МОДЕЛЬ ПРЕДСКАЗАНИЯ ПРОФИЛЯ ВОДИТЕЛЯ НА ОСНОВЕ ДАННЫХ ТЕЛЕМАТИКИ

Оглавление

Стр.

Аннотация

Целью работы является решение комплексной задачи по созданию профиля водителя, а также решение важных смежных задач, таких как, идентификация водителей, поиск потенциально опасных водителей, а также поиск признаков, наиболее характеризующих поведение водителей за рулем.

Представленный в данном исследовании метод создания профиля водителей отличается от существующих подходов тем, что он не классифицирует водителей по определенным категориям, а выдает для каждого индивидуальную и репрезентативную метрику.

В результате работы:

1. Представлена модель, которая создает динамически обновляемый профиль водителя. Данная модель выдает метрику, характеризующую поведение водителя, что может послужить основой для решения широкого спектра задач.
2. Найдено оптимальное сочетание признаков, которое наиболее точно характеризует поведение водителя, а также были добавлены признаки понятные среднестатистическому водителю. Данное сочетание признаков позволит использовать только наиболее важные данные для обучения.
3. Реализован поиск потенциально опасных водителей с помощью поиска аномалий, что позволит проверять асессорам только действительно потенциально опасных водителей, а не всех.
4. Реализована идентификация водителей с точностью 98.8%, что позволяет наиболее точно идентифицировать водителей, противодействовать страховому мошенничеству и предложить персонализированный сервис.

Введение

С развитием технологий и ростом автомобильной промышленности всё больше информации накапливается о поведении водителей на дорогах. Анализ этой информации позволяет сделать выводы о привычках, стиле и качестве вождения. Особый интерес вызывает возможность создания моделей предсказания профиля водителя на основе данных телематики. Данные телематики позволяют получать большое множество данных, некоторые из них знакомы обычному пользователю, такие как геопозиция, скорость движения, ускорение, расход топлива. С другой стороны датчики в автомобиле позволяют считывать такие признаки, как наклон педалей, содержание воздуха в топливной смеси, положение дроссельной заслонки, крутящий момент и т.д. Таким образом данные телематики предоставляют обширное количество данных для построения моделей с использованием машинного обучения.

Профиль водителя - это совокупность характеристик, которые описывают поведение водителя на дороге. Он включает в себя такие параметры, как скорость, дистанция, время реакции, умение оценивать ситуацию и принимать решения, а также другие факторы, которые влияют на безопасность дорожного движения. Также штрафы на нарушение правил дорожного движения и дтп могут помочь характеризовать водителя и его стиль вождения.

При создании профиля водителя на основе данных телематики, можно разделить водителей на основные следующие категории:

1. Опасные водители - это водители, которые часто превышают скорость, превышают допустимые нормы по скорости и расстоянию, а также нарушают правила дорожного движения.
2. Спокойные водители - это те, кто соблюдает правила дорожного движения, не нарушает скоростной режим и дистанцию, а также не имеет большого количества штрафов.
3. Средние водители - это люди, которые не попадают в категорию опасных или спокойных водителей, но имеют некоторые нарушения в правилах дорожного движения, например, превышение скорости на короткие расстояния или незначительные нарушения правил.

Модель профиля водителя может быть использована в различных отраслях, таких как автострахование, каршеринг и такси, а также при создании персонализированных сервисов. В каждом из вышеперечисленных примеров есть необходимость в данном продукте.

В такси профиль водителя позволит собирать информацию о качестве поездок, а также обнаруживать агрессивных водителей, что поможет минимизировать риски для пассажиров и других участников дорожного движения. Аналогично в случае каршеринга будет возможно обнаружение различных категорий водидетей, что позволит ограничить перечень услуг некоторым категориям или менять тарифную сетку в зависимости от риска, соответствующего вашему профилю. [1; 2]

В случае персонализированных сервиса, которыми занимаются все мировые автомобильные концерны, определение профиля водителя отчасти уже реализовано, например когда автомобиль подстраивает свои настройки и мультимедийную систему под определенного водителя, что несомненно является полезным при использовании одной машины несколькими людьми одновременно.[3] Но определение профиля водителя поможет автомобилю предугадывать возможные действия человека в различных ситуациях и соответственно быть готовым к любому развитию событий. Также знание профиля водителя и его привычек, позволяет предлагать различный сервис специально для него в зависимости от дороги, времени суток и цели поездки, например реклама заправочных станций при малом количестве топлива.

Если же рассматривать применение в автостраховании, то можно выделить несколько областей для применения.

1. Борьба со страховым мошенничеством. В случае существования модели, способной классифицировать водителей с точностью близкой к 100%, данная модель сможет быть использована для определения водителя за рулем во время ДТП или других страховых случаев, а также для обнаружения факта передачи управления автомобиля третьим лицам и для обнаружения большого множества других видов автомобильного мошенничества.
2. Создание индивидуальной стоимости автострахования в зависимости от стиля вождения водителя, что позволит учитывать риски индивидуально для каждого водителя. На сегодняшний день крупные россий-

ские страховые компании применяют данные телематики в основном для поощрения клиентов за аккуратное вождение и предоставление им скидок.

В данной работе будет рассмотрен метод создания профиля водителей, который отличается от существующих подходов. У него будет сразу несколько важных отличий. Во-первых, в отличие от существующих подходов к профилированию водителей, которые используют данные о вождении для классификации водителя по определенным категориям, результат нашей модели будет выдавать метрику уникальную и репрезентативную для каждого водителя. Во-вторых, в данной работе не будут использоваться модели, которым необходима асессорская разметка стиля вождения, так как она требует большим денежных затрат. При этом при решение поставленной задачи будет рассматривать решение важных смежных задач.

Целью данной работы является:

1. Разработать модель для классификации водителей с точностью не менее 95% и отобрать наиболее оптимальное сочетание признаков, характеризующее водителей, которые должны быть легко доступны для понимания обычным пользователям с целью обеспечения прозрачности итогового продукта.
2. Разработать поиск опасных водителей с помощью поиска аномалий.
3. Разработать модель профиля водителя, обновляемую каждую поездку, которая будет характеризовать стиль вождения водителя.

Объект исследования: поведение водителя за рулем транспортного средства.

Предмет исследования: показатели автомобиля во время движения, модели машинного обучения для поиска аномалий, кластеризации данных и классификации.

Объем и структура работы. Диссертация состоит из введения, четырёх глав, заключения и двух приложений. Полный объём исследовательской работы составляет 45 страниц с 9 рисунками и 11 таблицами. Список литературы содержит 10 наименований.

Глава 1. Распознавание водителей на основе данных вождения

1.1 Описание телематических данных

Данные телематики - это информация о перемещении и состоянии транспортного средства, собираемая с помощью специальных устройств, установленных на автомобиле. Необходимо выделить особенности данных телематики:

1. Точность и актуальность. Данные телематики собираются в режиме реального времени и позволяют получать информацию о месте нахождения и состоянии автомобиля в любой момент времени.
2. Разнообразие источников. Эти данные могут включать параметры движения автомобиля (скорость, ускорение, местоположение), информацию о работе двигателя, давления в шинах и другие характеристики. Это позволяет получить более полную картину о перемещении транспортного средства и его состоянии.
3. Большой объем: телематические устройства постоянно генерируют и обрабатывают огромное количество данных, что может представлять определенные сложности для хранения и анализа информации.
4. Данные телематики не всегда могут быть точными и полными. Например, они могут не учитывать такие факторы, как погодные условия, состояние дорог и т.д. Поэтому необходимо использовать данные телематики в сочетании с другими источниками информации, чтобы получить более полную и точную картину о передвижении транспортного средства.

В данной работе использовались данные, опубликованные HCRL (лаборатория) [4]. Основная область исследований лаборатории исследований по взлому и контрмерам (HCR Lab) - это безопасность, основанная на данных, которая базируется на технологиях машинного обучения и добычи данных для извлечения и изучения полезных знаний из больших данных.

Описание датасета:

1. Время в пути: около 23 часов
2. Протяженность маршрута: около 46 км (туда и обратно)

3. Маршрут движения: между Корейским университетом и стадионом чемпионата мира в Сангаме
4. Количество водителей: 10 (классы от A до J в наборе данных)
5. Табличные данные, состоящие из 94380 строк и 54 колонок, среди которых 6 признаков являются категориальными.

1.2 Выбор метрики для решения поставленной задачи

Перед построением любой модели, в первую очередь необходимо выбрать метрики для оценки итоговых решений. При решении данной задачи необходимо будет классифицировать водителей. При этом каждый водитель совершил две поездки по одинаковым маршрутам. Но каждый из них двигался с разной скоростью, поэтому в данной задаче классы не являются сбалансированными. Также мы хотим делать предсказывать для каждого водителя одинаково хорошо, даже в случае, если время, проведенное за рулем, значительно различается между водителями. Именно поэтому мы будем использовать макро усреднение, которое делает результаты более зависимыми от эффективности модели в отношении малочисленных классов, что особенно важно в ситуациях, когда все классы представляют равный интерес.

В данном случае мы будем использовать метрику "TotalF1:average=Macro". F1 вычисляется отдельно для каждого класса k, пронумерованного от 0 до $M-1$ по формуле $2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$.

Итоговая метрика считается, как макро усреднение формулы выше

$$\frac{\sum_{i=1}^{M} F1_i}{M}$$

.

1.3 Выбор модели для классификации водителей

Одним из лучших решений, которые применяется для классификации и который по умолчанию обычно выдает хорошие результаты, являются модели градиентного бустинга. В наших данных есть не только числовые признаки, а также порядковые и категориальные признаки. В случае порядковых признаков, например «Current_Gear» (текущая передача), обычно используется Label encoding, который каждому состоянию сопоставляет число, при этом порядок состояний должен сохраниться. В случае же категориальных признаков, например «PathOrder» (идентификатор маршрута), обычно не используют Label encoding, так как в таком случае, линейные модели будут плохо работать с таким признаков, модели на основе деревьев смогут работать, но потребуется большая глубина. Лучшим методом кодирования категориальных признаков является Mean encoding, который заменяет категорию на посчитанную статистику таргета у объектов, имеющих данную категорию[5].

`Пример работы Label encoding и Mean encoding`:

id	job	job_label	target
1	Doctor	1	1
2	Doctor	1	0
3	Doctor	1	1
4	Doctor	1	0
5	Teacher	2	1
6	Teacher	2	1
7	Engineer	3	0
8	Engineer	3	1
9	Waiter	4	1
10	Driver	5	0

а) Label encoding

id	job	job_mean	target
1	Doctor	0,50	1
2	Doctor	0,50	0
3	Doctor	0,50	1
4	Doctor	0,50	0
5	Teacher	1	1
6	Teacher	1	1
7	Engineer	0,50	0
8	Engineer	0,50	1
9	Waiter	1	1
10	Driver	0	0

б) Mean encoding

Рисунок 1.1 — Пример Label Encoding и Mean Encoding

Но у Mean Encoding есть заметный минус: При подсчете статистики используем таргет, эта проблема вызывает переобучение модели. Но в реализации Expanding mean, где для подсчета статистики для x_i используются только $y_1, \ldots, y_{i-1}$, самая маленькая утечка таргета среди всех реализаций Mean Encoding. Также данный способ реализован в CatBoost для обработки категориальных признаков. Поэтому Catboost будет использован для решения нашей

задачи, как идеальное сочетание градиентного бустинга и использовании Mean Encoding.

1.4 Обучение модели для классификации водителей

Для обучения и тестирования Catboost, мы разделили данные на 3 части: 20% на тест, 20% на валидацию и 60% на обучение модели.

Обучение Catboost, используя все признаки дал результат 0.96 F1-Macro, что является несомненно превосходным результатом для начала. Большое множество признаков могут запутывать модель, так как не все из них значимо влияет на таргет. Поэтому необходимо сделать отбор признаков, которые наиболее важны для модели и отбросить те, которые только мешают ей обучаться. Есть несколько способ для определения важности признаков, рассмотрим наиболее популярные и часто используемые.

1. MDI (Mean Decrease in Impurity)

   `Пример топ-10 признаков согласно MDI для Catboost`:

Рисунок 1.2 — Признаки по MDI для Catboost

2. Результат работы алгоритма Add-Del для отбора наиболее важных признаков
    - Accelerator_Pedal_value
    - Engine_in_fuel_cut_off
    - Long_Term_Fuel_Trim_Bank1
    - Torque_of_friction
    - Engine_coolant_temperature
    - Calculated_LOAD_value
    - Maximum_indicated_engine_torque
    - Wheel_velocity_rear_left
    - Master_cylinder_pressure
    - Calculated_road_gradient

Затем запустили CatBoost, обучая его только на признаках из списков выше. Результаты обучения представлены в таблице ниже.

Таблица 1

Сочетание признаков	F1:Macro
MDI	0.93
Add-del	0.99

Хорошо видно, что в случае MDI значение метрики на тесте только ухудшились, а в случае подбора оптимального списка сочетания признаков с помощью Add-Del F1:Macro улучшилась довольно сильно. При этом доля предсказаний, которые не равны таргету составляет 1.2%. Таким образом accuracy лучшей версии модели составляет 98.8%

Глава 2. Поиск оптимального сочетания признаков

2.1 Причина поиска оптимального сочетания признаков

Необходимо иметь список признаков, который будет явно охарактеризовать водителя, причем большая часть из них будет понятна среднестатическому пользователю. Так как одна из самых больших проблем любого нового продукта, представленного на рынке, с которым среднестатический пользователь плохо знаком – это недоверие к данному продукту. Таким образом, если пользователь будет явно понимать, какие признаки используют модель для создания профиля водителя, то продукт станет более прозрачным для итогового потребителя. При этом если брать только такие признаки, как ускорение, торможение и скорость, то будет невозможно построить модель с хорошим качеством.

2.2 Теоретические обоснование методов машинного обучения, используемых для решения поставленной задачи

Для решения поставленной задачи в данной главе были использованы два алгоритма, основанные на деревьях, а именно градиентный бустинг и случайный лес. Далее будут изложены основные моменты, связанные с реализацией указанных выше алгоритмов.

2.2.1 Random Forest

Случайный лес (Random Forest) – это алгоритм машинного обучения, который используется для классификации и регрессии. Он основан на идее объединения множества деревьев решений, каждое из которых обучается на случайной подвыборке данных.

Процесс работы случайного леса можно разделить на несколько этапов:

1. Обучение каждого дерева решений на случайной подвыборке тренировочных данных. Каждое дерево генерируется случайным образом, путем выбора случайных признаков и случайных значений для разделения данных на узлы. Важное Свойство решающего дерева с большой глубиной:
    (a) bias - низкий
    (b) variance - высокий
2. Объединение результатов предсказаний всех деревьев в одно решение. После обучения каждого дерева, результаты предсказаний объединяются в одно решение, которое используется для предсказания класса.

Таким образом, случайный лес - это эффективный алгоритм машинного обучения для решения задач классификации и регрессии, который основывается на объединении множества деревьев решений.[5; 6]

2.2.2 Градиентный бустинг над деревьями

Для начала необходимо разобрать общую идею работы градиентного бустинга. Gradient Boosting – это алгоритм машинного обучения, который использует последовательное построение слабых моделей и комбинирование их вместе для получения более сильной модели.

Принцип работы градиентного бустинга состоит из следующих шагов:

1. Инициализация: начинается с инициализации модели простой базовой моделью, например, решающим деревом, которое предсказывает среднее значение целевой переменной.
2. Обучение модели: затем модель обучается на train. Остатки предсказания считается как разность между ожидаемым значением и предсказанным значением.
3. Затем остатки из первой модели используются для создания следующей модели, которая будет предсказывать эти Остатки.
4. Построение следующих моделей: это процесс повторяется, итерация за итерацией. Каждый новый добавленный шаг учитывает оставшиеся

остатки, не обрабатываемые предыдущими шагами, повышая точность предсказания целевой переменной.

`Общая схема градиентого бустинга`:

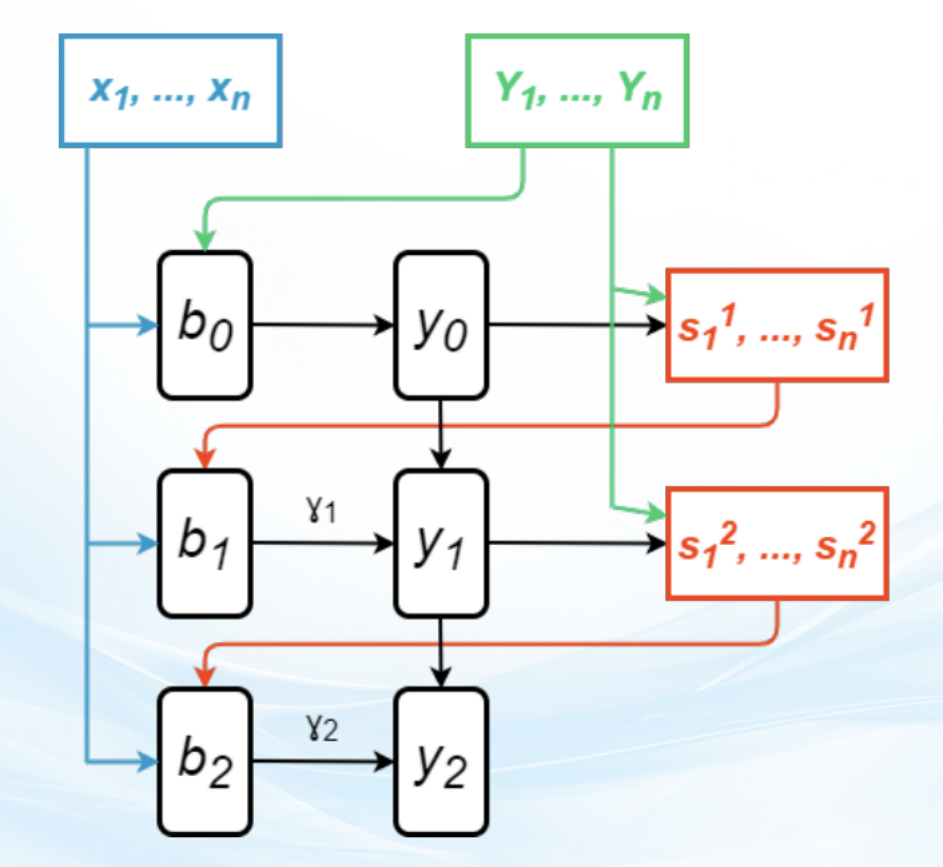


Рисунок 2.1 — Общая схема градиентого бустинга [5]

В итоге, градиентный бустинг является итеративным подходом, который использует остатки для обучения новых моделей в композитную модель. Каждый добавленный шаг уменьшает ошибку предсказания, в конечном итоге создавая более точную модель для решения задачи машинного обучения.[5; 7]

2.2.3 Модели для снижение размерности

Для начала необходимо выяснить причины использование моделей для понижения размерности при работе с большим количеством признаков:

– Уменьшение размерности помогает выделить наиболее важные признаки, характеризующие целевой признак.

– Некоторые алгоритмы машинного обучения плохо работают с большим количеством признаков, так как это может привести к переобучению модели, ухудшению ее качества и повышению времени обучения.
– Снижение размерности помогает убрать ненужные признаки, которые не несут полезной информации и могут мешать обучению модели.
– Также снижение размерности решает проблему, связанную с «проклятием размерности»

В данном подразделе были опробованы три модели понижения размерности, а именно: PCA, UMAP, LDA, при этом до применения каждого из данных методов, мы применили стандартизацию данных.

1. PCA (Principal Component Analysis) - это метод анализа данных, который используется для уменьшения размерности данных путем нахождения главных компонент. Он основан на идее, что большинство информации в данных находится в первых нескольких главных компонентах, которые могут быть использованы для представления данных в более компактной форме.
2. UMAP (Uniform Manifold Approximation and Projection) - это алгоритм машинного обучения, который также используется для уменьшения размерности данных. Он базируется на идее, что данные, которые находятся на близких расстояниях друг от друга, должны быть расположены близко на двумерном пространстве.
3. LDA (Linear Discriminant Analysis) - это статистический метод, который используется для классификации данных. Он основывается на идее, что класс может быть разделен на две или более группы, и что эти группы имеют разные средние значения и ковариационные матрицы.

При этом каждый из этих способов имеет свои преимущества и недостатки:

1. PCA позволяет уменьшить размерность данных, сохраняя при этом большую часть их информации. Это может быть полезно, например, для визуализации данных в пространстве меньшего числа измерений. Однако, PCA не учитывает нелинейные зависимости между переменными, что может привести к потере информации.
2. UMAP также позволяет уменьшить размерность данных и сохранить большую часть информации, но он также учитывает нелинейные отно-

шения между переменными. Это может быть полезным, например, при анализе данных, полученных из многомерных пространств, таких как изображения.

3. LDA используется для классификации данных на основе их распределения. Он учитывает различия в средних значениях и ковариационных матрицах классов, что позволяет более точно классифицировать данные. Однако, LDA может быть менее эффективным, если данные не имеют достаточно разных средних значений и ковариаций между классами.

2.3 Поиск важных признаков с помощью моделей машинного обучения

2.3.1 Использование случайного леса

В пункте 1.4, для обучения Catboost было найдено оптимальное сочетание признаков. Но данное оптимальное сочетание признаков было подобрано только для данной модели обучения, в данном подразделе будет реализовано использование других моделей машинного обучения для поиска важных признаков, а также модели машинного обучения, основанные на снижение размерности. Попробуем еще одну модель, основанную на деревьях, а именно «случайный лес» (Random Forest)

`Пример топ-10 признаков согласно MDI для Random Forest:`

Рисунок 2.2 — Признаки по MDI для Random Forest

Хорошо видно, что топ-10 признаков, полученные с помощью MDI для Random Forest отличается от аналогичных признаков для Catboost только порядком.

2.3.2 Использование PCA

Для поиска наиболее важных признаков с помощью алгоритмов снижения размерности предлагается:

1. Применить алгоритм снижения размерности на данных, причем предварительно к ним было применена стандартизация
2. Затем мы получаем n - компонент, где n - размер пространства, которое получается после применения снижения размерности. Затем для каждой из n компонент строится матрица корреляции с первоначальными признаками, а также считается статистическая значимость для проверки скоррелированности первоначальных признаков и полученных компонент. В данной задаче для более удобной визуализации будет использовано уменьшение размерности в 2-мерное пространство.

Так как первоначальных признаков было много, то будут приведены только топ-10 наиболее скоррелированных изначальных признаков для каждой из двух компонент.

Таблица 2 — Таблица корреляции с первой компонентой PCA

Признак	p-value	Корреляция
Engine_speed	0.0	0.94
Maximum_indicated_engine_torque	0.0	0.93
Torque_converter_turbine_speed_-_Unfiltered	0.0	0.91
Torque_converter_speed	0.0	0.91
Wheel_velocity_front_left-hand	0.0	0.88
Wheel_velocity_front_right-hand	0.0	0.88
Wheel_velocity_rear_left-hand	0.0	0.88
Wheel_velocity_rear_right-hand	0.0	0.88
Vehicle_speed	0.0	0.88
Absolute_throttle_position	0.0	0.82

Таблица 3 — Таблица корреляции со второй компонентой PCA

Признак	p-value	Корреляция
Calculated_LOAD_value	0.0	0.8068
Engine_torque	0.0	0.6591
Engine_torque_after_correction	0.0	0.6542
Acceleration_speed__-_Longitudinal	0.0	0.6389
Flywheel_torque	0.0	0.6129
Flywheel_torque__(after_torque_interventions)	0.0	0.6084
Current_Gear	0.0	-0.5578
Intake_air_pressure	0.0	0.5445
Wheel_velocity_rear_left-hand	0.0	-0.5230
Wheel_velocity_rear_right-hand	0.0	-0.5228

Также был построен график проекции точек на две компоненты при сжатии размерности с помощью алгоритма PCA приложения:А.

2.3.3 Использование UMAP

Также был построен график проекции точек на две компоненты при сжатии размерности с помощью алгоритма UMAP приложения:Б.

Таблица 4 — Таблица корреляции с первой компонентой UMAP

Признак	p-value	Корреляция
Activation_of_Air_compressor	0.0	-0.2442
Torque_of_friction	0.0	-0.1546
Converter_clutch	3.1796e-187	0.1059
Engine_in_fuel_cut_off	3.5962e-183	-0.1047
Engine_soacking_time	1.7080e-180	-0.1040
Long_Term_Fuel_Trim_Bank1	1.4279e-171	0.1014
Engine_coolant_temperature.1	3.1427e-167	-0.1001
Flywheel_torque__(after_torque_interventions)	6.6055e-156	0.0966
Flywheel_torque	4.2050e-155	0.0963
Maximum_indicated_engine_torque	2.3479e-148	0.0942

Таблица 5 — Таблица корреляции со второй компонентой UMAP

Признак	p-value	Корреляция
Activation_of_Air_compressor	8.9829e-117	-0.0835
Indication_of_brake_switch_ON/OFF	8.3157e-113	0.0820
Engine_soacking_time	4.6611e-93	-0.0744
Intake_air_pressure	1.3809e-90	-0.0734
Accelerator_Pedal_value	6.6703e-89	-0.0727
Engine_torque_after_correction	7.3076e-80	-0.0688
Engine_torque	1.6693e-76	-0.0673
Flywheel_torque__(after_torque_interventions)	4.6174e-74	-0.0662
Flywheel_torque	1.5342e-71	-0.0650
Fuel_consumption	2.0843e-68	-0.0635

2.3.4 Использование LDA

Таблица 6 — Таблица корреляции с первой компонентой LDA

Признак	p-value	Корреляция
Long_Term_Fuel_Trim_Bank1	0.0	0.7587
Torque_of_friction	0.0	-0.4779
Engine_coolant_temperature.1	0.0	-0.3630
Activation_of_Air_compressor	0.0	-0.3386
Engine_soacking_time	0.0	-0.1568
Calculated_LOAD_value	0.0	-0.1444
Master_cylinder_pressure	0.0	-0.1440
Engine_coolant_temperature	0.0	-0.1379
Maximum_indicated_engine_torque	5.1865e-308	0.1359
Current_spark_timing	3.4143e-152	0.0954

Таблица 7 — Таблица корреляции со второй компонентой LDA

Признак	p-value	Корреляция
Intake_air_pressure	0.0	-0.5829
Accelerator_Pedal_value	0.0	-0.4603
Engine_coolant_temperature.1	0.0	0.2545
Engine_soacking_time	0.0	-0.2147
Activation_of_Air_compressor	0.0	0.1928
Long_Term_Fuel_Trim_Bank1	0.0	0.1719
Maximum_indicated_engine_torque	0.0	-0.1550
Master_cylinder_pressure	1.5493e-222	0.1155
Minimum_indicated_engine_torque	2.5481e-168	-0.1004
Engine_speed	2.0416e-151	-0.0952

Также был построен график проекции точек на две компоненты при сжатии размерности с помощью алгоритма LDA приложения:B.

2.4 Итоговый вывод насчет поиска наиболее оптимального сочетания признаков

Среди методов снижения размерности, которые были опробованы в предыдущих пунктах только компоненты, полученные с помощью PCA сильно скоррелированы с изначальными признаками.

Также LDA нашел только один достаточно скоррелированный признак (с корреляцией больше 0.7). Причем это признаки Long_Term_Fuel_Trim_Bank1. Необходимо более внимательно посмотреть топ признаков, найденных с помощью PCA. Причем если смотреть только на те признаки, где корреляция больше 0.7, то надо смотреть полностью на первую таблицу и на признак Calculated_LOAD_value.

Напомним смысл каждого признака, а также встречался ли он раннее при подборе оптимального сочетания признаков для обучения CatBoost и правильно ли ставить на нем акцент для конечного потребителя, не запутаем ли мы его.

1. Engine_speed - Скорость оборотов двигателя в минуту. Данный признак не был в топе при обучение CatBoost и при подборе признаков с помощью алгоритма Add-Del. При этом данный признак нельзя включать в итоговый список признаков, так как для каждой машины, типа двигателя (атмосферный/атмосферный), от типа топлива (бензин/дизель) данный параметр будет находиться по умолчанию при разных значениях.
2. Maximum_indicated_engine_torque - максимальный указанный крутящий момент двигателя. Переменная Maximum_indicate_engine_torque представляет максимальное значение крутящего момента, которое может быть достигнуто автомобилем при заданных оборотах двигателя. Данный признак входил в топ признаков с помощью Add-Del и MDI.
3. Torque_converter_turbine_speed_-_Unfiltered - скорость конвертации крутящего момента в скорость движения, не отфильтрованная прибором. Данный признак не входил в топ по Add-Del. Также данная переменная сильно зависит от автомобиля, а не от водителя, что не позволяет ее использовать в итоговом списке.

4. Torque_converter_speed - скорость конвертации крутящего момента в скорость движения. Данный признак не входил в топ по Add-Del. Также данная переменная сильно зависит от автомобиля, а не от водителя, что не позволяет ее использовать в итоговом списке.
5. Wheel_velocity_front_left-hand - скорость переднего левого колеса. Данный признак не входил в топ по Add-Del, но данный признак понятен конечному пользователю.
6. Wheel_velocity_front_right-hand - скорость переднего правого колеса. Данный признак не входил в топ по Add-Del, но данный признак понятен конечному пользователю.
7. Wheel_velocity_rear_left-hand - скорость заднего левого колеса. Данный признак не входил в топ по Add-Del, но данный признак понятен конечному пользователю.
8. Wheel_velocity_rear_right-hand - скорость заднего правого колеса. Данный признак не входил в топ по Add-Del, но данный признак понятен конечному пользователю.
9. Vehicle_speed - скорость транспортного средства. Данный признак не входил в топ по Add-Del, но данный признак понятен конечному пользователю.
10. Absolute_throttle_position - абсолютное положение дроссельной заслонки. Данный признак не входил в топ по Add-Del, при этом данный признак сложно объяснить конечному пользователю, а также отчасти зависит от настроек автомобиля.
11. Calculated_LOAD_value - доля текущей нагрузки двигателя от максимально возможной. Данный признак очень важный, он также входил в предыдущие списки наиболее важных признаков.

Как хорошо видно, PCA хоть и нашел важные признаки, которые характеризуют поездку, но не поведение конкретного водителя. Таким образом итоговый список наиболее важных признаков, который характеризует водителя и зависит в первую очередь от паттернов поведения водителя – это полученный список с помощью алгоритма Add-Del, а также добавленные к нему признаки. Расширение исходного набора признаков для анализа включает в себя несколько хорошо интерпретируемых признаков, которые понятны пользователям. В частности, были добавлены такие признаки, как скорость транспортного средства

(Vehicle_speed), продольное ускорение (Acceleration_speed_-_Longitudinal), поперечное ускорение (Acceleration_speed_-_Lateral), скорость вращения рулевого колеса (Steering_wheel_speed) и угол поворота рулевого колеса (Steering_wheel_angle). Эти признаки обладают относительно простым и понятным описанием, что облегчит понимание результатов анализа не только специалистами, но и обычными пользователями данных.

Но при добавлении новых признаков, необходимо проверить, как это скажется на итоговую метрику качества классификаци водителей. Для этого сравним метрику на тесте обученного CatBoost с признаками, отобранными Add-Del и Add-Del с добавленными признаками.

Таблица 8 — Метрики обученного CatBoost на разном сочетании признаков

Сочетание признаков	F1:Macro	accuracy
Add-del	0.99	98.8%
Add-del + добавленные признаки	0.98	96.3%

Таким образом видно, что хоть добавление признаков и портит итоговые метрики, они все еще остаются на довольно высоком уровне, при этом появляются признаки, которые легко можно объяснить конечному потребителю и обосновать ту или иную стоимость услуг

Глава 3. Поиск опасных водителей с помощью поиска аномалий

3.1 Мотивация поиска опасных водителей с помощью поиска аномалий

Вдохновением для данного метода послужил пример работы антифрода[8].

А именно, в антифроде обычно используются как автоматические проверки, так и ручная модерация, для получения наибольшей эффективности в борьбе с мошенническими операциями.

Автоматическая проверка основана на использовании алгоритмов, которые анализируют большое количество данных. Эта проверка происходит быстро, в режиме реального времени, и может блокировать сомнительные операции или моментально оповестить пользователя о возможной подозрительной операции.

Однако автоматические проверки могут приводить к ошибкам и ложным срабатываниям, поэтому дополнительно используется ручная модерация. Ручной надзор означает, что операции проходят проверку представителей компании, которые могут перепроверить подозрительные операции, уточнить детали, запросить дополнительные документы и принять окончательное решение. Это позволяет значительно уменьшить количество ложных положительных срабатываний и увеличивает эффективность системы.

Как правило, в антифрод системах ручная модерация применяется только в случаях, когда автоматические проверки дали подозрительный результат, или когда на операцию требуется более тщательный анализ.

Таким образом, поиск опасных водителей – это всего лишь поиск потенциально опасных водителей, которые затем отправятся на ручную проверку.

3.2 Методы поиска аномалий

Поиск аномалий означает определение необычных и редких событий в данных путем обнаружения отклонений от общего шаблона. Решение этой задачи может быть реализовано с использованием нескольких подходов к машинному обучению:

1. Методы статистического моделирования, такие как методы, которые используют распределение данных, например, нормальное распределение.
2. Обнаружение аномалий на основе плотности: Этот метод основан на том, что распределение данных для нормальных значений является более плотным, чем распределение данных для аномалий. Поэтому этот метод основан на определении плотности данных для каждой переменной и сравнении этой плотности для каждого значения с порогом, который определяет, что является аномалией, а что нет.
3. Обнаружение аномалий на основе кластеризации: Формирование кластеров может помочь выявить аномальные наблюдения. Аномалии можно определить как объекты, которые не принадлежат ни одному кластеру или принадлежат кластеру менее числу объектов. Этот метод может быть очень эффективным в случае, когда у нас много наблюдений и наша задача заключается в выявлении глобальных аномалий.

Для нашей задачи мы будем использовать два разных, хорошо зарекомендованных метода для поиска аномалий:

1. Local Outlier Factor
2. Isolation Forest

При этом стоит вопрос какие данные будут использовать в данной задаче, а также нужна ли предобработка данных. В случае использования методов поиска аномалий, основанных на плотности данных, то необходимо предварительно стандартизировать данные. Также мы будем применять модель на всех данных, на данных после уменьшения размерности с помощью PCA, который показал себя лучше всего во второй главе, а также на данных только с признаками, которые попали в топ наиболее важных признаков по результату работы

во второй главе. В последующих подпунктах будет представлено подробное описание принципа работы каждого из данного способа.

Методы для поиска аномалий, которые будут использоваться являются примеров моделей машинного обучения без учителя, таким образом, перед нами стоит важная задача проверки корректности работы моделей. Тут было предложено мною два способа для решения данной задачи:

1. Ручной просмотр данных, порядка 100 различных точек, базовая проверка на корректность и подсчет accuracy.
2. Обучение алгоритмов на одной типе водителе, а тестирование модели на противоположном типе водителей, ожидаемые аномалии будут принадлежать противоположному типу водителю.

3.3 Local Outlier Factor

Модель Local Outlier Factor (LOF) - это метод нахождения аномалий в машинном обучении, который использует понятие локальной плотности данных для оценки степени аномальности объектов.

LOF оценивает локальную плотность данных для каждого объекта, сравнивая его окрестность с окрестностью его соседей. Окрестность определяется через выбранные пользователем параметры - минимальное количество соседей и радиус окрестности. Для каждого объекта LOF вычисляется как отношение средней плотности в его окрестности к плотности его соседей. Если LOF меньше единицы, то данный объект считается нормальным. В противном случае объект считается аномалией.

LOF является методом обнаружения аномалий в многомерных данных. Он также может быть применен для обнаружения аномалий в больших наборах данных, так как он не требует обработки всех данных сразу, а использует информацию только о локальности окрестностей объектов.

3.4 Isolation Forest

Isolation Forest - это алгоритм машинного обучения, используемый для обнаружения аномалий в данных. В отличие от традиционных методов обнаружения аномалий, Isolation Forest использует ансамбль деревьев решений, где каждое дерево строится путем разделения выборки на две части по случайно выбранному пороговому значению для случайно выбранного признака.

Таким образом, каждое дерево в лесу решений изолирует случайный поднабор данных, деля объекты на кучи по мере прохождения от корня дерева до его листов. Количество разбиений, необходимых для изоляции объекта, является мерой его аномальности. Каждый объект имеет значение аномалии, для которых он находится в тестовом узле.

Процесс построения Isolation Forest включает шаги:

1. Выбор случайного подмножества данных для построения дерева решений.
2. Случайный выбор признака из подмножества данных и случайный пороговое значение между максимумом и минимумом значения признака.
3. Разделение данных по заданным условиям.
4. Повторение шагов 1-3 до тех пор, пока не будет достигнуто максимальное количество деревьев в лесу.

После построения Isolation Forest для каждого объекта рассчитывается его аномальный показатель, который определяется как среднее количество разбиений, необходимых для изоляции объекта на всех деревьях решений в лесу.

Isolation Forest имеет несколько преимуществ по сравнению с традиционными методами обнаружения аномалий. Во-первых, он может обрабатывать большие объемы данных быстрее, чем многие другие методы. Во-вторых, он эффективно обнаруживает аномалии в многомерных данных, что невозможно для некоторых других методов. Наконец, Isolation Forest может обрабатывать данные с несбалансированной категориальной структурой, так как он не требует целостных данных.

3.5 Проверка корректности поиска аномалий

В общем случае при необходимости проверить корректность поиска аномалий, есть возможность обучить модель на train, затем же предсказать аномалии для test, который отличается от train и проверить полученный результат. При решении нашей задаче, также есть возможность предварительно разметить аномалии асессорами и затем уже запустить модель для предсказания аномалия. В таком случае мы сможем посчитать точность и полноту полученного алгоритма и принять решение о целесообразности использования модели или необходимости ее дообучения.

Так как в данной работе рассматриваются только модели без возможной доразметки данных асессорами, то был предложен алгоритм для проверки корректности:

1. Выбираем двух водителей, в идеале поведение которых за рулем наиболее сильно отличается друг от друга
2. Обучаем модель на данных одного водителя
3. Предсказываем аномалии на данных уже двух водителей
4. Подсчитываем, какая доля аномалий соответствует каким водителям, ожидаем, что 95% соответствует водителю, которого не было при обучении. Таким образом, мы допускаем 5% ошибок, по аналогии с 5% ошибкой 1 рода при проверке гипотез в математической статистике.

Но предложенный выше алгоритм можно улучшить, что и было произведено в работе, а именно: были взяты все пары водителей и подсчитана доля ошибок для каждого водителя, затем использовалась медиана доли ошибок при предсказании, которая не должна превышать 5%.

Также для дополнительной корректности было решено воспользоваться ручной проверкой корректности алгоритма. При ручной разметке необходимо было ответить на вопрос "Предсказание для данной точки явно не верно предсказано моделью?"и рассмотреть 100 произвольных точек после предсказания алгоритма, а также первоначальные признаки, которые соответствовали данной точке. Затем сравнивались ручная разметка и предсказания моделей. Требовалось, чтобы точность была более 95%.

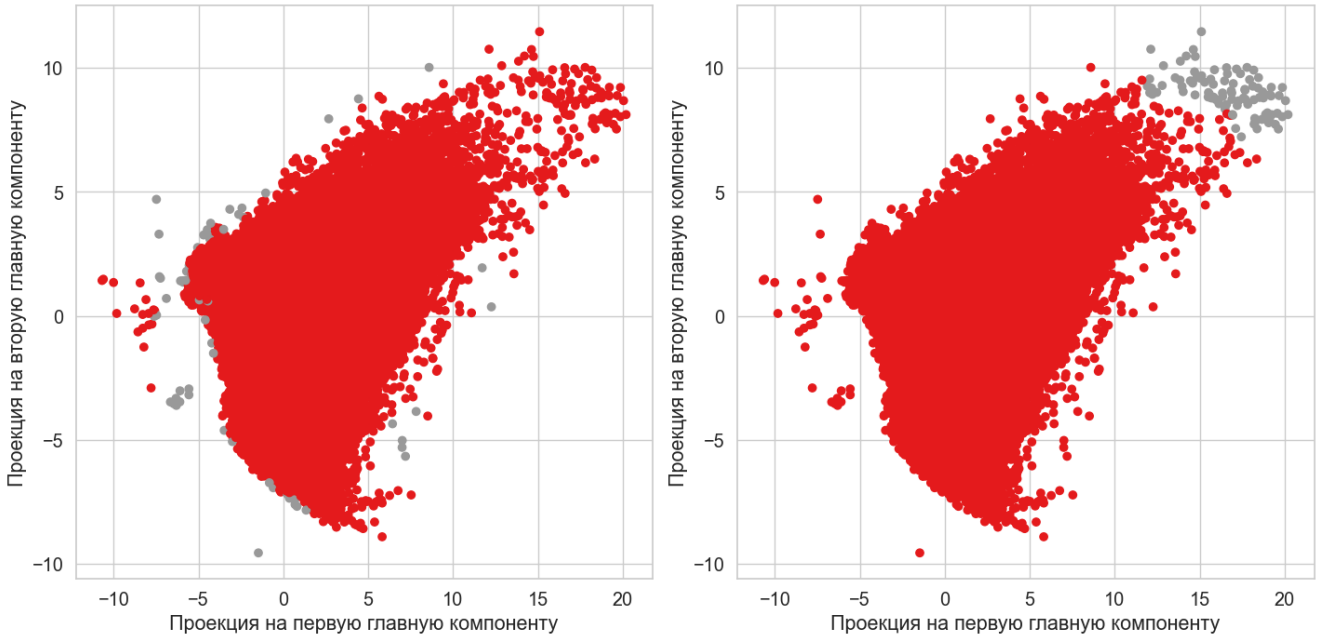
3.6 Результаты поиска аномалий

Ниже предоставлены результаты работы двух различных моделей для поиска аномалий. Так как при уменьшении размерности начальных данных с помощью PCA позволяет отобразить наглядные картинки, а также визуально увидеть отличия в работе двух алгоритмов.

Во всех ниже рассмотренных вариантах, к данным сначала была применена стандартизация данных, так как в случае модели Local Outlier Factor используется локальная плотность, т.е. данная модель является примером метрической модели.

В представленных картинках ниже, выбросы будут отмечены серым цветом.

`Применение Local Outlier Factor и Isolation Forest`:



а) Local Outlier Factor (contamination=0.001) б) Isolation Forest (contamination=0.001)

Рисунок 3.1 — Пример Local Outlier Factor и Isolation Forest

Хорошо видно, что работа алгоритмов отличается, в случае Local Outlier Factor аномалиями являются точки, которые на границе основного кластера и расположены удалены от других точек. В случае применения Isolation Forest аномалиями являются точки, которые условно выделяются в отдельный кластер, что согласуется с алгоритмом работы модели.

Таблица 9 — Таблица распределения аномалий водителей на данных преобразованных PCA

Водитель	Isolation Forest	Local Outlier Factor
A	0.231	0.149
B	0.096	0.146
C	0.199	0.118
D	0.115	0.123
E	0.083	0.083
F	0.046	0.094
G	0.021	0.048
H	0.076	0.092
I	0.047	0.072
J	0.086	0.075

Таблица 10 — Таблица распределения аномалий водителей на всех данных

Водитель	Isolation Forest	Local Outlier Factor
A	0.278	0.139
B	0.106	0.141
C	0.190	0.099
D	0.159	0.124
E	0.073	0.089
F	0.018	0.094
G	0.008	0.074
H	0.089	0.084
I	0.017	0.075
J	0.062	0.082

Вывод: При обучении на данных, преобразованных PCA, мы получили согласно модели IsolationForest топ-3 водителей с наибольшей долей аномалий – это водители A, C, D. Такой же результат мы получили при использовании всех признаков данной моделью. При обучении на данных, преобразованных PCA, мы получили согласно модели Local Outlier Factor топ-3 водителей с наибольшей долей аномалий – это водители A, B, D. При использовании модели Local Outlier Factor и всех признаков, получаем эту же тройку водителей, но в другом порядке, а именно: B, A, D. Таким образом, используя различные методы и данные, мы получили, что водители A и D попали в топ-3 водителей с наибольшей аномалией, а водители C и B зависят от используемой модели. Но необходимо помнить, что с помощью поиска аномалий происходит только поиск потенциально опасных водителей, которые затем могут отправиться на ручную проверку классифицированными специалистами. В следующем параграфе будет рассмотрен еще один способ для поиска опасных водителей и будет наиболее важно посмотреть на согласованность полученных результатов.

Глава 4. Динамический профиль водителя

4.1 Мотивация алгоритма по формированию онлайн профиля водителя

В предыдущих главах разбирались схожие задачи, но все же решающие более простую задачу. В первой и во второй главе было найдено оптимальное сочетание признаков, которое позволяло с высокой точностью классифицировать определенного водителя. В третьей главе же был реализован поиск опасных водителей с помощью поиска аномалий. Но это только позволяло выявить потенциальных опасных водителей и затем отправить выявленных водителей на ручную модерацию. Таким образом, поиск опасных водителей не решал задачу по созданию профиля водителя.

Модель, изложенная в данной главе, представляет собой эффективный инструмент для решения данной задачи. Но для начала перечислим возможные решения поставленной задачи:

1. Обучение с учителем. Для этого необходима разметка асессоров, что делает процесс дорогим и долгим, а также возможных только для довольно больших компаний. Причем написание инструкции для асессоров и ее редактирование до получения хорошего качества асессороской разметки может заниматься несколько месяцев и множество итераций в редактировании инструкции. Причем после разметки могут применяться нейронные сети: lstm, cnn-1d
2. Обучение без учителя учителем. В данной методе не нужна будет асессорская разметка данных, что несомненно упрощает сбор и обработку первоначальных данных. Но тут отсутствуют модели, которые дают хороший baseline.

Но мы будем рассматривать только обучение без учителя, так как нет готовых проверенных решений и нет ресурсов на ручную разметку и просмотр каждой секунды вождения водителей-испытателей.

4.2 Обоснование алгоритма по формированию онлайн профиля водителя

Данный метод использует кластеризацию, но не в классическом понимании. В большинстве методов кластеризация использовалась для поиска различных типов водителей. Тем не менее, при обработке больших объемов информации и разнообразных параметров, традиционные методы кластеризации оказываются недостаточно эффективными в данной задаче и не способны разделить классы с достаточной точностью.

Для того чтобы построить профиль водительского поведения на основе данных о водителе, нам необходимо использовать алгоритмы кластеризации, чтобы сегментировать данные на основе схожести в поведении и группировать их. Каждый из этих сегментов представляет собой микро-поведение водителя. Создание профиля водителя из данных о вождении - это повторяющийся двухэтапный процесс:

1. Сегментация данных о вождении (в данном случае будет применяться k-means) после каждой поездки.
2. 2. Гармонизация сегментов.

Каждый из этих шагов будет более детально рассмотрен в следующих подразделах. Процесс гармонизации запускается до тех пор, пока мы не получим стабильный список кластеров (т.е. новые данные по поездке не изменяют существующие кластеры). В данном параграфе мы также используем метрику частоту ошибочных обнаружений (FDR). Метрика FDR используется для оценки того, насколько точно сегменты поездки передают поведение водителя. FDR считается согласно формуле

$$FDR = \frac{FP}{FP + TP}$$

`Схема работы алгоритма`:

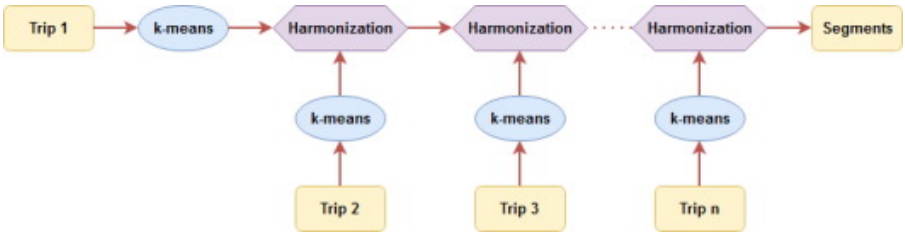


Рисунок 4.1 — Схема работы алгоритма по созданию профиля водителя[9]

4.2.1 Сегментация данных о вождении

Дальше будет рассматриваться применение алгоритма k-средних к кинематическим данным с целью создания кластеров, представляющих микро-стили вождения. Поскольку кластеризация применяется отдельно к каждому периоду вождения (поездке), мы получаем несколько списков кластеров (т.е. один список кластеров на каждую поездку). Эти списки очень зависимы друг от друга, что может привести к появлению избыточных или перекрывающихся кластеров. Это связано с тем, что мы имеем дело с одним водителем, который склонен к схожему поведению в разные периоды вождения. В связи с этим требуется дополнительный процесс объединения, перегруппировки и исключения избыточных кластеров, который мы называем гармонизацией. Цель гармонизации заключается в преобразовании кластеров, полученных из разных поездок, в единый список не перекрывающихся данных кластеров.

4.2.2 Процесс гармонизации кластеров

Напомню, что цель гармонизации заключается в преобразовании кластеров, полученных из разных поездок, в единый список не перекрывающихся данных кластеров. Таким образом ключевой проблемой при гармонизации является пересечение кластеров (сфер). В случае 2-мерного пространства данная задача легко решается, но нам необходимо ее решить в n-мерном пространстве. В данном случае, формулы значительно усложняются, однако концептуально ничего не изменяется, поскольку существует лишь три варианта пересечения сфер.

Можно выделить 3 три следующих случая взаимного расположения кластеров:

1. $S_1 \cap S_2 = \emptyset$: Кластеры S_1 и S_2 они разделены и не пересекаются.
2. $S_1 \cap S_2 \neq \emptyset$: Кластеры S_1 и S_2 пересекаются.
3. б.о.о. $S_1 \subseteq S_2$: Кластер S_1 полностью поглощается кластер S_2 .

Для того, чтобы выделять и обрабатывать вышеупомянутые случаи, мы в первую очередь измеряем объем (volume) и метрику IOU(Intersection Over Union), которая вычисляется по формуле:

$$\text{IOU}\left(S_1, S_2\right) = \frac{\text{volume}\left(S_1 \cap S_2\right)}{\text{volume}\left(S_1 \cup S_2\right)}$$

Очевидно, что для при знании volume(S_1), volume(S_2), volume(S_1) и volume($S_1 \cap S_2$), легко вычислить IOU. Таким образом, ключевой проблемой является вычисление volume($S_1 \cap S_2$), при знании только центра n-мерных сфер и их радиусов.

Распишем для каждого возможного взаимного расположения сфер, чему равен volume($S_1 \cap S_2$).

Но для начала выпишем формулы [10, с.66-70] для вычисления volume(S), где R и n являются радиусом, и размером пространства, а Γ это гамма-функция.

$$\text{volume}(S) = R^n \times \frac{\pi^{(n/2)}}{\Gamma\left(\frac{n}{2}+1\right)}$$

1. $S_1 \cap S_2 = \emptyset$: volume($S_1 \cap S_2$) = 0.
2. $S_1 \cap S_2 \neq \emptyset$: Для начало необходимо вычислить

$$c_1 = \frac{d^2 + r_1^2 - r_2^2}{2d}$$
$$c_2 = \frac{d^2 - r_1^2 + r_2^2}{2d}$$

   Затем объем сферической крышки может быть выражен в терминах гамма-функции Γ и регуляризованной неполной бета-функцией I.

$$V_n^{cap}(r, a \geq 0) = \frac{1}{2}\frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} r^n I_{1-a^2/r^2}\left(\frac{n+1}{2}, \frac{1}{2}\right)$$
$$V_n^{cap}(r, a < 0) = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} r^n - V_n^{cap}(r, -a)$$

   Тогда volume($S_1 \cap S_2$) = $V_n^{cap}\left(r_1, c_1\right) + V_n^{cap}\left(r_2, c_2\right)$.
3. $S_1 \subseteq S_2$ или $S_2 \subseteq S_1$: volume($S_1 \cap S_2$) = min(volume(S_1), volume(S_2)).

4.3 Результат применения модели по созданию профиля водителя

Таблица 11 — Количество полученных кластеров для каждого водителя

Водитель	Кол-во кластеров	FDR
A	354	11.7%
B	368	4.5 %
C	304	5.1%
D	301	5.8%
E	337	4.5 %
F	265	6.0 %
G	276	9.2 %
H	271	6.5 %
I	298	10.0 %
J	280	10.2%

Хорошо видно, что водители с наибольшим количеством кластеров, то есть с наибольшим количеством задействованных паттернов при вождении – это водители: A,B,E. Водители: C, D, I возглавляют среднюю группу, оставшиеся водители: J, H, G,F - попадают в группу наиболее спокойных водителей. При этом важно отметить, что частично данный результат согласуется с результатами в 3 параграфе, а именно A, B являются наиболее активными водителями, их можно включать в кластер "потенциально опасных водителей"или "агрессивных водителей". Также алгоритм выделил водителя E, при поиске аномалий водитель E был в медиане распределения, таким образом нет явного противоречия. Также необходимо отметить, что средний FDR равен 7.15%, что является хорошим показателем. Таким образом, была получена модель создания профиля водителя, причем с возможностью динамически обновляемого профиля.

Заключение

Основные результаты работы заключаются в следующем.

Были рассмотрены 4 крупные задачи в рамках данной исследовательской работы. В каждой из них были предложены способы решения поставленной задачи, реализация, а также указаны причины выбора того или иного решения.

1. Обучен CatBoost для классификации водителей и реализован жадный перебор признаков Add-Del для поиска наиболее оптимального сочетания признаков для обучения модели, получили метрики: F1:Macro=0.99 и accuracy=98.8%.
2. Найдено оптимальное сочетание признаков, которое достаточно для качественного обучения моделей машинного обучения, но при этом содержит большую часть признаков, понятных среднестатистическому пользователю.
3. Попробовали разные модели для поиска потенциально опасных водителей, при этом результаты довольно сильно совпадают при сравнении наиболее опасных водителей. Также показали, что использование данных после уменьшения размерности исходных данных, не сильно влияет на итоговые предсказания модели.
4. Предложили и реализовали онлайн модель профиля водителя, для которой было необходимо реализовать пересечение n-мерных сфер, а также проконтролировали сходимость алгоритма с помощью метрики FDR. Сравнили полученные результаты в 3 и 4 параграфе, они согласуются друг с другом. При этом полученная модель выдает метрику, которую можно использовать при проведедении A/B тестов

Дальнейшие перспективы продолжения исследований:

1. Ключевым ограничением являлось отсутствие возможности использовать асессорскую разметку. В случае наличия асессорской разметки открывается горизонт возможностей, а именно:
    (a) Возможность реализовать поиск потенциально опасных водителей с помощью обучения с учителем, а также контролировать с большей точностью качество уже реализованных моделей поиска потенциально опасных водителей.

   (b) Использования различных нейронных моделей, для которых необходимо разметить большое количество данных, с другой стороны появляется возможность использовать как базовые модели, такие как lstm и cnn-1d, так и более современные и мощные решения. Дальнейший выбор модели зависит от качества разметки, а также технических возможностях для обучения модели.

2. При работе онлайн модели профиля водителя в продовом процессе с большим количеством данных необходимо будет настроить частоту обновления профилей водителя, что несомненно является отдельной сложной задачей, где необходимо будет найти компромисс между частотой обновлений, ресурсами и необходимостью в обновлении профиля для конкретных водителей.

Список литературы

1. Санкции водителям за регулярное превышение скорости. — URL: https://vc.ru/transport/52001-yandeks-taksi-budet-nakazyvat-voditeley-za-regulyarnoe-prevyshenie-skorosti (дата обр. 17.06.2023).
2. Профиль вождения. — URL: https://yandex.ru/support/drive/about/safe-driver.html (дата обр. 17.06.2023).
3. What are the driving modes how do they work? [электронный ресурс]. — 2019. — URL: https://carbiketech.com/driving-modes/ (visited on 06/17/2023).
4. Driving Dataset. Hacking and Countermeasure Research Lab. — URL: https://ocslab.hksecurity.net/Datasets/driving-dataset (visited on 11/13/2022).
5. *Волков Н. А.* Лекции по машинному обучению, 7 семестр. — 10.2022.
6. RandomForestClassifier [электронный ресурс]. — 2007. — URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 06/17/2023).
7. *Е. Е., К. Л.* Градиентный бустинг [электронный ресурс]. — URL: https://academy.yandex.ru/handbook/ml/article/gradientnyj-busting (дата обр. 17.06.2023).
8. Как работает модерация в Авито? [электронный ресурс]. — 2019. — URL: https://vc.ru/avito/71145-moderation (дата обр. 17.06.2023).
9. Driver profiling: The pathway to deeper personalization [электронный ресурс]. — 2022. — URL: https://www.sciencedirect.com/science/article/pii/S1319157822003160 (visited on 06/17/2023).
10. *Li S.* Concise formulas for the area and volume of a hyperspherical cap. — 2011.

Список рисунков

Список таблиц

Приложение А

Применение PCA

Визуализации данных в двумерном пространстве (PCA):

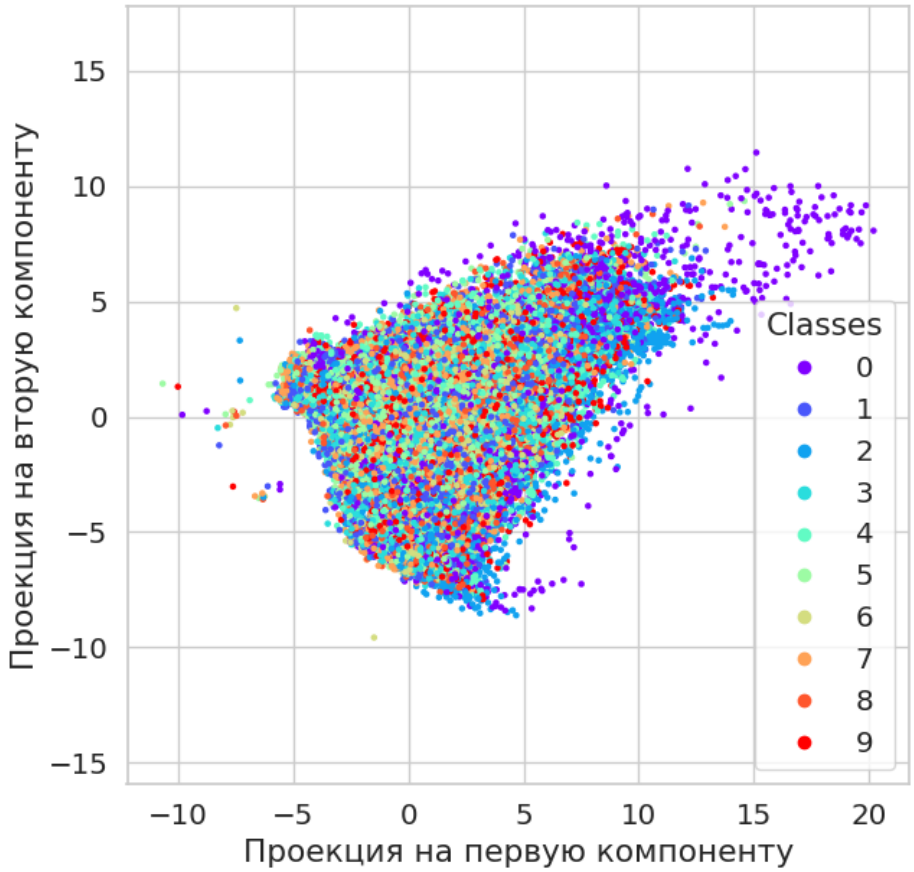


Рисунок А.1 — Визуализация PCA

Приложение Б

Применение UMAP

Визуализации данных в двумерном пространстве (UMAP):

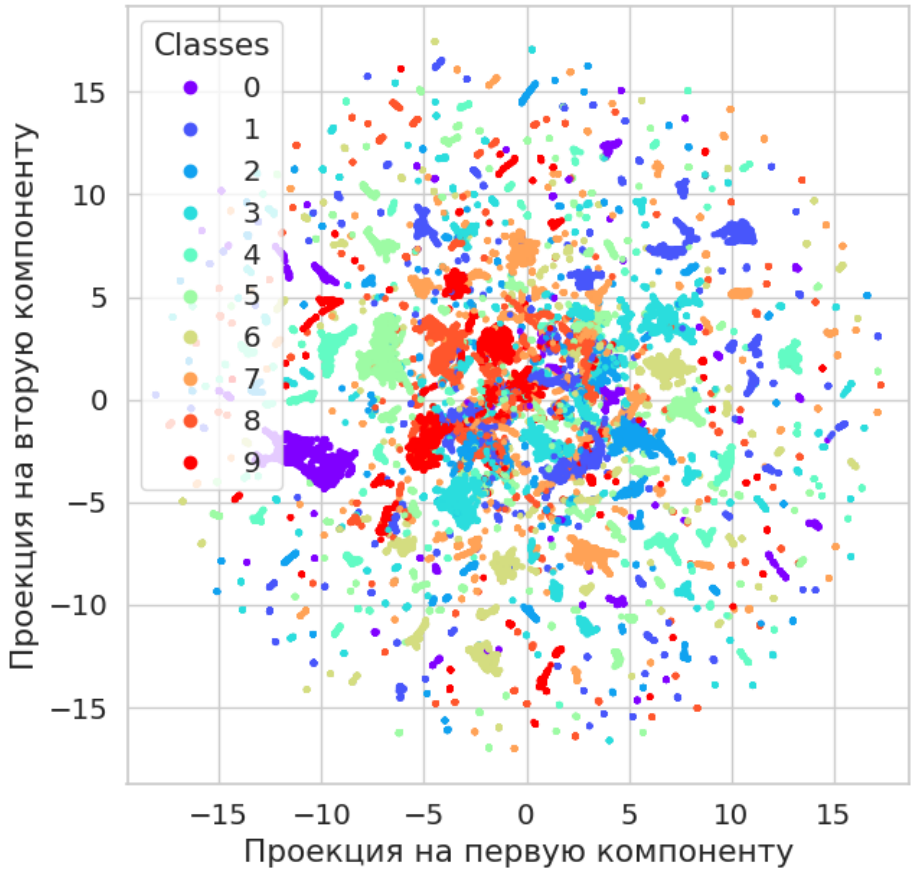


Рисунок Б.1 — Визуализация UMAP

Приложение В

Применение LDA

Визуализации данных в двумерном пространстве (LDA):

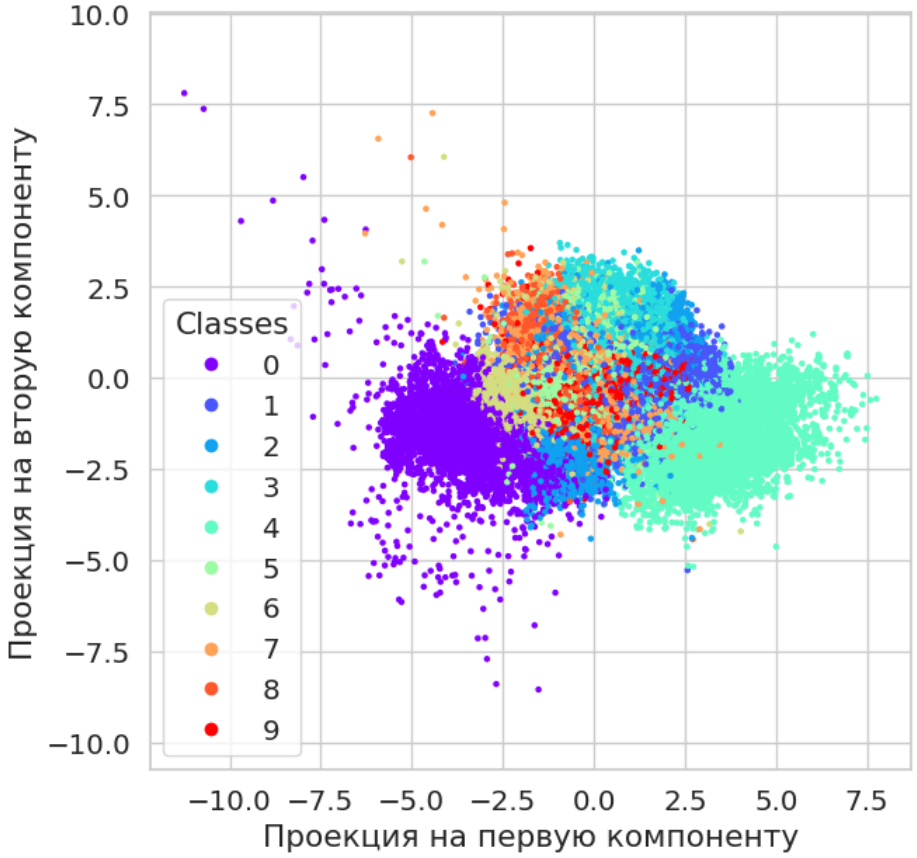


Рисунок В.1 — Визуализация LDA