков.

Для решения каждой из указанных задач существует целый ряд алгоритмов машинного обучения, каждый из которых имеет свой набор гиперпараметров и может применяться в определенных ситуациях. Таким образом, выбор подходящего алгоритма и настройка его гиперпараметров является нетривиальной задачей. Автоматическое машинное обучение

Работа выполнена при финансовой поддержке РФФИ в рамках бюджетной темы FFZF-2022-0006.

направлено на упрощение поиска подходящего алгоритма и настройку гиперпараметров и автоматизацию процесса создания модели машинного обучения.

Автоматическое машинное обучение

Автоматическое машинное обучение (Automated Machine Learning, AutoML) [10] — это процесс автоматической оптимизации моделей машинного обучения. Он обеспечивает выбор и настройку наилучшей модели машинного обучения для решения конкретной прикладной задачи. AutoML может применяться к различным задачам машинного обучения, включая классификацию, регрессию и кластеризацию.

AutoML-подход позволяет быстро получить высококачественную модель машинного обучения, даже если у пользователя нет большого опыта в области машинного обучения. Это позволяет сэкономить время и снизить затраты на разработку моделей машинного обучения, а также повысить качество получаемых моделей. Существует несколько подходов к реализации AutoML, включая использование генетических алгоритмов, байесовской оптимизации, а также алгоритмов управляемого случайного поиска. Каждый из этих подходов имеет свои преимущества и недостатки, и выбор определенного подхода зависит от конкретной задачи машинного обучения.

Задача, решаемая AutoML-подходом, — автоматический подбор архитектуры модели и оптимизация гиперпараметров — носит название CASH-проблемы [11].

CASH

CASH (англ. *Combined Algorithm Selection and Hyperparameter optimization* — комбинированный выбор алгоритма и оптимизация гиперпараметров) — это проблема выбора и настройки оптимального алгоритма машинного обучения и его параметров для решения конкретной задачи. Один из способов решения CASH-проблемы — использование методов автоматического машинного обучения, таких как Grid search (поиск по сетке) [12], SMAC (англ. *Sequential Model-Based Algorithm Configuration* — последовательная настройка алгоритма по модели) [13], genetic programming (генетическое программирование) [14].

Grid search работает путем определения набора значений гиперпараметров модели, которые необходимо протестировать. Затем модель обучается и оценивается на каждой комбинации заданных значений гиперпараметров. В конце процесса выбирается набор гиперпараметров, который дал наилучший результат на тестовых данных. Преимуществами Grid search являются:

- простота реализации;
- полнота поиска: Grid search охватывает все возможные значения гиперпараметров, что позволяет в теории найти оптимальные значения;
- репродуцируемость: при наличии фиксированного набора значений гиперпараметров результаты будут воспроизводимыми.

Недостатки Grid search включают:

- вычислительная сложность: Grid search может иметь высокую вычислительно сложность при большом числе гиперпараметров и значений для каждого гиперпараметра;
- неэффективность: Grid search может быть неэффективным, поскольку многие наборы гиперпараметров могут оказаться неинформативными и не давать улучшения качества модели;
- необходимость тщательной настройки: необходимо тщательно выбирать значения гиперпараметров для каждого из наборов.

Альтернативные методы поиска призваны устранить недостатки, присущие Grid search.

SMAC основан на использовании моделей прогнозирования и последовательном испытании различных конфигураций параметров алгоритма. Он использует модели прогнозирования для оценки качества конфигурации параметров на основе результатов прошлых испытаний. Это позволяет SMAC эффективно выбирать следующую конфигурацию, которая, вероятно, приведет к лучшим результатам, учитывая ранее полученные данные. SMAC имеет следующие ключевые особенности:

- использование моделей прогнозирования: SMAC использует модели прогнозирования для оценки качества конфигураций параметров алгоритмов, что позволяет ему быстро и эффективно выбирать следующую конфигурацию;
- последовательное испытание: SMAC выполняет последовательное испытание конфигураций параметров алгоритма, это позволяет ему находить наилучшие параметры даже при наличии множества параметров;
- адаптивность: SMAC адаптивен к характеристикам задачи оптимизации и способу ее решения, он может автоматически настраивать параметры для различных типов задач оптимизации.

Основной идеей генетического программирования является эмуляция естественного отбора в биологической эволюции. Вместо того чтобы создавать программу вручную, генетическое программирование генерирует множество случайных программ и оценивает их по заданной метрике качества. Затем лучшие программы копируются и мутируют, чтобы создать новое поколение программ, которое снова оценивается по метрике качества. Этот процесс повторяется до тех пор, пока не будет достигнуто определенное значение критерия остановки, например достижение заданного уровня точности или заданного числа итераций. К преимуществам генетического программирования относятся:

- автоматическое создание программ без участия человека;
- способность генерировать программы, которые могут быть сложными или неочевидными для человека;
- возможность применять генетическое программирование для широкого спектра задач, включая классификацию, регрессию и прогнозирование.

Недостатки генетического программирования включают:

- вычислительная сложность: генетическое программирование может иметь высокую вычислительную сложность при больших размерах пространства решений;
- необходимость большого количества данных: генетическое программирование может потребовать большого количества данных для эффективного обучения модели.

Рассмотренные подходы выбора алгоритма и оптимизации гиперпараметров были реализованы разработчиками различных AutoML-фреймворков, позволяющих решать CASH-проблему на практике.

AutoML-фреймворки

Наиболее распространенными фреймворками для AutoML являются Auto-WEKA [11], H2O AutoML [15], TPOT [16], AutoKeras [17], Auto-Sklearn [18]. Эти фреймворки обычно имеют удобный интерфейс для пользователя и предоставляют возможность автоматического подбора и настройки моделей машинного обучения в соответствии с заданными критериями.

Auto-WEKA — это инструмент автоматического выбора и настройки моделей машинного обучения, основанный на фреймворке WEKA (Waikato Environment for Knowledge Analysis) [19], который является одним из наиболее широко используемых пакетов для анализа данных и машинного обучения. Auto-WEKA имеет следующие особенности:

- автоматический выбор модели машинного обучения;
- автоматическая настройка гиперпараметров;
- использование перекрестной проверки для оценки качества моделей;
- интеграция с фреймворком WEKA и использование его графического интерфейса.

Auto-WEKA предлагает широкий набор классификаторов, включая наивный байесовский классификатор, решающие деревья, случайный лес, SVM, нейронные сети, градиентный бустинг и многие другие. Кроме того, пользователи могут добавлять собственные алгоритмы в WEKA, расширяя тем самым набор доступных классификаторов.

TPOT (Tree-based Pipeline Optimization Tool) создает множество конвейеров обработки данных и моделей машинного обучения, а затем использует генетический алгоритм, чтобы обеспечить эффективный выбор наилучшего конвейера в пространстве возможных конвейеров. TPOT может решать как задачи классификации, так и задачи регрессии.

Особенности TPOT:

- автоматическое построение конвейеров, которые включают предобработку данных, генерацию признаков, отбор признаков и выбор модели машинного обучения;
- гибкость и настраиваемость: TPOT предоставляет пользователю возможность гибкой настройки параметров генетического алгоритма и конвейеров;
- поддержка различных моделей: TPOT может использовать различные модели машинного обучения, включая деревья решений, случайный лес, градиентный бустинг и нейронные сети;
- визуализация результатов: TPOT предоставляет инструменты для визуализации и интерпретации результатов, включая матрицы ошибок, ROC-кривые и оценки значимости признаков.

H2O AutoML использует множество алгоритмов машинного обучения и автоматически подбирает оптимальные гиперпараметры для каждой модели. Он также может автоматически выполнять предварительную обработку данных, включая заполнение пропущенных значений, масштабирование и кодирование категориальных признаков.

H2O AutoML обладает следующими возможностями:

- автоматический выбор моделей;
- автоматическая настройка параметров;
- интерпретация результатов: H2O AutoML предоставляет инструменты для интерпретации результатов, включая визуализацию значимости признаков и интерпретацию прогнозов модели;
- расширенная масштабируемость: H2O AutoML может обрабатывать большие объемы данных и распределять обработку на кластер из нескольких компьютеров.

H2O AutoML предоставляет более двенадцати групп алгоритмов машинного обучения, включая градиентный бустинг, нейронные сети, случайный лес и линейные модели.

AutoKeras — это открытая библиотека автоматического машинного обучения на языке Python для автоматизации процесса построения моделей глубокого обучения. Она обладает следующими возможностями:

- автоматический выбор архитектуры: выбор оптимальной для задачи архитектуры модели;
- автоматическая настройка параметров;
- встроенная поддержка обработки данных: инструменты для предварительной обработки данных, включая заполнение пропущенных значений, масштабирование и кодирование;
- интерпретация результатов, включая визуализацию значимости признаков и интерпретацию прогнозов модели.

AutoKeras может использовать различные типы нейронных сетей, такие как сверточные, рекуррентные, полносвязные и другие, строить различные комбинации слоев, функций активации и других параметров.

Auto-Sklearn — это библиотека автоматического машинного обучения для Python, которая автоматически настраивает гиперпараметры и выбирает модели машинного обучения для решения задач классификации и регрессии. Auto-Sklearn является расширением библиотеки Scikit-learn [20] и предоставляет удобный интерфейс для автоматического поиска оптимальных гиперпараметров и моделей. Особенности Auto-Sklearn:

- автоматический поиск моделей;
- интерактивные инструменты визуализации, которые помогают анализировать и интерпретировать результаты автоматического поиска;
- встроенные инструменты для предварительной обработки данных, включая масштабирование, кодирование категориальных признаков и заполнение пропущенных значений.

Auto-Sklearn включает в себя несколько десятков классификаторов, которые могут быть использованы для решения задач машинного обучения. Среди них можно выделить классификаторы на основе деревьев решений (например, Random Forest, Extra Trees), наивный байесовский классификатор, линейные модели (например, логистическая регрессия, SVM), методы на основе градиентного бустинга (например, XGBoost, LightGBM), а также нейронные сети. В Auto-Sklearn присутствует возможность использования ансамблей моделей, таких как стекинг и бэггинг.

Сравнительные характеристики рассмотренных AutoML-систем приведены в таблице 1.

Таблица 1

Сравнение AutoML систем

Характеристика	**AutoML-библиотека**				
	Auto-WEKA	**Auto-Sklearn**	**TPOT**	**H2O**	**AutoKeras**
CASH	SMAC	SMAC	Genetic program	Grid search	SMAC
Библиотека	WEKA	Scikit-learn	Scikit-learn	H2O	Keras/TensorFlow
ЯП	Java	Python	Python	Java/Python	Python
ГПИ	Есть	Нет	Нет	Есть	Есть
Отображение результатов	Метрики из WEKA	Метрики как у Scikit-learn	Вывод только точности	Вывод встроенных метрик	Метрики и интерпретация прогнозов
Масштабируемость	Нет	Нет	Нет	На кластер	На GPU
Предобработка данных	Вручную	Вручную	Есть, без очистки данных	Есть	Есть
Мета-обучение	Нет	Есть	Нет	Нет	Есть

Использование AutoML для прогнозирования тяжести дорожно-транспортных происшествий

Для проверки применимости различных способов прогнозирования к определению тяжести ДТП был обработан набор исторических данных UK Road Traffic Collision Dataset [21], предоставляемый на платформе Kaggle [22], который содержит информацию о ДТП на дорогах Великобритании за период с 2005 по 2017 годы.

Рассматриваемый набор данных о ДТП состоит из двух частей. Первая часть содержит информацию о произошедших ДТП и включает в себя свыше 1 000 000 записей, каждая из которых описывает отдельное ДТП. Каждое ДТП характеризуется 34 атрибутами, из которых для целей классификации были выбраны следующие:

- Accident_Index — уникальный идентификатор;
- Light_Conditions — условия освещения в момент происшествия;
- Number_of_Casualties — количество пострадавших в ДТП;
- Number_of_Vehicles — количество транспортных средств, участвовавших в ДТП;
- Road_Surface_Conditions — состояние дорожного покрытия в момент происшествия;
- Road_Type — тип дороги, на которой произошло ДТП;
- Speed_limit — ограничение скорости на участке дороги, на котором произошло ДТП;
- Time — время происшествия;
- Weather_Conditions — погодные условия в момент происшествия.

Также в этой части данных содержится целевой атрибут, по которому будет произведена классификация — Accident_Severity — тяжесть ДТП (тяжелое, средней тяжести, легкое).

Вторая часть данных содержит информацию о водителях и транспортных средствах и включает больше 2 000 000 записей, каждая из которых описывает отдельное транспортное средство, участвовавшее в ДТП. Каждая запись описывается 24 атрибутами, из которых для целей классификации были выбраны следующие:

- Accident_Index — уникальный идентификатор;
- Age_Band_of_Driver — возрастная группа водителя;
- Age_of_Vehicle — возраст транспортного средства;
- Propulsion_Code — тип двигателя транспортного средства;
- Sex_of_Driver — пол водителя;
- Vehicle_Leaving_Carriageway — информация о транспортных средствах, покинувших полосу движения;
- Vehicle_Manoeuvre — маневр, выполняемый транспортным средством в момент ДТП;
- Vehicle_Reference — уникальный идентификатор транспортного средства, участвовавшего в ДТП;
- Vehicle_Type — тип транспортного средства, участвовавшего в ДТП;
- 1st_Point_of_Impact — область первого удара транспортного средства.

Данные о ДТП, водителях и транспортных средствах связываются с помощью уникального идентификатора ДТП Accident_Index методом *inner join*, результат внутрен-

него соединения содержит только те строки, которые соответствуют заданному условию, исключая несоответствующие строки из каждой таблицы.

Зависимость между признаками может быть проанализирована с помощью корреляционной матрицы. Корреляционная матрица позволяет проанализировать связи между признаками и выявить наиболее значимые для задачи признаки. Корреляционная матрица может быть полезна в следующих случаях:

- оценка взаимосвязи признаков: если два признака сильно коррелируют между собой, то, возможно, один из них можно исключить из модели без ухудшения ее качества;
- определение наиболее значимых признаков: признаки, которые имеют высокую корреляцию с целевой переменной, могут рассматриваться как наиболее значимые для модели;
- оценка мультиколлинеарности: мультиколлинеарность — наличие нескольких признаков, которые взаимно связаны между собой, что может привести к проблемам при обучении модели;
- проверка гипотез о связи между признаками и целевой переменной.

Корреляционная матрица признаков, используемых для прогнозирования ДТП приведена на рисунке 1.

Соотношение классов тяжести ДТП в наборе данных представлено на рисунке 2.

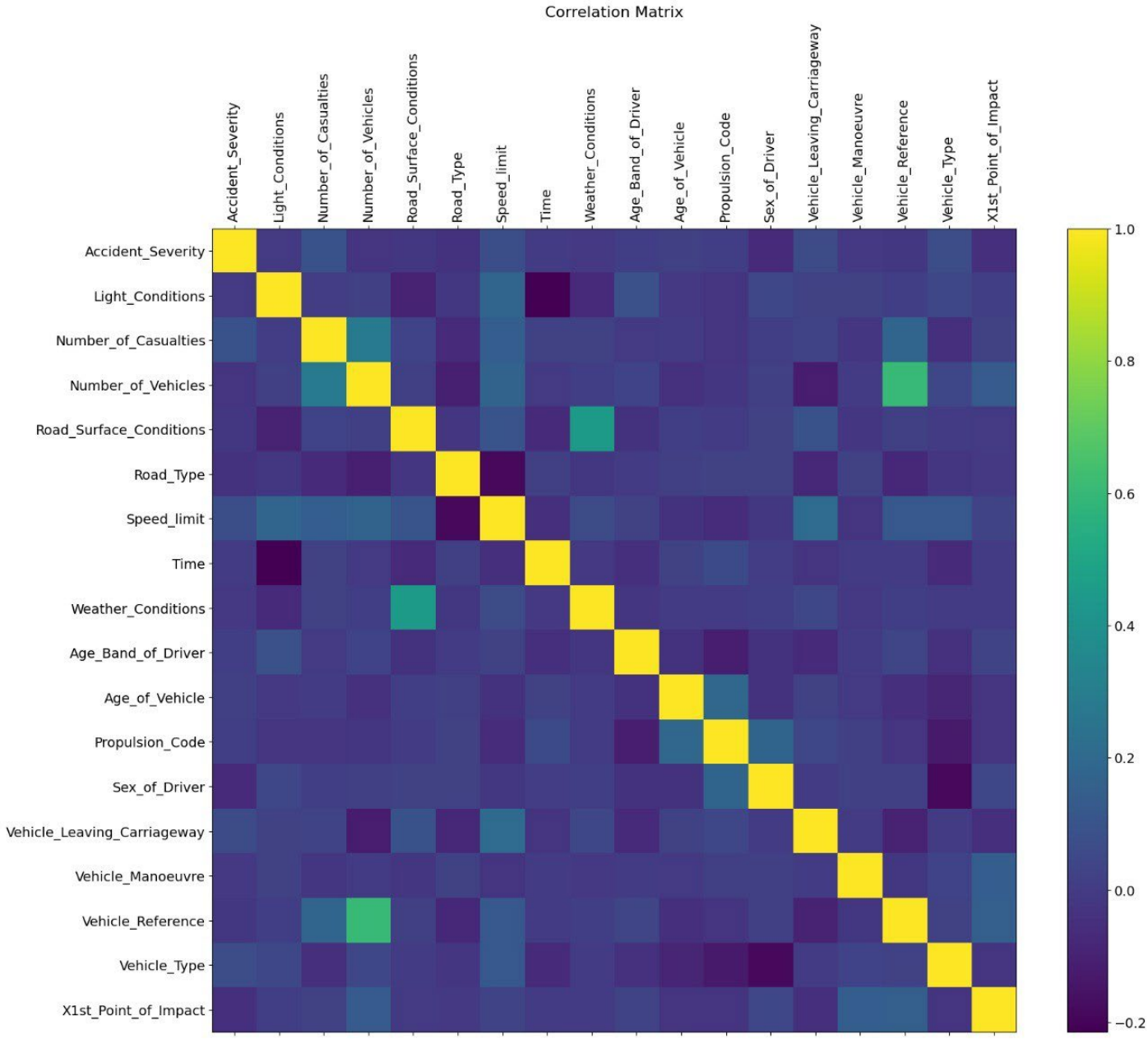

Рис. 1. Корреляционная матрица

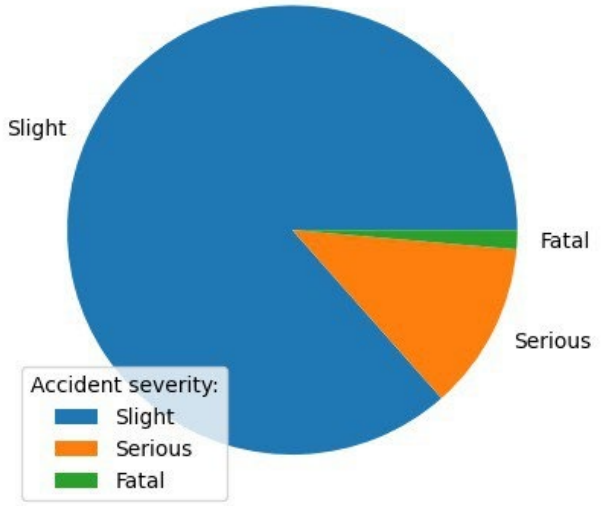

Рис. 2. Исходное соотношение классов тяжести ДТП

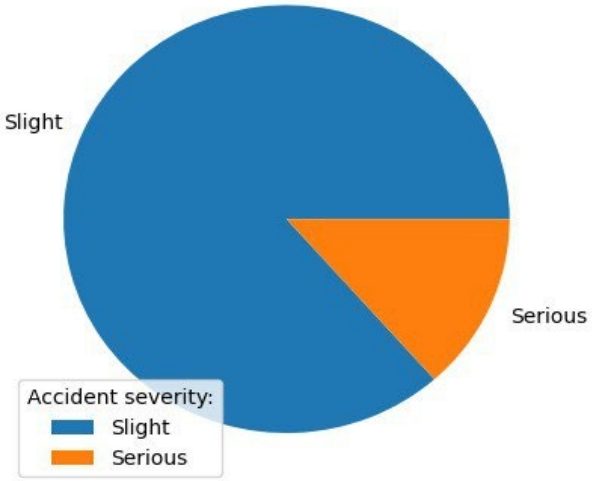

Рис. 3. Результирующее соотношение классов тяжести ДТП

Эксперименты

Из диаграммы на рисунке 2 видно, что решаемая задача — это задача несбалансированной классификации. Несбалансированная классификация возникает в случае, когда в обучающем наборе данных количество объектов в одном классе существенно превосходит количество объектов в другом классе. В такой ситуации модель может не получить достаточно информации об объектах редкого класса и склонна будет присваивать им более часто встречающийся класс.

Для улучшения результатов несбалансированной классификации могут быть использованы следующие методы:

1. Oversampling — увеличение числа объектов редкого класса путем дублирования или генерации.

2. Undersampling — уменьшение числа объектов часто встречающегося класса путем удаления или выборки.

3. Установление весов — установление большего веса для объектов редкого класса, чтобы они оказывали большее влияние на обучение модели.

4. Использование алгоритмов, учитывающих несбалансированность классов, таких как решающие деревья, бустинг или SVM.

5. Использование метрик оценки модели, которые учитывают несбалансированность классов, например: F1 score, ROC AUC и PR AUC.

Для улучшения результатов было проведено объединение фатальных и серьезных ДТП в один класс, чтобы свести мультиклассовую классификацию к бинарной (рис. 3).

Получившийся набор данных был разделен на тренировочный и тестовый наборы с долями 4/5 и 1/5 соответственно. Для сравнения результатов решения задачи несбалансированной классификации каждой из AutoML-систем были использован ряд метрик:

1. Accuracy (точность) — метрика оценки качества классификации, которая показывает, как много объектов было классифицировано правильно относительно общего количества объектов:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

где TP (True Positive) — истинно положительные прогнозы;
TN (True Negative) — истинно отрицательные прогнозы;
FP (False Positive) — ложно положительные прогнозы;
FN (False Negative) — ложно отрицательные прогнозы.

2. Precision (точность) — метрика, показывающая соотношение истинно положительных объектов (TP) к общему количеству объектов, классифицированных как положительные (TP + FP):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

3. Recall (полнота) — метрика, показывающая соотношение истинно положительных объектов (TP) к общему количеству объектов, которые на самом деле являются положительными (TP + FN):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

4. F1 (F-мера) — метрика, которая объединяет точность и полноту в одну метрику. Она вычисляется как гармоническое среднее точности и полноты:

$$F(\beta) = (1 + \beta^2)\frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}.$$

5. ROC AUC (площадь под кривой ROC) — метрика, которая показывает, насколько хорошо модель различает между положительными и отрицательными классами.

6. PR AUC (площадь под кривой точности-полноты) — метрика, которая показывает, насколько хорошо модель различает между положительными и отрицательными классами на основе точности и полноты.

Результаты приведены в таблице 2.

Таблица 2

Результаты работы AutoML-систем

Решение	Метрики						Выбранная модель
	Accuracy	Precision	Recall	F1	ROC AUC	PR AUC	
Auto-WEKA	0,870	0,627	0,081	0,143	0,706	0,311	Random Forest
TPOT	0,867	0,558	0,064	0,115	0,704	0,298	BernoulliNB
H2O	0,868	0,661	0,038	0,072	0,517	0,414	Stacked Ensemble
AutoKeras	0,868	0,597	0,058	0,105	0,526	0,391	Neural networks
Auto-Sklearn	0,867	0,677	0,039	0,074	0,518	0,423	Random Forest

Рассмотрим подробнее выбранные AutoML-системами модели.

Random Forest (случайный лес) — это алгоритм машинного обучения, основанный на методе ансамбля деревьев решений. Random Forest создает множество деревьев решений на основе обучающей выборки. Каждое дерево строится независимо от других деревьев, используя случайное подмножество признаков и случайное подмножество обучающих примеров. При классификации новых примеров каждое дерево в лесу принимает свое решение, и конечный результат определяется голосованием.

Random Forest имеет следующие достоинства:

- более устойчив к переобучению за счет использования случайного выбора признаков и обучающих примеров;
- позволяет эффективно работать с большим количеством признаков и данными высокой размерности;
- может применяться для решения задач классификации и регрессии, где необходима высокая точность.

Random Forest имеет следующие недостатки:

- чувствителен к наличию коррелированных признаков;
- возможны сложности с интерпретацией из-за большого количества деревьев в лесу;
- более низкая скорость работы, чем у одиночных деревьев решений.

BernoulliNB — это наивный байесовский классификатор, который используется для решения задач бинарной классификации, где каждый признак является бинарным (принимает значение 0 или 1).

Основная идея BernoulliNB заключается в том, чтобы построить вероятностную модель на основе обучающей выборки. Для каждого признака BernoulliNB оценивает вероятность того, что он будет равен 1. Для новых примеров классификатор использует эти вероятности и применяет формулу Байеса, чтобы вычислить вероятность принадлежности к каждому классу, и выбирает класс с более высокой вероятностью.

BernoulliNB имеет следующие достоинства:

- быстро обучается и быстро работает при классификации новых примеров;
- может эффективно работать с большими объемами данных и большим количеством признаков;
- применим для решения задач на основе данных с бинарными признаками.

BernoulliNB также имеет следующие недостатки:

- возможно снижение точности результатов в условиях, когда признаки или зависимости между ними не бинарные;
- не учитывает взаимосвязи между признаками, что может привести к снижению точности;
- неприменим для обработки непрерывных признаков.

Stacked Ensemble — это метод ансамбля моделей машинного обучения, который объединяет прогнозы от разных моделей в одну модель для достижения лучшей точности предсказаний.

Основная идея стекинга заключается в том, чтобы обучить несколько базовых моделей на обучающей выборке, затем использовать прогнозы этих моделей в качестве входных данных для более высокоуровневой модели. Каждая модель обучается на той же обучающей выборке, но может использовать разные признаки или методы обучения.

Итоговая модель в стекинге может быть обучена с использованием различных алгоритмов, таких как линейная регрессия, деревья решений или нейронные сети.

Stacked Ensemble имеет следующие достоинства:

- обеспечивает более точные результаты, чем каждая отдельная модель, используемая в стекинге;
- использует различные модели и методы обучения, что позволяет обеспечить лучшие обобщающие способности;
- применим к задачам классификации и регрессии.

Stacked Ensemble имеет следующие недостатки:

- применение большого количества моделей может привести к увеличению времени обучения и увеличению ресурсов, требуемых для обучения и применения модели;
- более высокая сложность настройки и интерпретации, чем для отдельных моделей;
- возможно возникновение проблемы переобучения, если модели, используемые в стекинге, мало отличаются, или используется слабый итоговой алгоритм.

Нейронные сети — это мощный инструмент машинного обучения, основанный на принципах имитации функционирования человеческого мозга, состоящего из множества взаимодействующих нейронов.

Для классификации нейронные сети обучаются на обучающих данных, где каждый пример содержит набор признаков и метку класса. На этапе обучения нейронная сеть настраивает веса между нейронами, чтобы минимизировать функцию ошибки на обучающих данных. После обучения нейронная сеть может быть использована для предсказания меток классов для новых примеров с помощью весов, настроенных на этапе обучения.

Нейронные сети имеют следующие преимущества:

- Могут выявлять и обрабатывать сложные взаимосвязи между признаками и метками классов, которые другие методы выявлять не позволяют.

• Способны работать с неструктурированными данными, такими как изображения, звук и текст.

• Способны к обобщению на новые данные.

Однако, у нейронных сетей также есть существенные ограничения:

• склонны к переобучению;

• требуют большого количества данных для обучения;

• требуют много вычислительных ресурсов и времени для обучения, в частности, высокие требования предъявляются при использовании глубоких нейронных сетей.

Модели, построенные на основе Random Forest и BernoulliNB были обучены с использованием библиотеки Scikit-learn без предварительной подгонки гиперпараметров. Результаты приведены в таблице 3.

Визуализация значений метрик, полученных по результатам работы AutoML-систем и по результатам работы классификаторов из библиотеки Sklearn (таблиц 2 и 3) представлена на рисунке 4. Слева направо приведены результаты Random Forest, BernoulliNB, Auto-WEKA, TPOT, H2O, AutoKeras, Auto-Sklearn.

Таблица 3

Результаты работы классификаторов из библиотеки Sklearn

Классификатор	**Метрики**					
	Accuracy	**Precision**	**Recall**	**F1**	**ROC AUC**	**PR AUC**
Random Forest	0,858	0,443	0,097	0,159	0,539	0,332
BernoulliNB	0,559	0,146	0,447	0,220	0,512	0,335

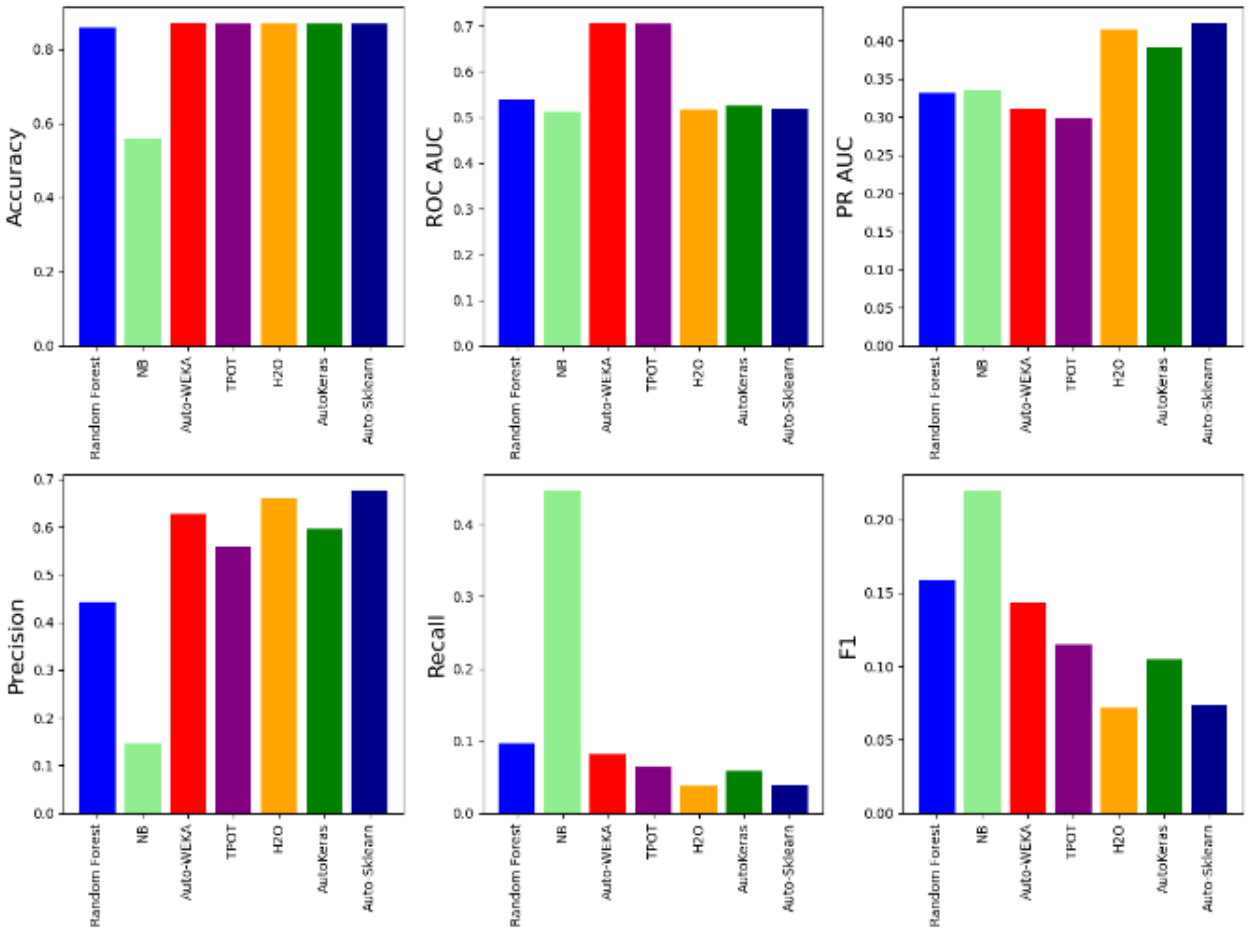

Рис. 4. Визуализация объединения таблиц 2 и 3

Заключение

В результате проведенного исследования была показана применимость методов автоматизированного машинного обучения для прогнозирования тяжести ДТП. В экспериментах были использованы пять AutoML-систем: Auto-WEKA, TPOT, H2O, AutoKeras, Auto-Sklearn. Исходя из результатов экспериментов, можно сделать следующие выводы:

1. Auto-WEKA, TPOT, H2O, AutoKeras и Auto-Sklearn показали более высокую точность, чем классификаторы, настроенные вручную (Random Forest и BernoulliNB), что свидетельствует об эффективности использования AutoML-систем для прогнозирования тяжести ДТП. Однако решалась задача несбалансированной классификации, а для данного класса задач метрика точности (accuracy), не

всегда позволяет получить объективную оценку построенных моделей.

2. Auto-WEKA, TPOT и AutoKeras показали более высокие значения Precision, Recall и F1-score, чем остальные системы, что может указывать на их большую универсальность и способность находить более разнообразные закономерности в данных.

3. Auto-WEKA и Auto-Sklearn выбрали случайный лес в качестве лучшей модели, что может указывать на преимущества этой модели для предсказания тяжести ДТП.

4. Несмотря на то, что Auto-WEKA и TPOT показали более высокие значения AUC и PR AUC, чем другие системы, значения AUC и PR AUC у всех систем остаются на среднем уровне, что может указывать на недостаточную разделимость классов в данных.

Использование AutoML-систем для решения задачи предсказания тяжести ДТП может существенно упростить и ускорить процесс разработки моделей и повысить их точность по сравнению с классификаторами, настроенными вручную, однако на текущий момент времени формируемые модели требуют дальнейшего улучшения. Для улучшения моделей рекомендуется:

1. Использовать больше данных, улучшить их качество. По результатам Recall можно сделать вывод, что ни одна из моделей не справилась с идентификацией всех тяжелых ДТП в обучающей выборке. Это может быть связано с тем, что обучающая выборка содержит недостаточное количество примеров тяжелых ДТП. Для улучшения результатов Recall может потребоваться провести дополнительный анализ данных и увеличить количество примеров тяжелых ДТП в обучающей выборке.

2. Настроить AutoML-системы. Каждая система обладает своим набором параметров, которые, в свою очередь, могут быть настроены. В частности, некоторые системы позволяют задать метрику, по которой будет определяться, что одна модель лучше другой. В данной работе использовались настройки по умолчанию, в которых модели сравниваются по точности предсказания (accuracy).

Литература

1. Global Status Report on Road Safety 2018 // World Health Organization. — 2018. — 17 June.
URL: http://www.who.int/publications/i/item/9789241565684 (дата обращения 27.04.2023)

2. О федеральной целевой программе «Повышение безопасности дорожного движения в 2013–2020 годах»: Постановление Правительства Российской Федерации от 3 октября 2013 г. № 864: ред. от 16 мая 2020 г. № 703.

3. Texas Department of Transportation Strategic Plan 2021–2025. 118 p. URL: http://ftp.dot.state.tx.us/pub/txdot/sla/strategic-plan-2021-2025.pdf (дата обращения 27.04.2023).

4. Taamneh, M. Data-Mining Techniques for Traffic Accident Modeling and Prediction in the United Arab Emirates / M. Taamneh, S. Alkheder, S. Taamneh // Journal of Transportation Safety and Security. 2017. Vol. 9, Is. 2. Pp. 146–166.
DOI: 10.1080/19439962.2016.1152338.

5. Çelik, A. Predicting Traffic Accident Severity Using Machine Learning Techniques / A. Çelik, O. Sevli // Turkish Journal of Nature and Science. 2022. Vol. 11, Is. 3. Pp. 79–83.
DOI: 10.46810/tdfd.1136432.

6. Aci, C. Predicting the Severity of Motor Vehicle Accident Injuries in Adana-Turkey Using Machine Learning Methods and Detailed Meteorological Data / C. Aci, C. Özden // International Journal of Intelligent Systems and Applications in Engineering. 2018. Vol. 6, No. 1. Pp. 72–79.
DOI: 10.18201/ijisae.2018637934.

7. Guelman, L. Gradient Boosting Trees for Auto Insurance Loss Cost Modeling and Prediction // Expert Systems with Applications. 2012. Vol. 39, Is. 3. Pp. 3659–3667.
DOI: 10.1016/j.eswa.2011.09.058.

8. Бишоп, К. М. Распознавание образов и машинное обучение = Pattern Recognition and Machine Learning / пер. с англ. и редакция Д. А. Клюшина. — Санкт-Петербург: Диалектика, 2020. — 960 с.

9. Hastie, T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition / T. Hastie, R. Tibshirani, J. Friedman. — New York: Springer Science and Business Media, 2016. — 767 p. — (Springer Series in Statistics).

10. He, X. AutoML: A Survey of the State-of-the-Art / X. He, K. Zhao, X. Chu // Knowledge-Based Systems. 2021. Vol. 212. Art. No. 106622. 27 p.
DOI: 10.1016/j.knosys.2020.106622.

11. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms / C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown // Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), (Chicago, IL, USA, 11–14 August 2013). — New York: Association for Computing Machinery, 2013. — Pp. 847–855.
DOI: 10.1145/2487575.2487629.

12. Shekhar, S. A Comparative Study of Hyper-Parameter Optimization Tools / S. Shekhar, A. Bansode, A. Salim // Proceedings of the 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), (Brisbane, Australia, 08–10 December 2021). — Institute of Electrical and Electronics Engineers, 2021. — 6 p.
DOI: 10.1109/CSDE53843.2021.9718485.

13. Hutter, F. Sequential Model-Based Optimization for General Algorithm Configuration / F. Hutter, H. H. Hoos, K. Leyton-Brown // Learning and Intelligent Optimization (LION 5): Selected Papers of the Fifth International Conference (Rome, Italy, 17–21 January 2011) / C. A. C. Coello (eds.). — Berlin: Springer-Verlag, 2011. — Pp. 507–523. — (Lecture Notes in Computer Science, Vol. 6683).
DOI: 10.1007/978-3-642-25566-3_40.

14. Application of an Automated Machine Learning-Genetic Algorithm (AutoML-GA) Coupled with Computational Fluid Dynamics Simulations for Rapid Engine Design Optimization / O. Owoyele, P. Pal, A. Vidal Torreira, [et al.] // International Journal of Engine Research. 2022. Vol. 23, Is. 9. Pp. 1586–1601. DOI: 10.1177/14680874211023466.

15. LeDell, E. H2O AutoML: Scalable Automatic Machine Learning / E. LeDell, S. Poirier // AutoML 2020: 7th International Conference on Machine Learning (ICML) Workshop on Automated Machine Learning (Vienna, Austria, 18 July 2020). Paper No. 61. 16 p.

16. Olson, R. S. TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning / R. S. Olson, J. H. Moore // Automated Machine Learning: Methods, Systems, Challenges / F. Hutter, [et al.] (eds.). — Cham: Springer

Nature, 2019. — Pp. 151–160. — (Springer Series on Challenges in Machine Learning).
DOI: 10.1007/978-3-030-05318-5_8.

17. AutoKeras: An AutoML Library for Deep Learning / H. Jin, F. Chollet, Q. Song, X. Hu // Journal of Machine Learning Research. 2023. Vol. 24. Art. No. 6. 6 p.

18. Auto-Sklearn 2.0: Hands-Free AutoML Via Meta-Learning / M. Feurer, K. Eggensperger, S. Falkner, [et al.] // Journal of Machine Learning Research. 2022. Vol. 23. Art. No. 261. 61 p.

19. Holmes, G. WEKA: A Machine Learning Workbench / G. Holmes, A. Donkin, I. H. Witten // ANZIIS '94: Proceedings of Australian and New Zealand Conference on Intelligent Information Systems (Brisbane, Australia, 29 November–02 December 1994). — Institute of Electrical and Electronics Engineers, 1994. — Pp. 357–361.
DOI: 10.1109/ANZIIS.1994.396988.

20. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort, [et al.] // Journal of Machine Learning Research. 2011. Vol. 12. Pp. 2825−2830.

21. Khaliq, S. UK Road Traffic Collision Dataset / S. Khaliq, R. Gregson // Kaggle: Your Machine Learning and Data Science Community. — Обновлено 09.11.2022.
URL: http://www.kaggle.com/datasets/salmankhaliq22/road-traffic-collision-dataset (дата обращения 27.04.2023).

22. Kaggle: Your Machine Learning and Data Science Community. URL: http://www.kaggle.com (дата обращения 27.04.2023).

DOI: 10.24412/2413-2527-2023-234-15-26

Automated Machine Learning Methods for Traffic Accidents Classification

K. K. Mosin
Saint Petersburg Electrotechnical University
Saint Petersburg, Russia
msnknstntn@gmail.com

M. Sc. V. E. Kovalevsky, Grand PhD N. A. Zhukova
St. Petersburg Federal Research Center
of the Russian Academy of Sciences
Saint Petersburg, Russia
darkeol@mail.ru, nazhukova@mail.ru

Abstract. **Automated Machine Learning is an approach for automating the machine learning process by automatically selection of the most suitable machine learning algorithm and tuning its hyperparameters to create a machine learning model. Use of AutoML methods for prediction of traffic accidents severity can help improve the quality of models used to estimate the probability of different accidents based on various factors such as weather conditions, road and vehicle types, and driver behavior. The use of AutoML can significantly reduce the time required to create and tune models, as well as improve the accuracy of traffic accidents severity predictions, which in turn can lead to more efficient traffic management and fewer accidents. In this work we explore the applicability of different Auto ML libraries to the task of traffic accidents prediction and compare them with manually selected and tuned algorithms.**

Keywords: **AutoML, machine learning, hyperparameters optimization, CASH problem.**

REFERENCES

1. Global Status Report on Road Safety 2018, *World Health Organization*. Published online at June 17, 2018. Available at: http://www.who.int/publications/i/item/9789241565684 (accessed 27 Apr 2023).

2. On the Federal Target Program «Improving Road Safety in 2013–2020»: Resolution of the Government of the Russian Federation [O federalnoy tselevoy programme «Povyshenie bezopasnosti dorozhnogo dvizheniya v 2013–2020 godakh»: Postanovlenie Pravitelstva Rossiyskoy Federatsii] from October 3, 2013 No. 864 (last ed. May 16, 2020 No. 703).

3. Texas Department of Transportation Strategic Plan 2021–2025, 118 p. Available at: http://ftp.dot.state.tx.us/pub/txdot/sla/strategic-plan-2021-2025.pdf (accessed 27 Apr 2023).

4. Taamneh M., Alkheder S., Taamneh M. Data-Mining Techniques for Traffic Accident Modeling and Prediction in the United Arab Emirates, *Journal of Transportation Safety and Security*, 2017, Vol. 9, Is. 2, Pp. 146–166.
DOI: 10.1080/19439962.2016.1152338.

5. Çelik A., Sevli O. Predicting Traffic Accident Severity Using Machine Learning Techniques, *Turkish Journal of Nature and Science*, 2022, Vol. 11, Is. 3, Pp. 79–83.
DOI: 10.46810/tdfd.1136432.

6. Aci C., Özden C. Predicting the Severity of Motor Vehicle Accident Injuries in Adana-Turkey Using Machine Learning Methods and Detailed Meteorological Data, *International Journal of Intelligent Systems and Applications in Engineering*, 2018, Vol. 6, No. 1, Pp. 72–79.
DOI: 10.18201/ijisae.2018637934.

7. Guelman L. Gradient Boosting Trees for Auto Insurance Loss Cost Modeling and Prediction, *Expert Systems with Applications*, 2012, Vol. 39, Is. 3, Pp. 3659–3667.
DOI: 10.1016/j.eswa.2011.09.058.

8. Bishop C. M. Pattern Recognition and Machine Learning. Saint Petersburg, Dialektika Computer Publishing, 2020, 960 p.

9. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition. New York, Springer Science and Business Media, 2016, 767 p.

10. He X., Zhao K., Chu H. AutoML: A Survey of the State-of-the-Art, *Knowledge-Based Systems*, 2021, Vol. 212, Art. No. 106622, 27 p. DOI: 10.1016/j.knosys.2020.106622.

11. Thornton C., Hutter F., Hoos H. H., Leyton-Brown K. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13), Chicago, IL, USA, August 11–14, 2013*. New York, Association for Computing Machinery, 2013, Pp. 847–855.
DOI: 10.1145/2487575.2487629.

12. Shekhar S., Bansode A., Salim A. A Comparative Study of Hyper-Parameter Optimization Tools, *Proceedings of the 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Brisbane, Australia, December 08–10, 2021*. Institute of Electrical and Electronics Engineers, 2021, 6 p. DOI: 10.1109/CSDE53843.2021.9718485.

13. Hutter F., Hoos H. H., Leytion-Brown K. Sequential Model-Based Optimization for General Algorithm Configuration. In: *Coello C. A. C. (eds.) Learning and Intelligent Optimization (LION 5): Selected Papers of the Fifth International Conference, Rome, Italy, January 17–21, 2011. Lecture Notes in Computer Science*, Vol. 6683. Berlin, Springer-Verlag, 2011, Pp. 507–523.
DOI: 10.1007/978-3-642-25566-3_40.

14. Owoyele O., Pal P., Vidal Torreira A., et al. Application of an Automated Machine Learning-Genetic Algorithm (AutoML-GA) Coupled with Computational Fluid Dynamics Simulations for Rapid Engine Design Optimization, *International Journal of Engine Research*, 2022, Vol. 23, Is. 9, Pp. 1586–1601. DOI: 10.1177/14680874211023466.

15. LeDell E., Poirier S. H2O AutoML: Scalable Automatic Machine Learning, *AutoML 2020: 7th International Conference on Machine Learning (ICML) Workshop on Automated Machine Learning, Vienna, Austria, July 18, 2020*, Paper No. 61, 16 p.

This work was supported by the Russian Foundation for Basic Research, Project No. FFZF-2022-0006.

16. Olson R. S., Moore J. H. TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning. In: *Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning: Methods, Systems, Challenges*. Cham, Springer Nature, 2019, Pp. 151–160.
DOI: 10.1007/978-3-030-05318-5_8.

17. Jin H., Chollet F., Song Q., Hu X. AutoKeras: An AutoML Library for Deep Learning, *Journal of Machine Learning Research*, 2023, Vol. 24, Art. No. 6, 6 p.

18. Feurer M., Eggensperger K., Falkner S., et al. Auto-Sklearn 2.0: Hands-Free AutoML Via Meta-Learning, *Journal of Machine Learning Research*, 2022, Vol. 23, Art. No. 261, 61 p.

19. Holmes G., Donkin A., Witten I. H. WEKA: A Machine Learning Workbench, *ANZIIS '94: Proceedings of Australian and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia, November 29–December 02, 1994*. Institute of Electrical and Electronics Engineers, 1994, Pp. 357–361. DOI: 10.1109/ANZIIS.1994.396988.

20. Pedregosa F., Varoquaux G., Gramfort A. et al.Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 2011, Vol. 12, Pp. 2825−2830.

21. Khaliq S., Gregson R. UK Road Traffic Collision Dataset, *Kaggle: Your Machine Learning and Data Science Community*. Last updated at November 09, 2022. Available at: http://www.kaggle.com/datasets/salmankhaliq22/road-traffic-collision-dataset (accessed 27 Apr 2023).

22. Kaggle: Your Machine Learning and Data Science Community. Available at: http://www.kaggle.com (accessed 27 Apr 2023).