**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](#_Toc155808772)

[Введение 5](#_Toc155808773)

[Описание задачи 6](#_Toc155808774)

[Основные шаги выполнения. EDA 7](#_Toc155808775)

[Основные шаги выполнения. Подготовка данных 10](#_Toc155808776)

[Основные шаги выполнения. Разделение на выборки 16](#_Toc155808777)

[Основные шаги выполнения. Построение модели 17](#_Toc155808778)

[Основные шаги выполнения. Лучшая модель 20](#_Toc155808779)

[Основные шаги выполнения. Интерпретация лучшей модели 23](#_Toc155808780)

[Разработка приложения для тестирования 25](#_Toc155808781)

[Заключение 26](#_Toc155808782)

[Список использованных источников и программ 27](#_Toc155808783)

[Приложение 28](#_Toc155808784)

# Основные термины и сокращения

ДТП – дорожно-транспортное происшествие.

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

Мультиколлинеарность – это явление, при котором независимые переменные или состоящие из других сильно коррелируют между собой. Из-за этого способность модели к предсказанию может ухудшиться и усложнить ее интепретацию.

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

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

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.

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

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

AUC ROC – метрика, позволяющая оценить качество модели.

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

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

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

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

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

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

REST – (representational state transfer) протокол передачи данных.

# 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 события “будет или не будет ДТП”, учитывая погодные условия.

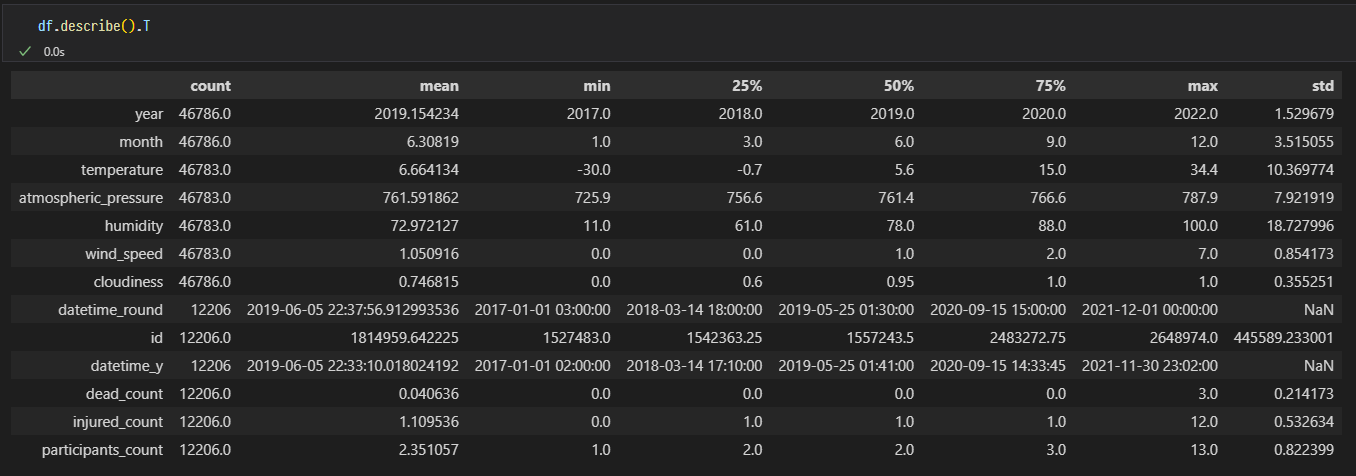
# Описание задачи

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

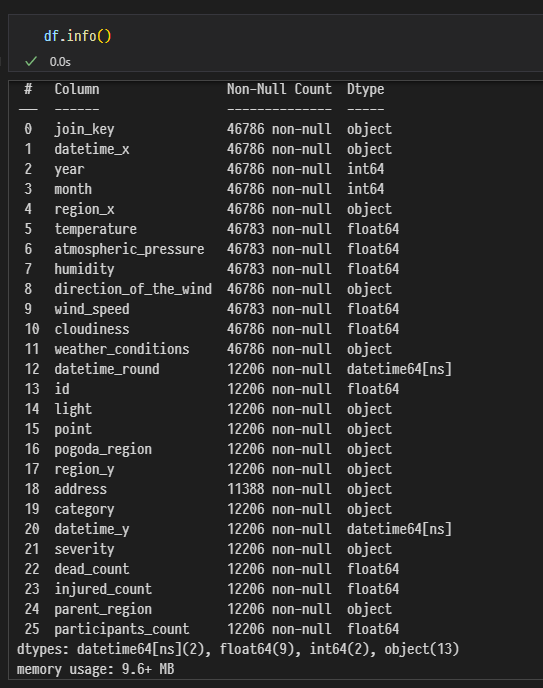
# Основные шаги выполнения. EDA

В данной курсовой работе были использованы данные в формате .xlsx с информацией по ДТП – “DataDTPкор.xlsx” и по погоде – “ Pogoda.xlsx”. В качестве языка программирования выбран Python 3.10 и среда разработки Jupyter Notebook. После загрузки данные были объединены по признаку “join\_key”. Затем нами были отобраны регионы, относящиеся к северной части России. Объем данных составил 46786 строк и 26 столбцов, представляющие 12 признаков (Рис.1).

*Рис.1. Статистическая информация о датасете*



*Рис.2. Сведения о датасете*



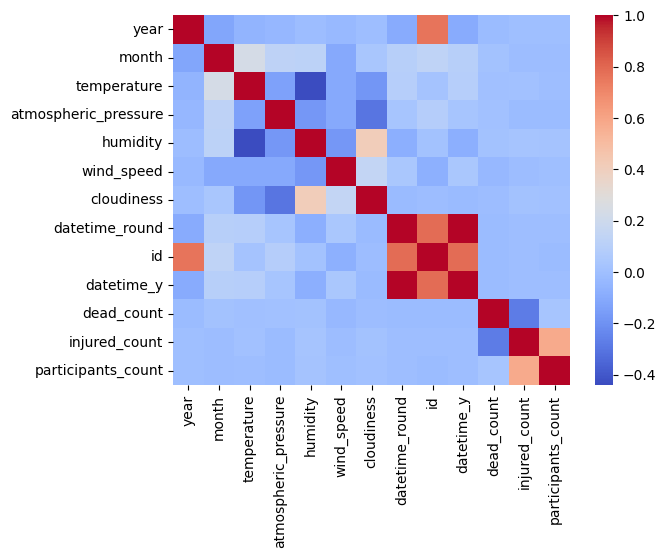
Основными действиями в рамках этапа EDA стали удаление дубликатов, приведение корректного типа в некоторых столбцах, заполнение и удаление пропусков, удаление ненужных признаков.

*Рис.3. Работа со столбцами «hour» и «season»*



Было принято решение разбить столбец с датой записи на “hour” (в какой час была создана запись) и “season” (в какой сезон создана запись: зима, весна, лето или осень) (Рис.3).

*Рис.4. Корреляционная матрица признаков*

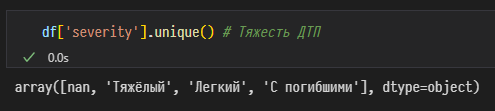


Построив корреляционную матрицу (Рис.4), заметим, что есть признаки, которые имеют мультиколлинеарность. Потенциально это способно оказать негативное влияние на предсказательную способность нашей модели, поэтому в следующих шагах необходимо устранить этот недостаток.

# Основные шаги выполнения. Подготовка данных

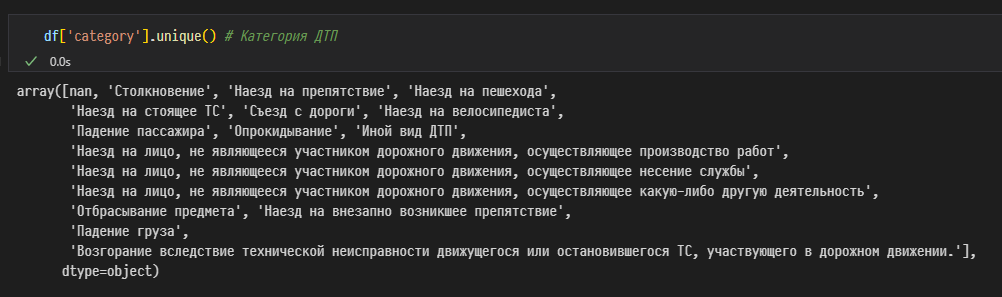
Особое внимание было уделено признакам, которые напрямую связаны с исходом ДТП. С целью возможности дальнейшего использования в данные был внесен ряд изменений.

*Рис.5. Работа с признаком «severity»*



Признак “severity” описывает степень тяжести происшествия (Рис.5). Его пропуск (nan) означает, что ДТП не было.

*Рис.6. Работа с признаком «category»*

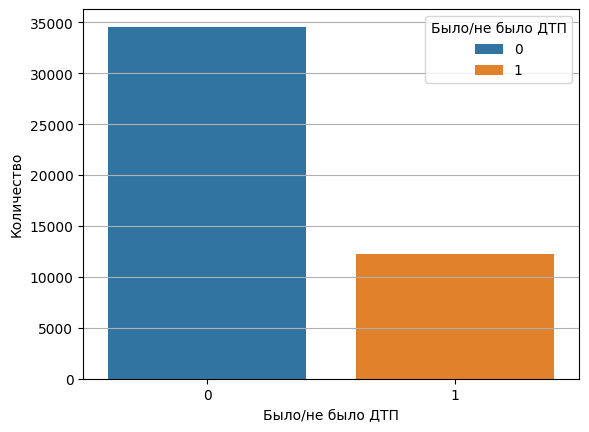


Признак “category” описывает категорию происшествия (Рис.6), например, «падение пассажира», «столкновение» или другой тип. Его пропуск означает, что ДТП не произошло.

Исходя из того, что значение nan в этих признаках означает отсутствие ДТП, было принято решение о замене значений, а именно 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 участников) пропуски были заменены на 0, что означает отсутствие записи о происшествии.

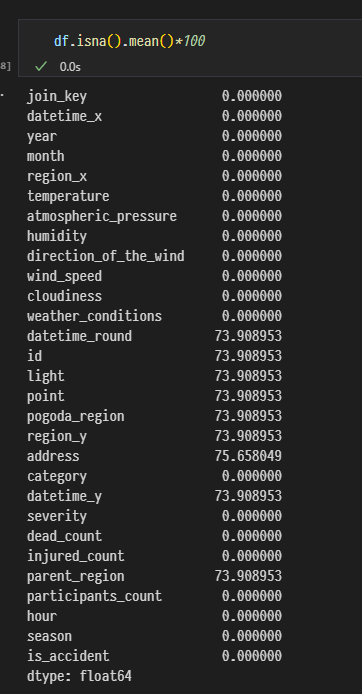
Было принято решение создать столбец “is\_accident”, который отображает наличие происшествия, если значение 1 или его отсутствие со значением 0. Его формирование происходило исходя из признаков, прямо отображающих наличие записи о ДТП – “dead\_counts”, “injured\_counts”, “participants\_count”, “severity”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

*Рис.7. Столбчатая гистограмма, отражающая соотношение событий ДТП*



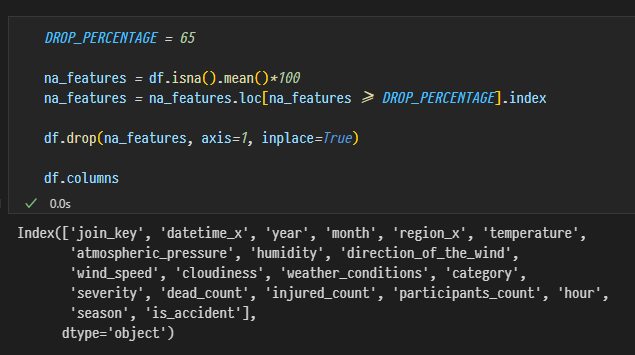
Заметим, что количество записей отсутствия ДТП и наличия сильно отличается, что видно из построенной столбчатой гистограммы (Рис.7). Это может негативно сказаться на обучении модели. Поэтому на этапе разделения данных на выборки нужно будет провести балансировку классов, то есть примерно уравнять их количество.

*Рис.8. Процент пропущенных значений в признаках*



Благодаря (Рис. 8) мы можем оценить процент пропусков в каждом столбце. Заметим, что в основном процент достаточно большой, а также некоторые признаки никак не понадобятся при обучении модели.

*Рис.9. Отбор признаков с процентом пропусков больше 65%*



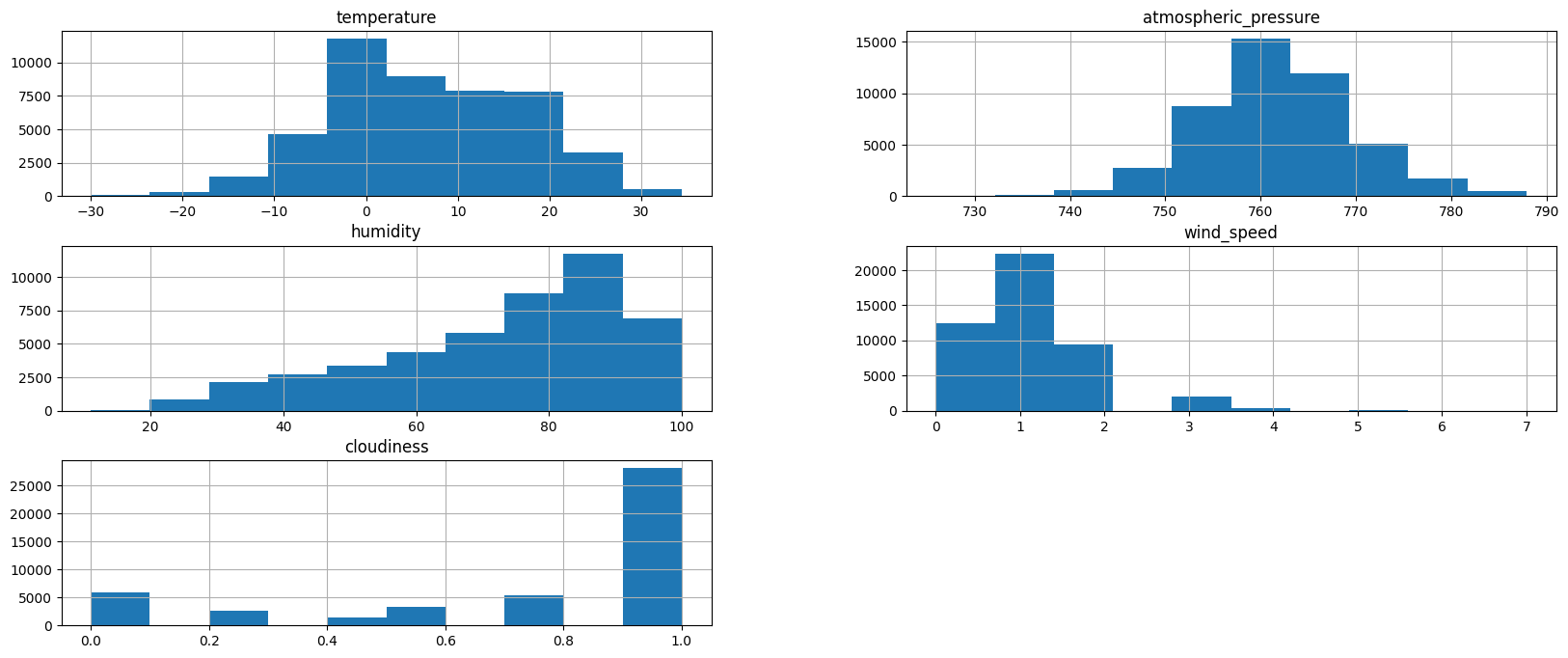
Признаки, в который процент пропусков больше 65, что является большим значением, были удалены (Рис.9). После удаления в датасете оставлено 20 признаков: join\_key, datetime\_x, year, month, region\_x, temperature, atmpspheric\_pressure, humidity, direction\_of\_the\_wind, wind\_speed, cloudiness, weather\_conditions, category, severity, dead\_count, injured\_count, participants\_count, hour, season и is\_accident.

*Рис.10. Отбор неиспользуемых признаков*



Из оставшихся столбцов были удалены “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 признаки, напрямую связанные с ДТП, поскольку они были объединены в целевой признак “is\_accident” (будет ли ДТП) (Рис.10).

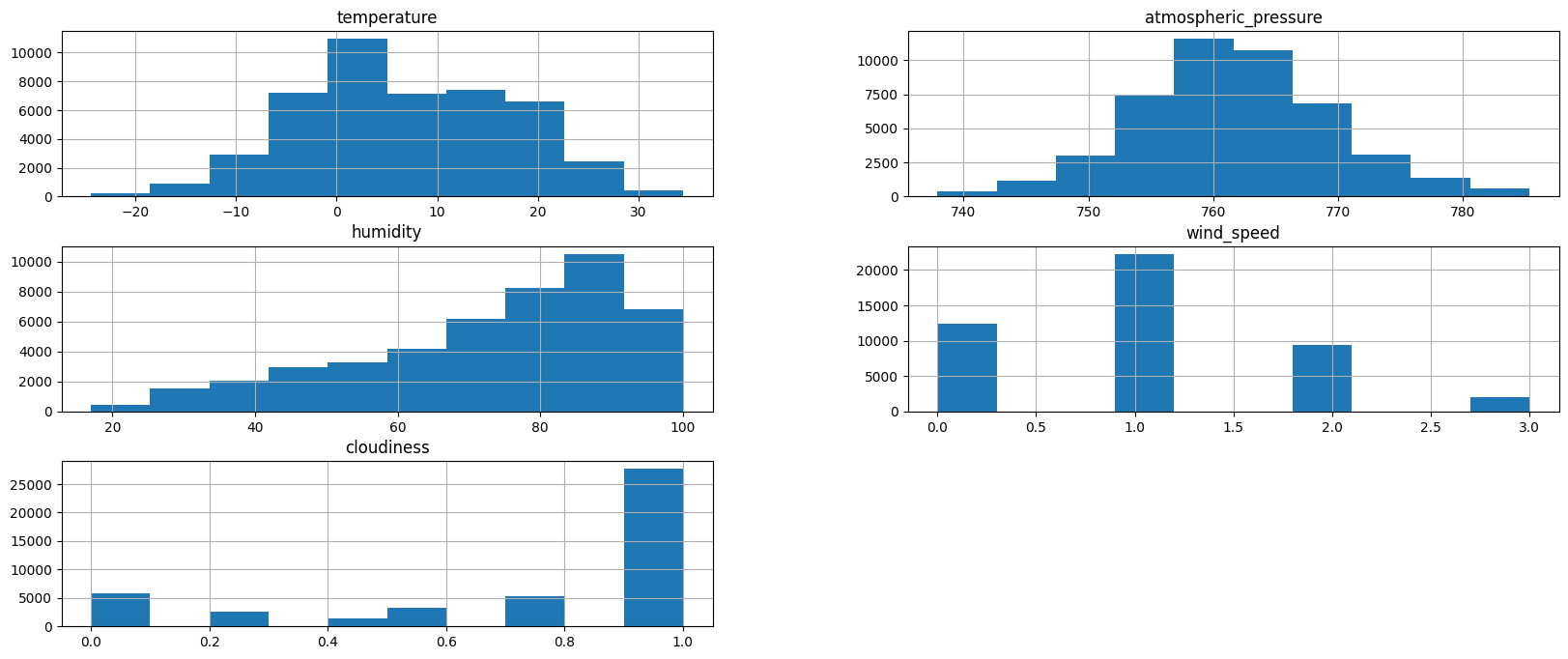
*Рис.11. Распределение значений в признаках да удаления выбросов*



Обратим внимание на имеющиеся “хвосты” графиков распределения значений в признаках (Рис.11), а также резкие перепады значений в некоторых участках. Отметим, что это может отказаться аномалией.

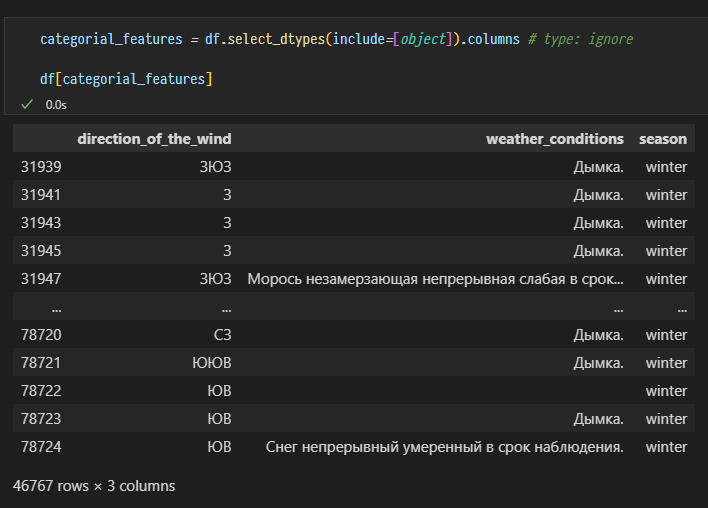
Дополнительно было проведено выявление выбросов с помощью метода трех сигм. Основная идея метода заключается в том, что 99.7% всех измерений должны находиться в пределах отклонения в три стандартных отклонения от среднего значения. Для этого была создана функция “is\_out\_std”, внутри неё установлен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

*Рис.12. Распределение значений в признаках после удаления выбросов на копии*



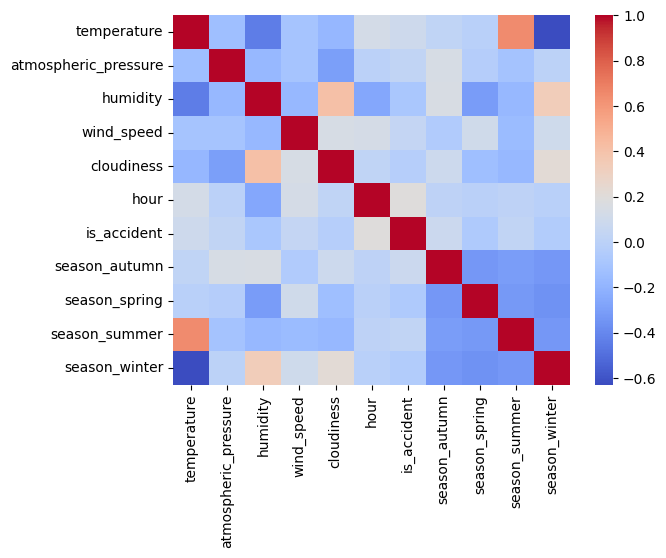
Теперь рассмотрим получившийся результат после удаления выбросов (Рис.12). Мы видим, что графики немного сгладились, резких пиков, наблюдаемые нами ранее, исчезли.

*Рис.13. Оставшиеся категориальные признаки*



Теперь рассмотрим оставшиеся категориальные признаки (Рис.13). 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.

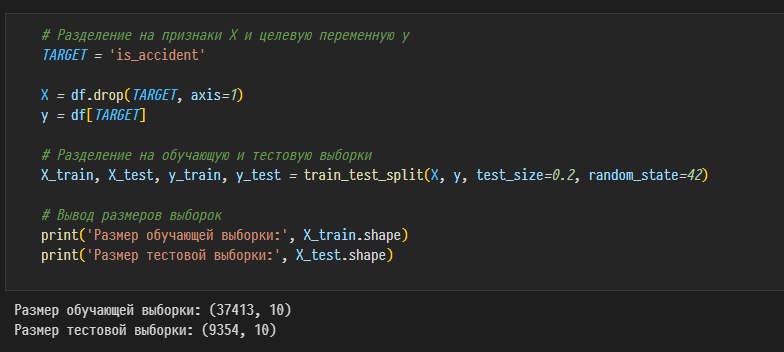
*Рис.14. Итоговая корреляционная матрица*



Рассмотрим итоговую корреляционную матрицу (Рис.14). Оставшиеся признаки включают в себя следующую информацию: 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).

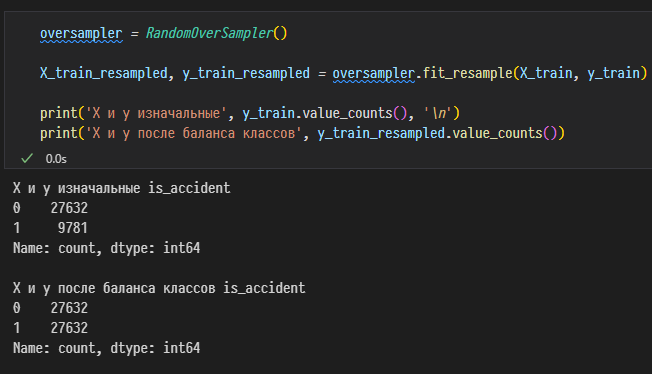
# Основные шаги выполнения. Разделение на выборки

*Рис.15. Разделение на выборки*



Разделим данные (Рис.15) на обучающую 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.16. Проведение балансировки классов*



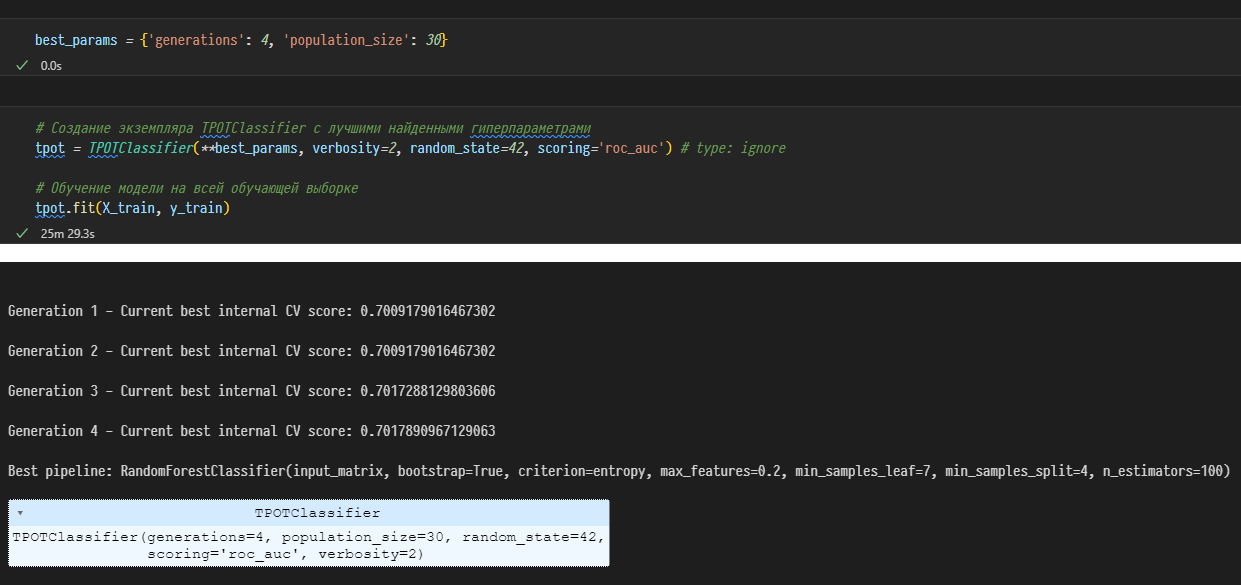
Проведем балансировку классов для тренировочной выборки (Рис.16).

# Основные шаги выполнения. Построение модели

Для построения базовой модели был выбран 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 оптимальные и подходящие гиперпараметры, был использован фреймворк Optuna.

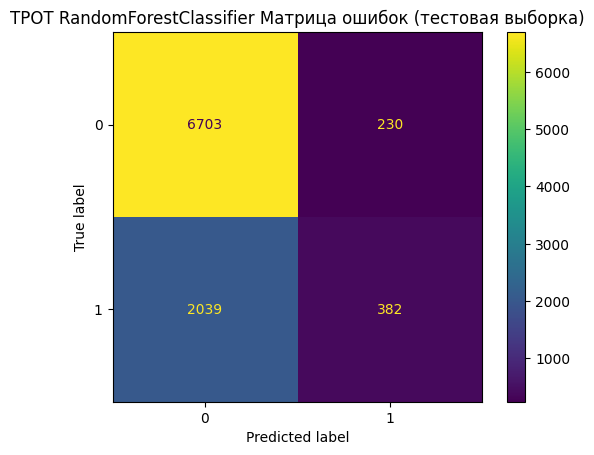
Generations отвечает за количество поколений эволюционного алгоритма, где каждое поколение – новая финальная модель, а population\_size определяет количество моделей-кандидатов в одном поколении.

*Рис.17. Результат автоматизированной модели ТРОТ*



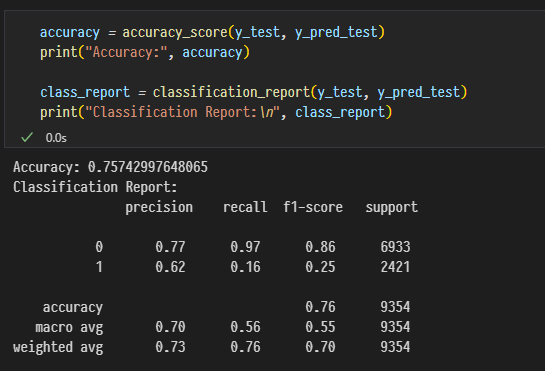
Рассмотрим результат автоматизированной модели TPOT (Рис.17). Лучшим пайплайном был выбран RandomForestClassifier.

*Рис.18. Матрица ошибок АМ TPOT (тестовая выборка)*

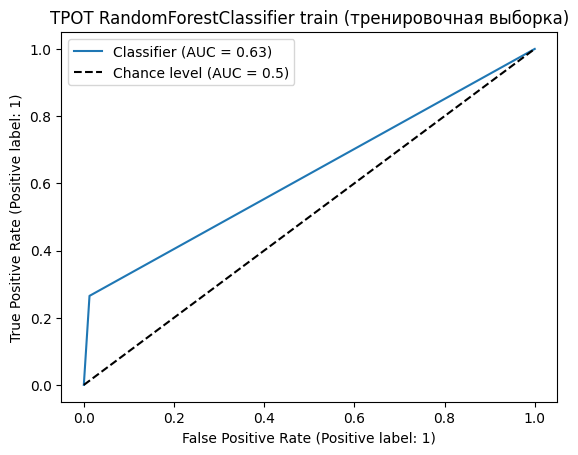


Заметим, что данная модель плохо справляется с прогнозом случая, когда произошло или произойдет ДТП (Рис.18).

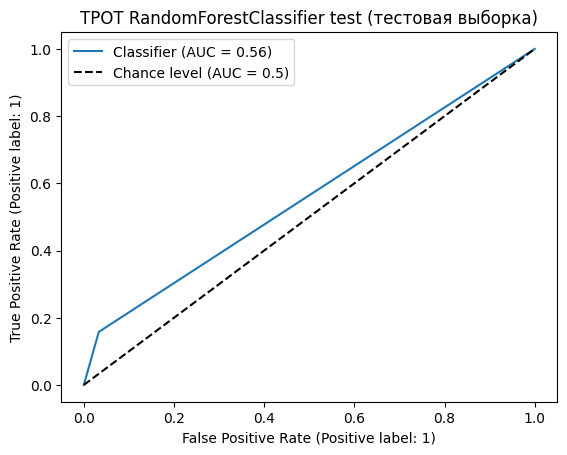
*Рис.19. метрики АМ ТРОТ*



*Рис.20. АМ TPOT AUC ROC (тренировочная выборка)*



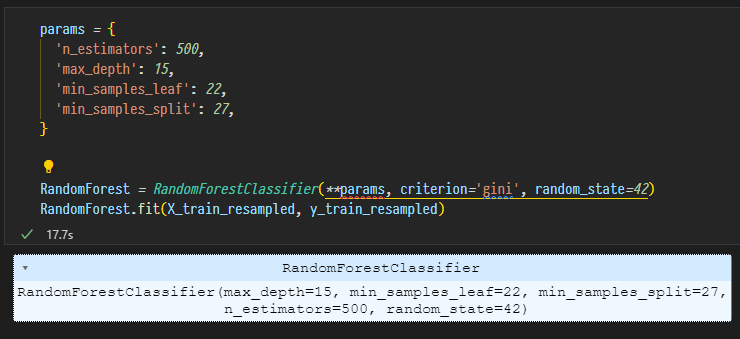
*Рис.21. АМ TPOT AUC ROC (тестовая выборка)*



# Основные шаги выполнения. Лучшая модель

На этом этапе были построены собственные модели машинного обучения с учетом тестовой выборки, на которых был применен баланс классов, а также подбор гиперпараметров с помощью Optuna. Кроме того, была поставлена цель превзойти результат базовой автоматизированной модели по основным метрикам, например, такой как AUC ROC кривая. CatBoost и DecisionTree (дерево решений) показали удовлетворительные результаты, однако недостаточные для полноценного использования. Собственный RandomForestClassifier с подбором гиперпараметров оказался лучшим среди всех моделей, в том числе и автоматизированного TPOT. Качество прогноза, исходя из метрик, стало лучше, также переобучения и недообучения не выявлено.

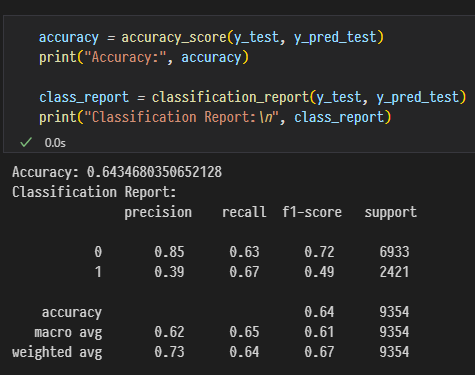
*Рис.22. Собственная модель случайного леса*



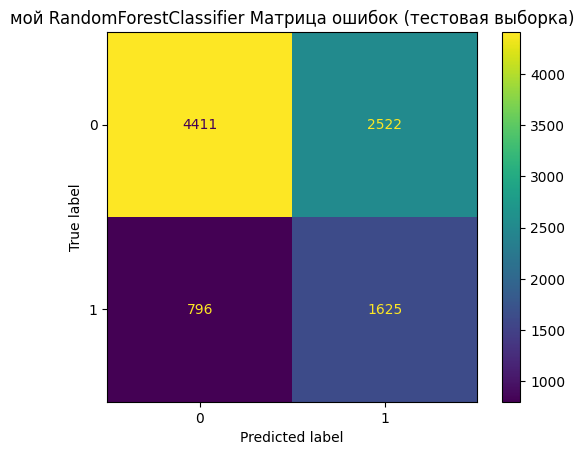
Наша модель случайного леса с подобранными параметрами состоит из следующих параметров (Рис.22): количество деревьев – 500, максимальная глубина дерева – 15, минимальное количество наблюдений в листьях дерева – 22, а максимальное, для разделения внутреннего узла, - 27. Критерий выбран gini (Джинни), 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.23. Метрики собственной модели*

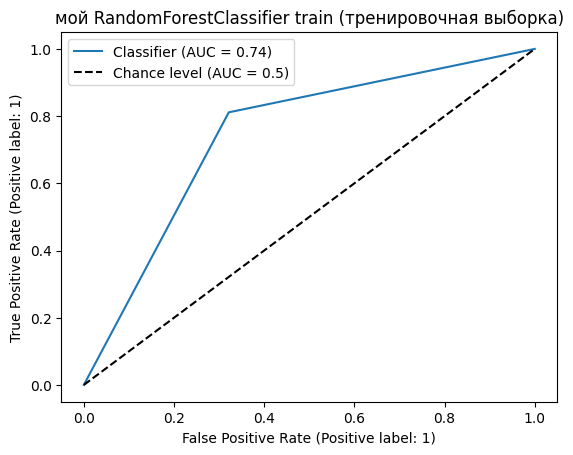


*Рис.24. Матрица ошибок собственной модели (тестовая выборка)*

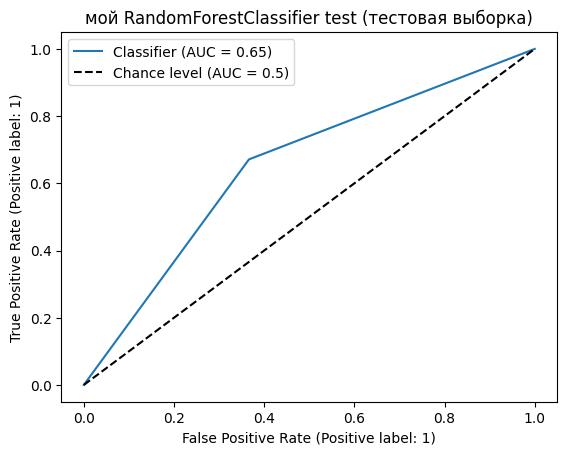


Рассмотрим матрицу ошибок для тестовой выборки. Можно заметить, что относительно автоматизированной модели выявление случаев с ДТП улучшилось (Рис.24).

*Рис.25. Собственная модель случайного леса AUC ROC (тренировочная выборка)*



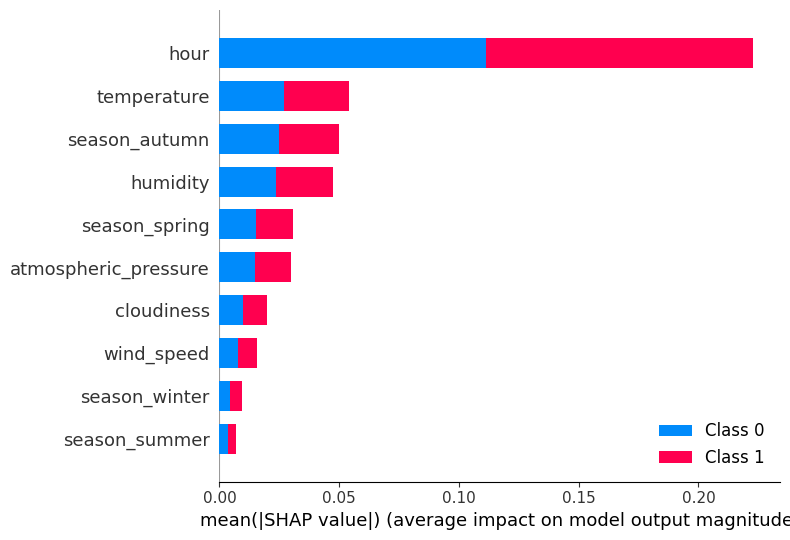
*Рис.26. Собственная модель случайного леса AUC ROC (тестовая выборка)*



# Основные шаги выполнения. Интерпретация лучшей модели

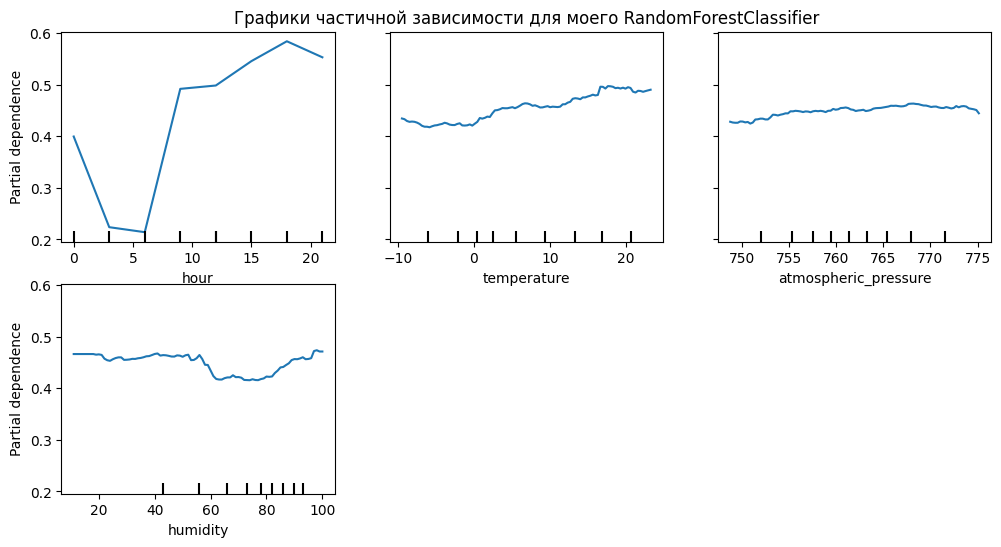
Для лучшей модели, моего случайного леса, было принято решение рассмотреть ряд наиболее важных факторов и то, как они влияют на прогноз (Рис.27).

*Рис.27. Самые важные факторы и их влияние на прогноз в собственной модели*



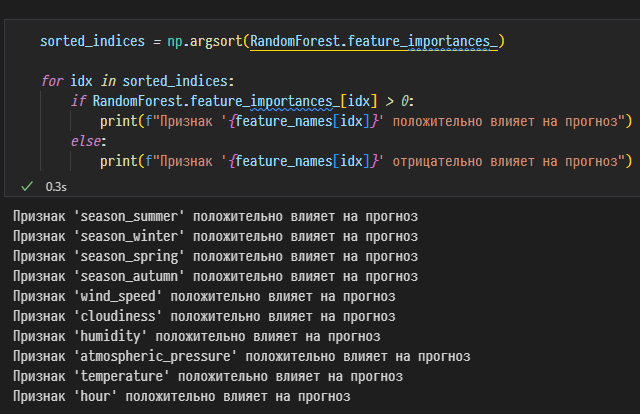
Самым важным фактором оказалось время суток, затем температура и наличие сезона «осень».

*Рис.28. Графики частичной зависимости для собственной модели*



Исходя из графиков частичной зависимости (Рис.28), мы можем увидеть, что в ночное время возможность случая ДТП резкое падает, а в остальное время растет. Также при повышенной влажности инциденты происходят реже.

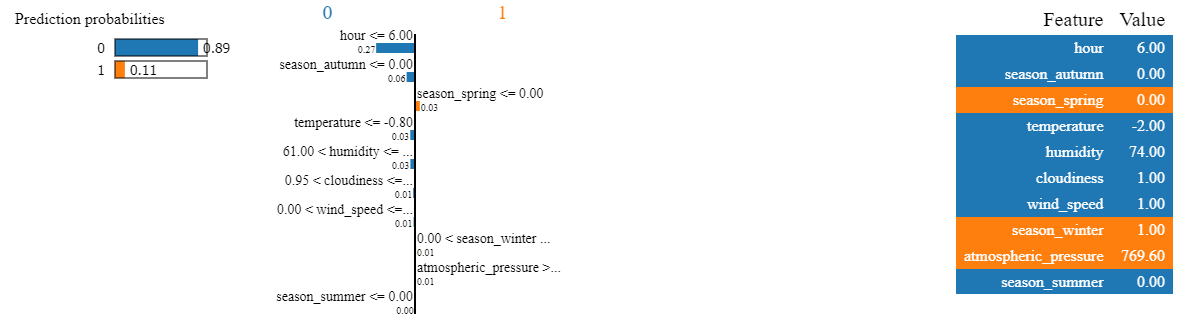
*Рис.29. Оценка влияния признаков на прогноз*



*Рис.30. Локальная интерпретация случая с ДТП для собственной модели*



*Рис.31. Локальная интерпретация случая без ДТП для собственной модели*



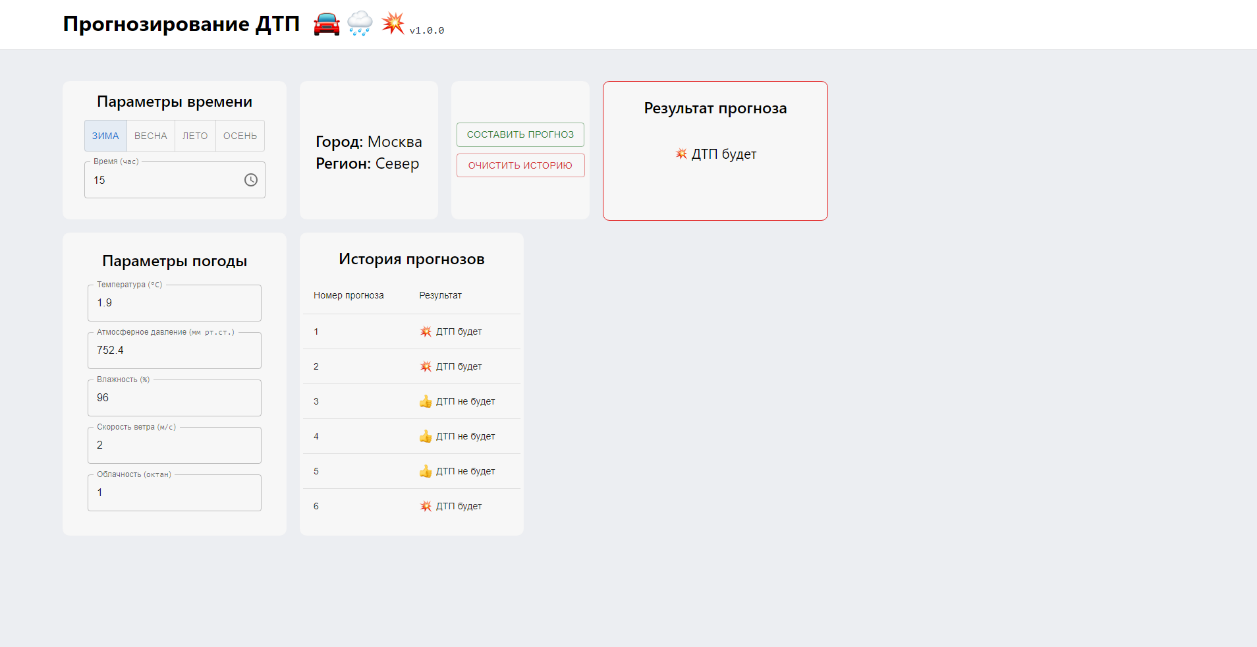
# Разработка приложения для тестирования

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. Результат веб-приложения*



На рисунке 31 можно увидеть интерфейс веб-приложения, в котором возможно задать параметры для составления прогноза ДТП на основе собственной модели.

# Заключение

В рамках проведенной работы были построены модели машинного обучения, которые на основе погодных условий составляют прогноз события “будет или не будет ДТП”. С помощью подбора параметров удалось создать собственную модель, которая стала самой лучшей по следующим 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. Оказалось, что вечернее время и сезон зимы ДТП может произойти с высокой вероятностью. Также разработано полноценное приложение, с помощью которого можно 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>