**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

Институт/факультет «Академия “Высшая Инженерная Школа”, АВИШ»

Специальность/Направление подготовки Информатика и вычислительная техника

**КУРСОВАЯ РАБОТА**

**по дисциплине:** Машинное обучение и анализ данных

|  |  |
| --- | --- |
| **на тему:** | “Прогнозирование ДТП на основе погодных условий с помощью машинного обучения” |

|  |  |
| --- | --- |
| **Студент группы ШАД 212** | **(Воронин А.А. )** |
| **Научный руководитель** | **(Атамасов Н.В. )** |

**Москва 2023 г.**

Оглавление

[Основные термины и сокращения 3](#_Toc155673406)

[Введение 4](#_Toc155673407)

[Описание задачи 5](#_Toc155673408)

[Основные шаги выполнения. EDA 6](#_Toc155673409)

[Основные шаги выполнения. Подготовка данных 10](#_Toc155673410)

[Основные шаги выполнения. Разделение на выборки 17](#_Toc155673411)

[Основные шаги выполнения. Построение модели 18](#_Toc155673412)

[Основные шаги выполнения. Лучшая модель 21](#_Toc155673413)

[Основные шаги выполнения. Интерпретация лучшей модели 24](#_Toc155673414)

[Разработка приложения для тестирования 26](#_Toc155673415)

[Заключение 27](#_Toc155673416)

[Список использованных источников и программ 28](#_Toc155673417)

[Приложение 29](#_Toc155673418)

# Основные термины и сокращения

ДТП – дорожно-транспортное происшествие.

AUC ROC - метрика, позволяющая оценить качество модели.

Датасет – набор данных, используемый для анализа и машинного обучения.

EDA – (exploratory data analysis) разведочный анализ данных (этап работы с датасетом).

Nan – (not a number) величина, не являющееся числом.

Енкодинг – процесс, с помощью которого категориальные переменные преобразуются в подходящую ​​алгоритмам машинного обучения форму с помощью кодирования.

OHE – (one-hot-encoding) кодирование категориальной переменной путем разбития на n столбцов, исходя из количества уникальных значений в этом столбце.

Пайплайн - последовательность шагов обработки данных и моделирования, объединенных вместе.

Переобучие – это явление, при котором модель машинного обучения слишком точно подстроена под обучающий набор данных, что приводит к плохим результатам на новых

Недообучение – это явление, при котором модель машинного обучения не смогла полностью извлечь информацию из обучающего набора данных и поэтому показывает плохие результаты как на обучающих, так и на новых данных.

Frontend (фронтенд) – лицевая часть программной системы, пользовательский интерфейс. В контексте этой курсовой работы – веб-приложение.

Backend (бэкенд) – внутренняя часть серверного приложения, скрытая от пользователя.

Ендпоинт – конечная точка (адрес) на сервере, на который отправляются запросы для их дальнейшего выполнения.

REST – (representational state transfer) протокол передачи данных.

API – (application programming interface) – программный интерфейс для взаимодействия программ дуг с другом по определенным правилам.

# Введение

ДТП представляют собой серьезную проблему, они приводят к материальным потерям, физическим травмам и потере человеческих жизней. Основываясь на данных о наличии и отсутствии аварий и погоды, машинное обучение позволяет спрогнозировать возникновение возможного инцидента на дороге для дальнейшего принятия мер.

В ходе работы будет проведен анализ данных, а затем построение базовой автоматизированной и собственной моделей для прогнозирования величины “будет или не будет ДТП”, учитывая погодные условия.

# Описание задачи

Основной задачей стоит цель построить модель машинного обучения, которая на основе погодных условий будет составлять прогноз такой величины, как “будет или не будет ДТП”. Такая задача называется бинарной классификацией. Для того, чтобы оценить качество модели, будем рассматривать такие метрики, как матрица ошибок и AUC ROC кривая. Также необходимо провести моделирование с помощью библиотеки автоматизированного машинного обучения и собственной, выбранной модели, и сравнить их.

# Основные шаги выполнения. EDA

В данной курсовой работе были использованы данные из Excel по ДТП – “DataDTPкор.xlsx” и по погоде – “ Pogoda.xlsx” После загрузки объединил все в один датасет по признаку “join\_key”. Затем выбрал интересующие регионы, относящиеся к северной части. Начальный объем данных стал из себя представлять 46786 строк и 26 столбцов, представляющие признаки.

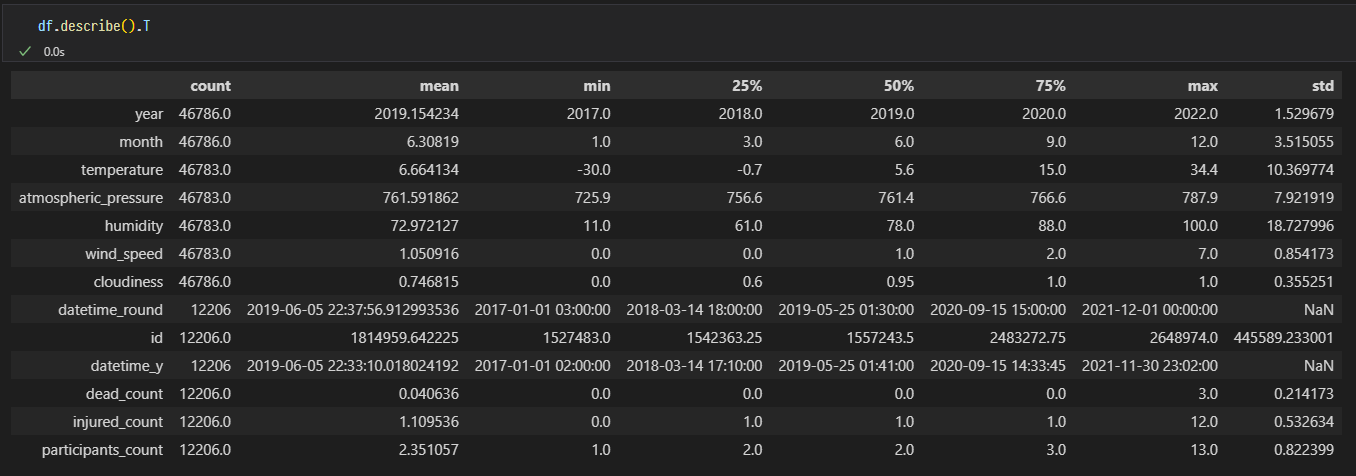


Рис. 1

Описание датасета.

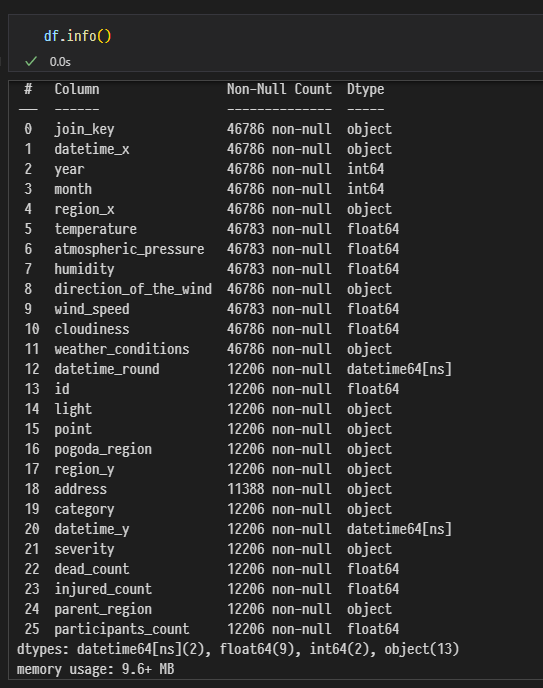


Рис. 2

Информация по датасету.

Основными действиями для этапа EDA стали удаление дубликатов, приведение корректного типа в некоторых столбцах, заполнение и удаление пропусков, удаление ненужных признаков.



Рис. 3

Столбец с датой записи я разбил на “hour” (в какой час была создана запись) и на “season” (в какой сезон создана запись: зима, весна, лето или осень).

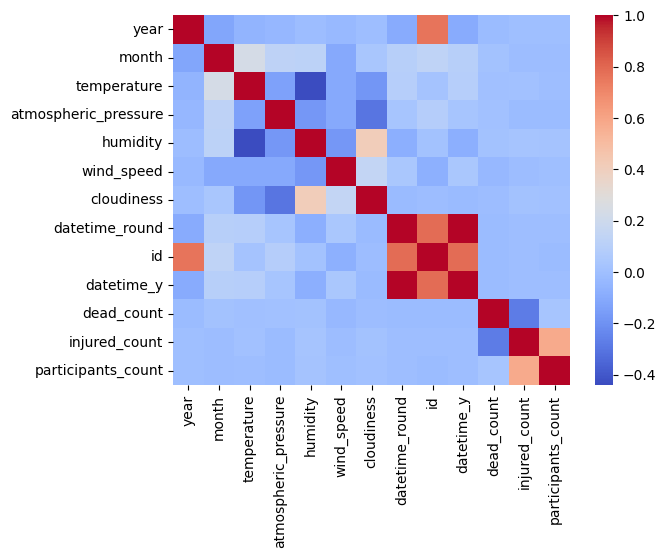


Рис. 4

Построив корреляционную матрицу, заметим, что есть признаки, которые имеют мультиколлинеарность. Это может плохо повлиять на способность предсказания нашей модели, поэтому в следующих шагах нужно исправить.

# Основные шаги выполнения. Подготовка данных

Особое внимание уделил признакам, которые напрямую связаны с исходом ДТП. Для дальнейшего корректного их использования я их изменил.

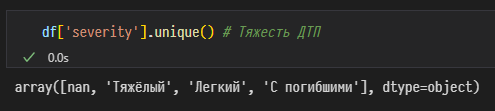


Рис. 5

Признак “severity” описывает тяжесть происшествия. Его пропуск (nan), означает, что ДТП не было.

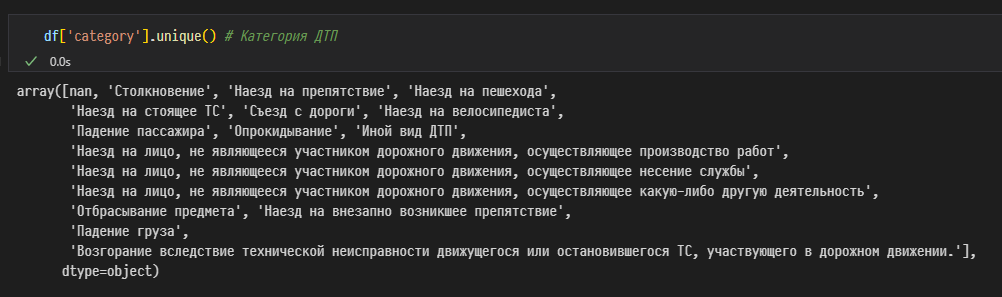


Рис. 6

Признак “category” описывает категорию происшествия (что произошло). Его пропуск означает, что ДТП не было.

Исходя из того, что значение nan в этих признаках присуще отсутствию ДТП, я заменил в них значения: 1 и 0 в зависимости от наличия записи. Также в численных признаках “dead\_count” (количество летальных исходов), “injured\_count” (количество пострадавших) и “participants\_count” (количество участниках) заменил пропуски (отсутствие записи о происшествии) на 0.

Было принято решение создать столбец “is\_accident”, который отображает наличие происшествия, если значение 1 или его отсутствие со значением 0. Его формирование происходило исходя из признаков, прямо отображающих наличие записи о ДТП – “dead\_counts”, “injured\_counts”, “participants\_count”, “severity” и “category”.

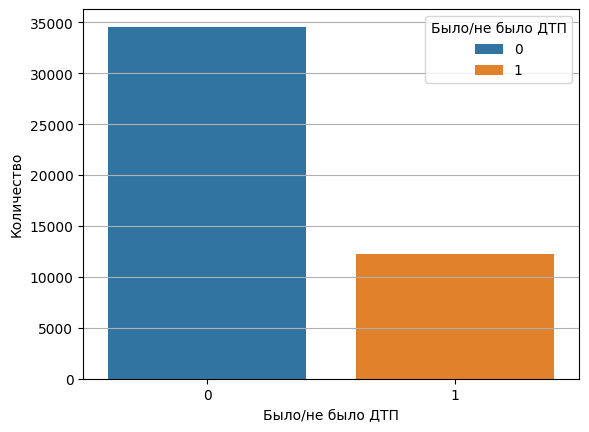


Рис. 8

Заметим, что количество записей отсутствия ДТП и наличия сильно отличается. Это может плохо повлиять на обучение модели. Поэтому на этапе разделения на выборки, нужно будет провести баланс классов, то есть примерно уравнять их количество.

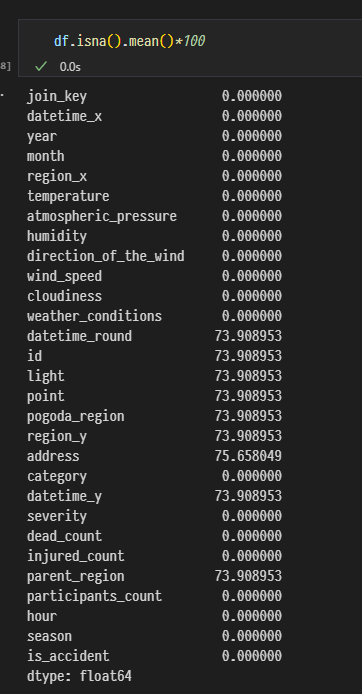


Рис. 9

Процент пропусков в каждом столбце. Заметим, что в основном процент достаточно большой, а также некоторые признаки никак не понадобятся при обучении модели.

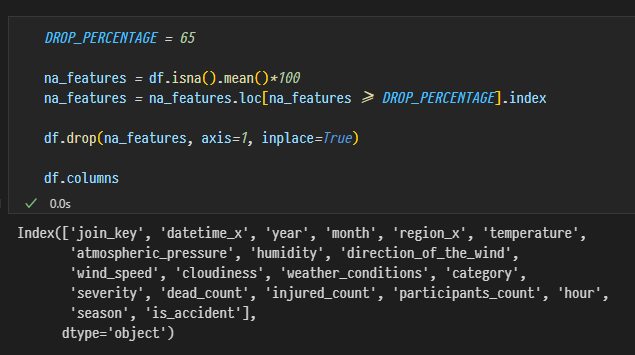


Рис. 10

Признаки, в который процент пропусков больше 65, что является большим значением, были удалены. Затем выделил, что осталось.



Рис. 11

Из оставшихся столбцов выделил “join\_key” (ключ по которому связывали изначальные данные), “datetime\_x” (время записи), “year” (год записи), “month” (месяц записи), “region\_x” (регион, где была создана запись), а также все, напрямую связанные с ДТП признаки, поскольку они были объединены в целевой признак “is\_accident” (будет ли ДТП).

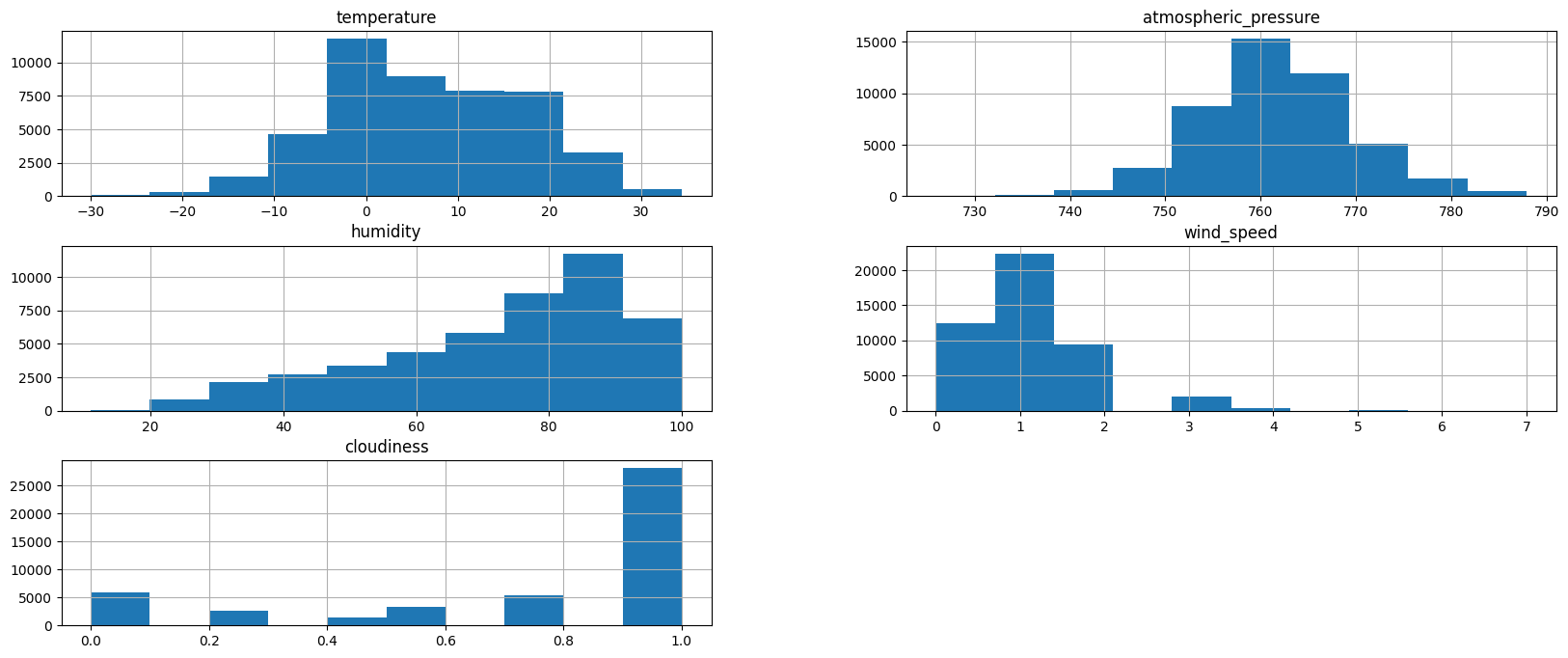


Рис. 12

Обратим внимание на имеющиеся “хвосты” графиков и в некоторых местах резкие перепады значения. Это может отказаться аномалией.

Дополнительно было проведено выявление выбросов с помощью метода трех сигм. Основная идея метода заключается в том, что 99.7% всех измерений должны находиться в пределах отклонения в три стандартных отклонения от среднего значения. Для этого создал функцию “is\_out\_std”, внутри неё установил лимит для определения аномалий в 3 стандартных отклонения от среднего значения, определил верхнюю и нижнюю границы, в которых данные считаются нормальными. Затем добавил в каждой строке, является ли запись аномалией, 0 – нет, 1 – да. Однако во время построения модели оказалось, что удаление выбросов плохо сказывается на способности прогнозирования.

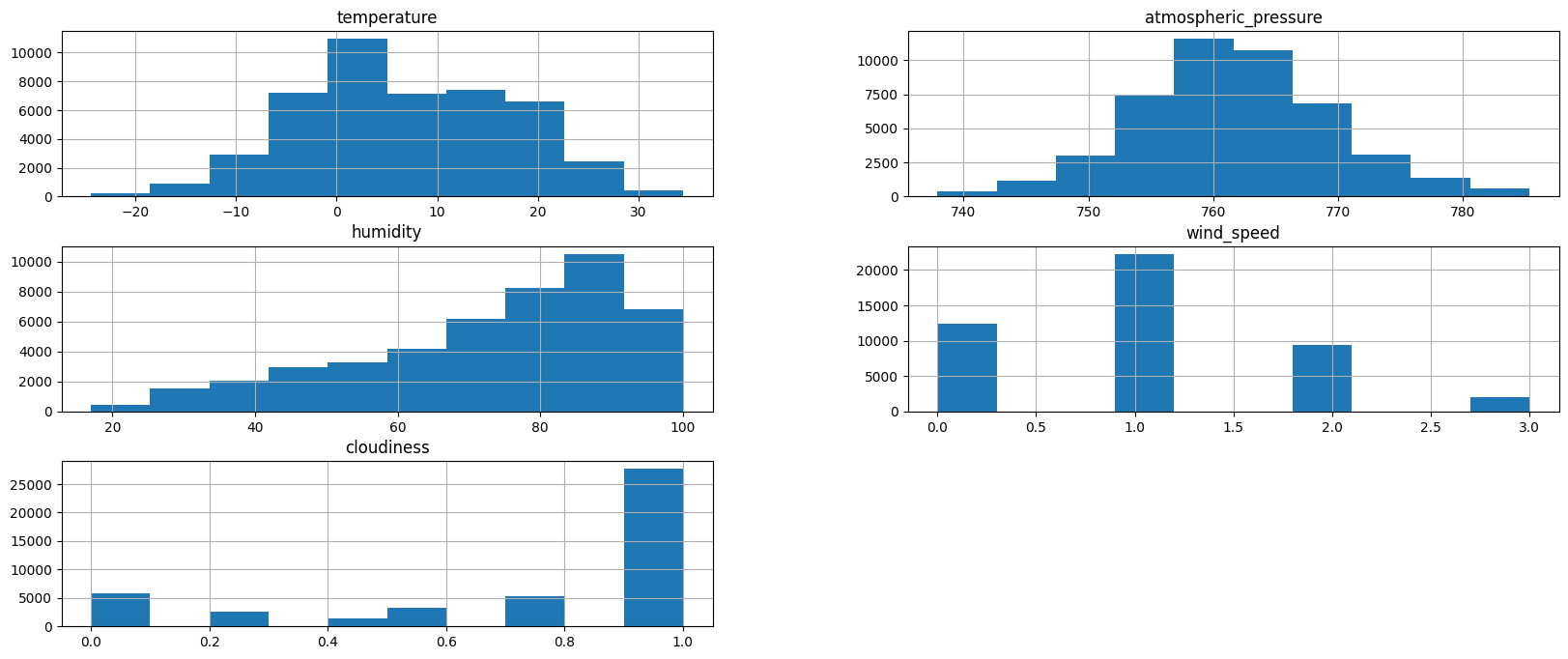


Рис. 13

Результат после удаления выбросов, графики немного “сгладились”.

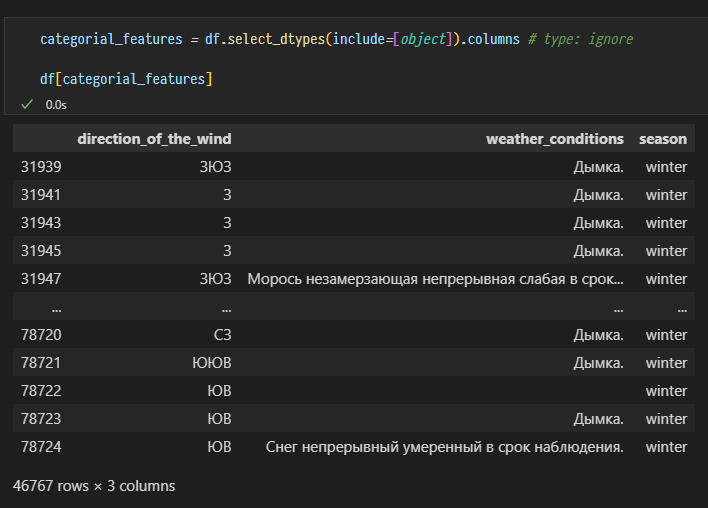


Рис. 14

Оставшиеся категориальные признаки. Столбцы “direction\_of\_the\_wind” (направление ветра) и “weather\_conditions” (погодные условия) удалим, поскольку они плохо влияют на прогнозы модели, а для “season” (сезон) проведем енкодинг c помощью быстрого кодирования OHE.

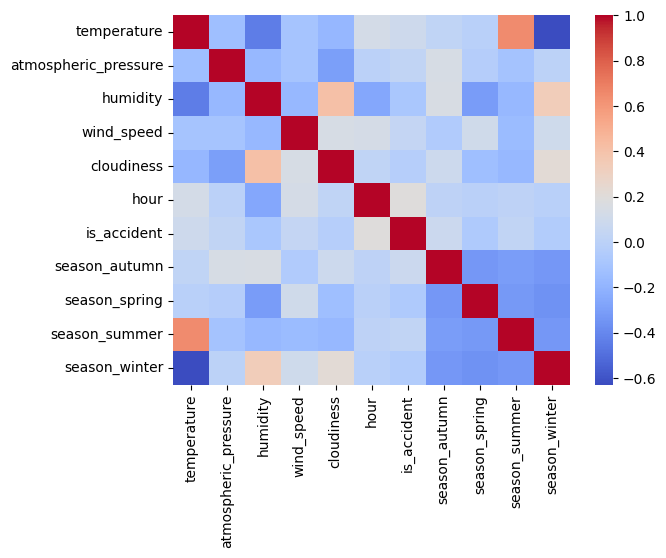


Рис. 15

Итоговая корреляционная матрица. Оставшиеся признаки температура, атмосферное давление, влажность, скорость ветра, облачность, время суток (час), и 4 сезона: зима, весна, лето и осень. Исходя из этих параметров, модель будет строить прогноз целевой переменной “is\_accident” (будет/не будет ДТП).

# Основные шаги выполнения. Разделение на выборки

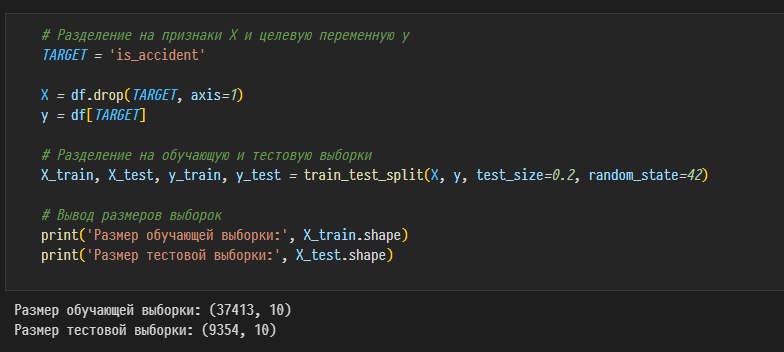


Рис. 16

Разделение на обучающею выборку в соотношение 80% и тестовую выборку 20% с параметром random\_state равным 42, он позволяет задать начальное состояние для генерации случайных чисел при разделении выборки.

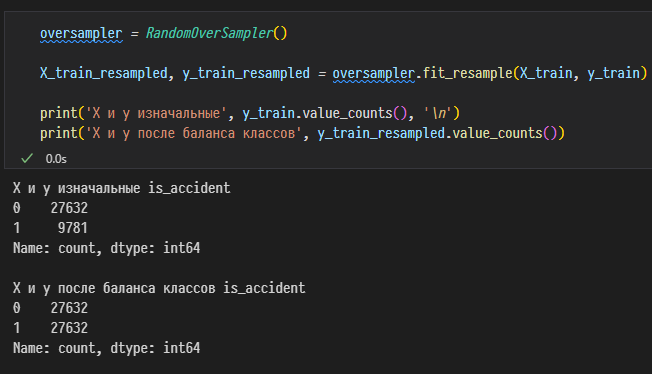


Рис. 17

Проведение баланса классов для тренировочной выборки.

# Основные шаги выполнения. Построение модели

Для построения базовой модели я выбрал TPOT. Это автоматизированный инструмент машинного обучения, предназначенный для автоматической оптимизации пайплайнов машинного обучения. Он использует генетический алгоритм для оптимизации множества моделей машинного обучения и их параметров, чтобы создать пайплайн, который лучше всего подходит для конкретной задачи. Также, чтобы подобрать наиболее оптимальные и подходящие гиперпараметры, я использовал Optuna.

Generations отвечает за количество поколений эволюционного алгоритма, где каждое поколение – новая финальная модель, а population\_size определяет количество моделей-кандидатов в одном поколении.

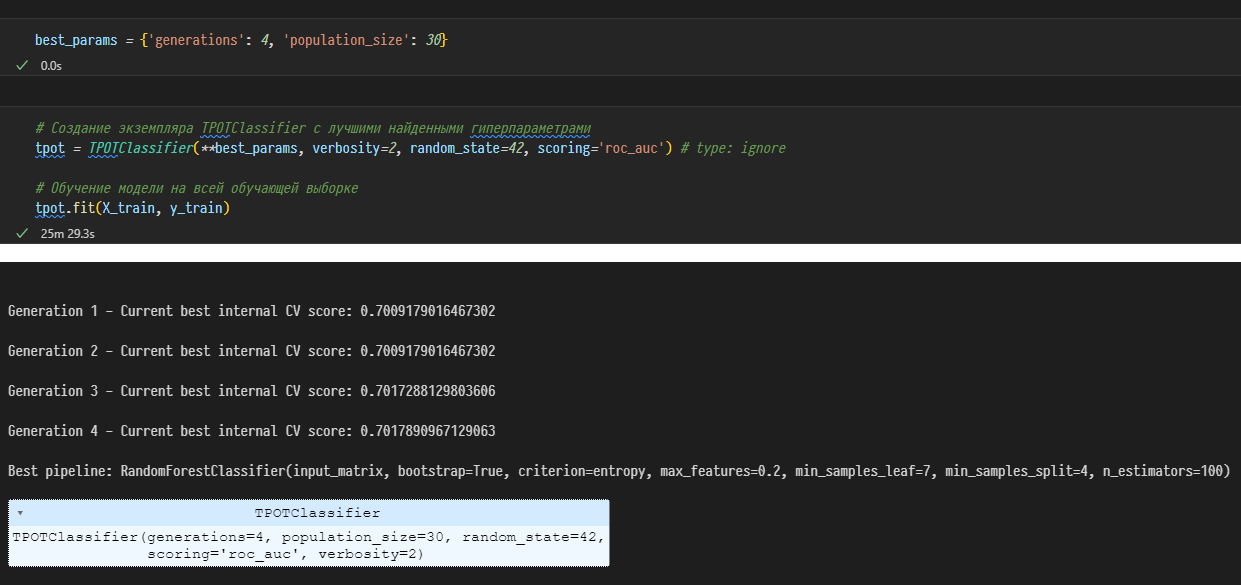


Рис. 18

Результат автоматизированной модели TPOT. Лучшим пайплайном был выбран RandomForestClassifier.

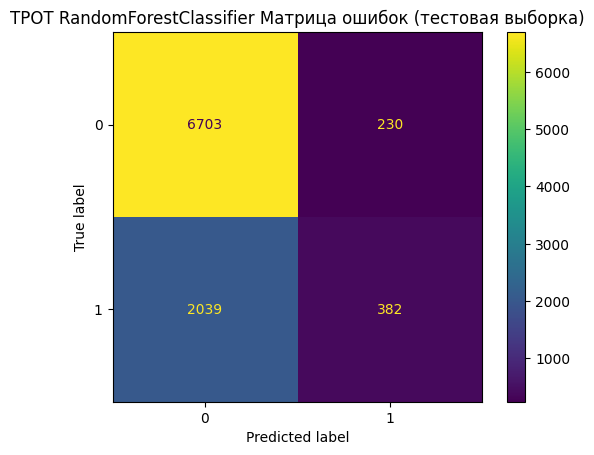


Рис. 19

Заметим, что данная модель плохо справляется с прогнозом случая, когда произошло/произойдет ДТП.

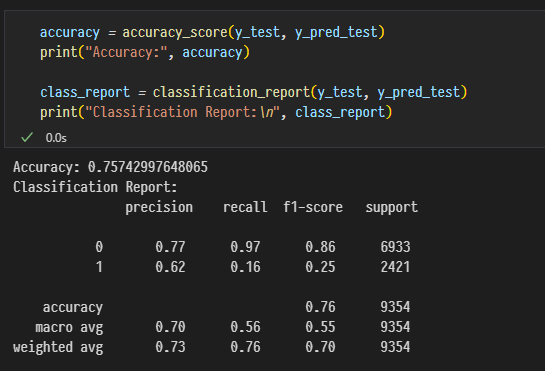


Рис. 20

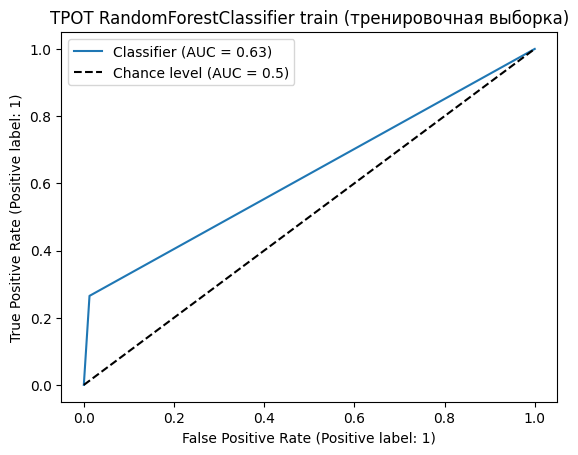


Рис. 21

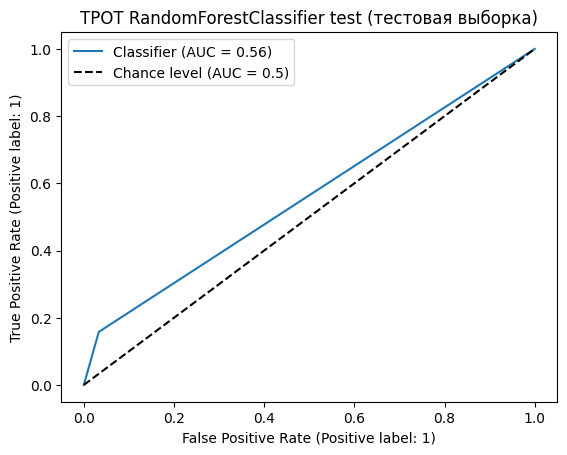


Рис. 22

# Основные шаги выполнения. Лучшая модель

На этом этапе я построил собственные модели машинного обучения с учетом тестовой выборки, на котором применен баланс классов, и также подбора гиперпараметров с помощью Optuna. Поставлена цель превзойти результат базовой автоматизированной модели по основным метрикам, например, как AUC ROC кривая. CatBoost и DecisionTree (дерево решений) выдали неплохие результаты, однако недостаточные для полноценного использования. Собственный RandomForestClassifier с подбором гиперпараметров оказался лучше среди всех моделей, в том числе, автоматизированного TPOT. Качество прогноза, исходя из метрик, стало лучше, также переобучения и недообучения не выявлено.

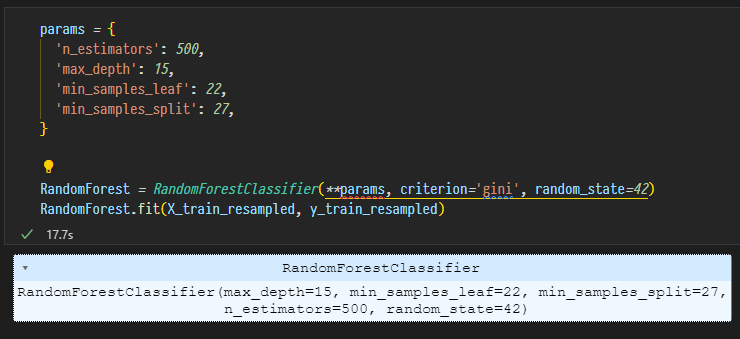


Рис. 23

Моя модель случайного леса с подобранными параметрами: количество деревьев – 500, максимальная глубина дерева – 15, минимальное количество наблюдений в листьях дерева – 22, а максимальное, для разделения внутреннего узла, - 27, а критерий выбран gini (Джинни), который показывает насколько чистыми (однородными) являются классы в узле. Энтропия показала результаты хуже.

Лучшей собственной и автоматизированной моделью из TPOT одновременно оказался RandomForestClassifier. Отличиями стали гиперпараметры, метрики и выдаваемые результаты. Моя модель обнаруживает случая с возможным ДТП чаще, чем предложенная библиотекой.

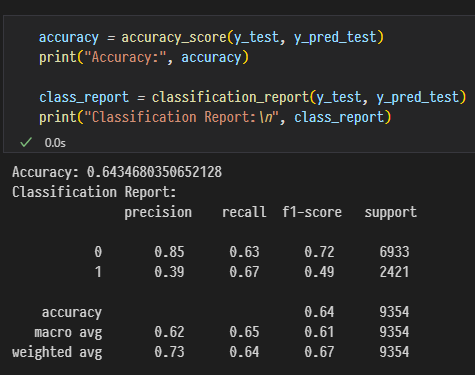


Рис. 24

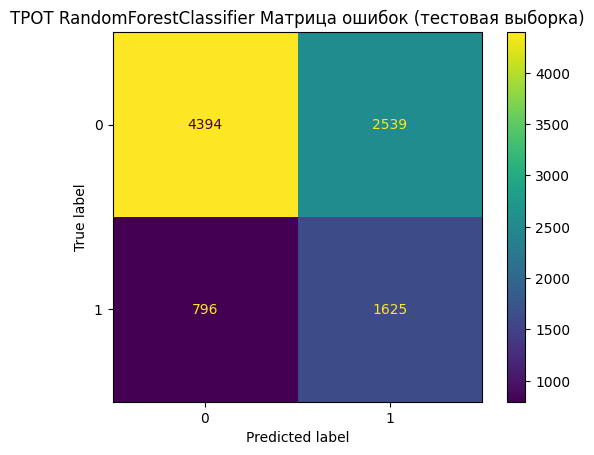


Рис. 25

Относительно автоматизированной модели, выявление случаев с ДТП улучшилось.

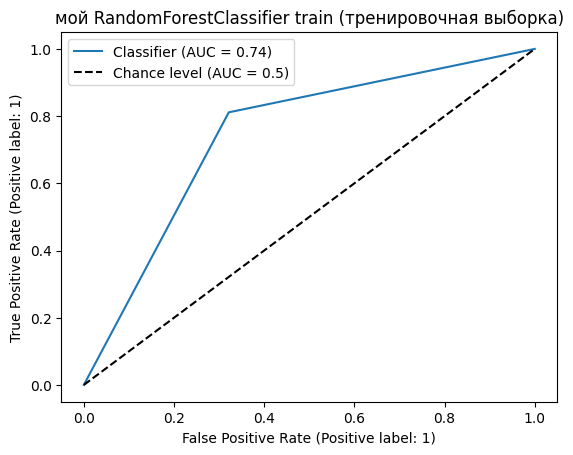


Рис. 26

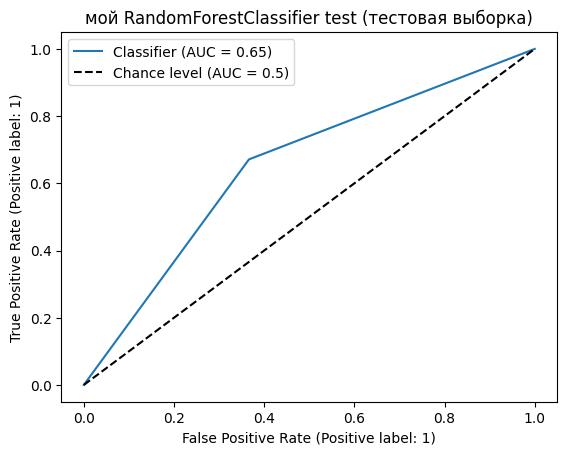


Рис. 27

# Основные шаги выполнения. Интерпретация лучшей модели

Для лучшей модели было принято решение рассмотреть наиболее важных факторов, и то, как они влияют на прогноз.

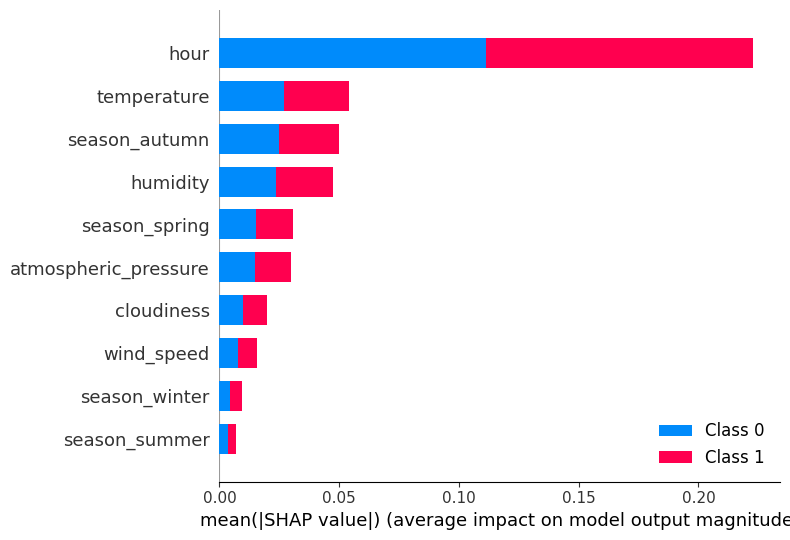


Рис. 28

Самым важным фактором оказалось время суток, затем температура и наличие сезона осень.

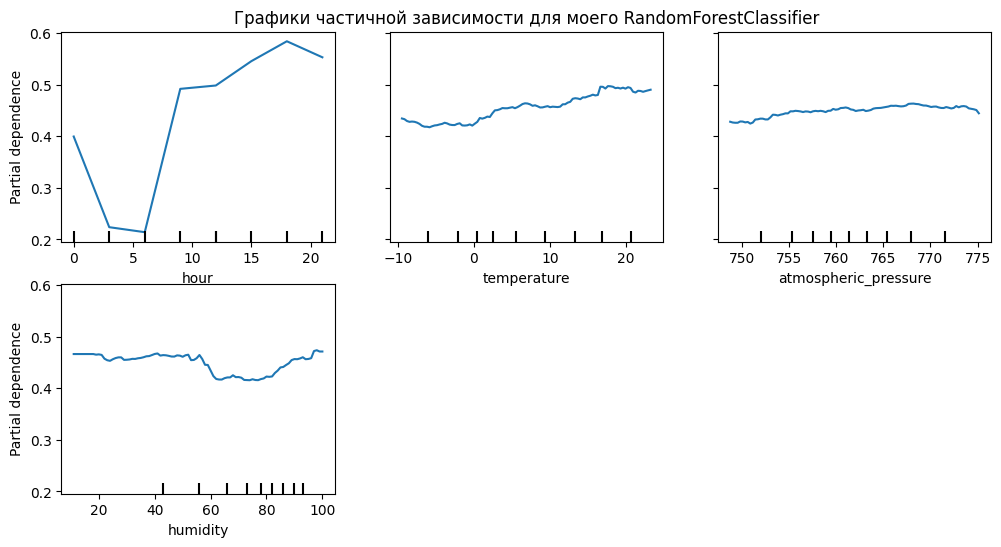


Рис. 29

Исходя из графиков частичной зависимости, видно, что в ночное время возможность случая ДТП резкое падает, а в остальное время растет. Также при повышенной влажности инциденты происходят реже.

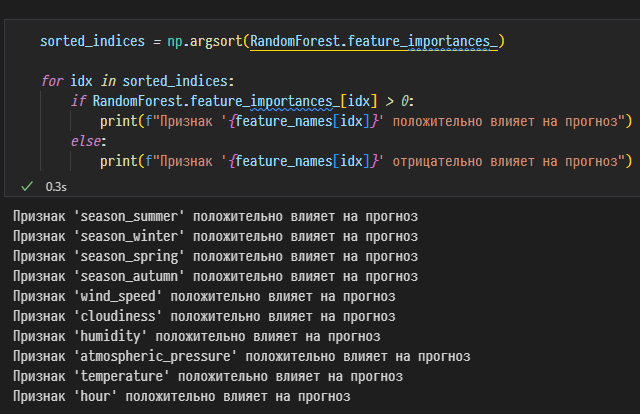


Рис. 30



Рис. 31

# Разработка приложения для тестирования

Дополнительно было решено разработать backend приложение, для создания прогноза удаленно на сервере по запросу на ендпоинт, и frontend веб-приложения для удобного пользователю взаимодействия. В качестве интерфейса для связи между клиентом и сервером был выбран протокол REST API.

Backend написан на flask и обрабатывает POST запрос на один ендпоинт “/forecast” и ожидает тело запросов в формате JSON: массив объектов с полями temperature, atmospheric\_pressure, humidity, wind\_speed, cloudiness, hour, season\_autumn, season\_spring, season\_summer и season\_winter.

Frontend написан на javascript с надстройкой typescript, когда позволяет добавить строгую типизацию в проект, и на библиотеке react, которая позволяет создавать реактивные одностраничные приложения.

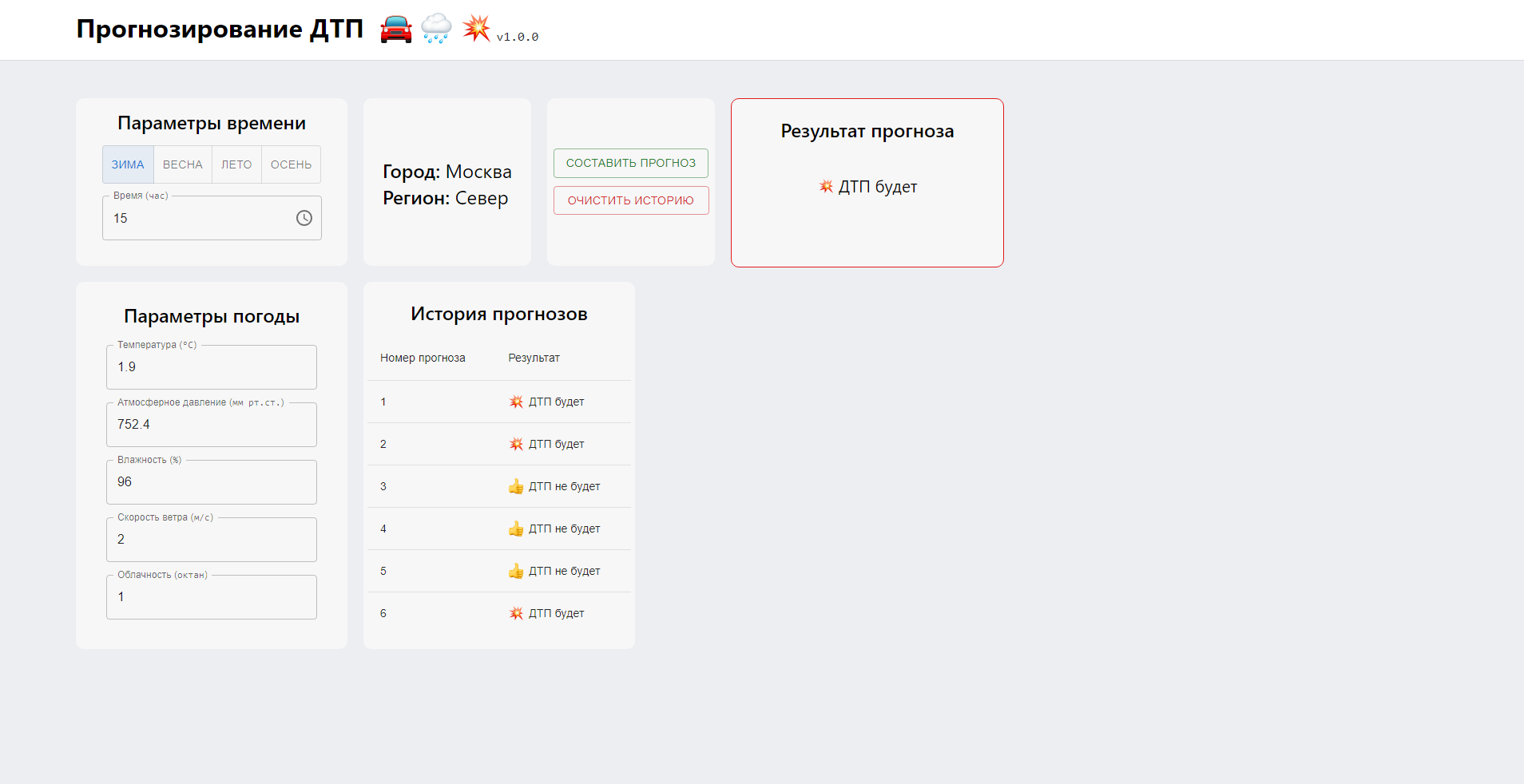


Рис. 32

Интерфейс веб-приложения, в котором можно задать параметры для составления прогноза ДТП на основе собственной модели.

# Заключение

Были построены модели машинного обучения, которые на основе погодных условий составляют прогноз величины “будет или не будет ДТП”. С помощью подбора параметров, удалось создать собственную модель, которая стала самой лучшей по метрикам: матрица ошибок и AUC ROC кривая, и хорошо справляется с задачей бинарной классификации. С помощью SHAP и LIME выделил признаки, которые играют наибольшую роль при создании прогноза. Также разработано полноценное приложение, с помощью которого можно без проблем получить предсказание лучшей модели, исходя из заданных параметров погоды.

Стоит отметить, что предсказание возможности ДТП в реальной жизни зависит от множества факторов, включая поведение водителей, дорожные условия и другие важные аспекты, которые можно учитывать при сборе данных для последующего моделирования для того, чтобы улучшить качество прогнозов

# Список использованных источников и программ

1. Библиотеки для проведения анализа: numpy, pandas, matplotlib, seaborn.
2. Библиотеки для построения моделей: sklearn, catboost, tpot, joblib, imblearn, shap, optuna, graphviz, lime.
3. Библиотеки для разработки backend приложения: flask, pandas, joblib.
4. Библиотеки для разработки frontend веб-приложения: react, react-router-dom, axios, styled-components, MUI.

# Приложение

1. Файл с кодом (анализ данных и построение модели) – <https://github.com/jsinkx/traffic-accident-ml>
2. Файл с кодом (backend) – <https://github.com/jsinkx/traffic-accident-backend>
3. Файл с кодом (frontend) – <https://github.com/jsinkx/traffic-accident-frontend>