Анализ обучения на данных разными методами

Лузгов Тимур

Преподаватель:
Капранов Иван Константинович

5 мая 2024 г.

Содержание

1 Введение

В данном проекте мы будем обучаться на датасете [1]. Датасет из себя представляет характеристики телефона и наша цель определит в какой ценовой категории находится устройство. Для этого воспользуемся 3 различнины-ми методами машинного обучения. Сравним точность определения категории и определим какой метод работает лучше всего на нашем датасете.

2 Описание метода логической регрессии

Из литературы [4] логистическая регрессия - это статистический метод анализа, используемый для прогнозирования вероятности возникновения некоторого события путем сопоставления набора независимых переменных. Этот метод часто используется в машинном обучении для задач классификации.

1. **Подготовка данных**: Собранные данные подготавливаются для анализа. Это может включать в себя очистку данных, обработку пропущенных значений и масштабирование переменных.

2. **Выбор независимых переменных**: Выбираются переменные, которые могут влиять на целевую переменную (событие, которое мы пытаемся предсказать).

3. **Построение модели**: Строится математическая модель, которая связывает независимые переменные с вероятностью возникновения целевой переменной. В случае логистической регрессии используется логистическая функция для этого.

4. **Обучение модели**: Модель обучается на тренировочных данных, чтобы определить коэффициенты, наилучшим образом соответствующие данным.

5. **Оценка модели**: После обучения модели ее необходимо оценить на тестовых данных, чтобы определить ее точность и эффективность.

6. **Использование модели для предсказаний**: После успешной оценки модели ее можно использовать для предсказания вероятности возникновения события для новых наблюдений.

3 Описание метода решающих деревьев

Из литературы [2] решающие деревья - это графический метод принятия решений в виде древовидной структуры. Они используются для прогнозирования или классификации данных на основе набора правил, выводимых из обучающих данных.

1. **Построение дерева**: Алгоритм строит дерево, разбивая данные на подгруппы по определенным критериям, чтобы максимизировать однородность внутри каждой подгруппы и минимизировать неоднородность между ними.

2. **Выбор критериев разбиения**: Для разбиения узла на подузлы выбираются различные критерии, такие как индекс Джини или энтропия, которые оценивают неоднородность данных.

3. **Подрезка дерева**: После построения дерева его можно подрезать для уменьшения переобучения и улучшения обобщающей способности модели.

4. **Прогнозирование и классификация**: После построения модели решающего дерева она может использоваться для прогнозирования значений для новых данных или классификации объектов в соответствующие категории.

4 Описание метода ближайшего соседа

Из литературы [3] Метод ближайшего соседа (k-nearest neighbors, k-NN) - это простой и интуитивно понятный алгоритм машинного обучения, используемый для классификации и регрессии. Принцип работы заключается в том, что объект классифицируется на основе классов его ближайших соседей из обучающего набора данных.

1. **Хранение обучающего набора**: Вся информация из обучающего набора данных сохраняется в памяти. Это включает в себя объекты (векторы признаков) и соответствующие им метки классов.

2. **Выбор числа соседей (k)**: Необходимо определить количество ближайших соседей, которые будут использоваться для классификации нового объекта. Это число k может быть задано заранее или выбрано на основе кросс-валидации.

3. **Определение ближайших соседей**: Для классификации нового объекта вычисляется расстояние (чаще всего используется евклидово расстояние) между ним и всеми объектами обучающего набора. Затем выбираются k объектов с наименьшим расстоянием до нового объекта.

4. **Прогнозирование класса**: Классификация нового объекта осуществляется путем голосования среди его k ближайших соседей. То есть, класс, к которому принадлежит большинство из этих соседей, считается классом нового объекта.

Метод ближайшего соседа прост в реализации и легко адаптируется к различным типам данных. Однако, его эффективность может сильно зависеть от выбора метрики расстояния и числа соседей.

5 Описание датасета

- **battery_power**: Общая емкость батареи (mAh)
- **blue**: Наличие Bluetooth на устройстве
- **clock_speed**: Скорость выполнения инструкций микропроцессором
- **dual_sim**: Наличие двух SIM-карт в устройстве
- **fc**: Качество фронтальной камеры в мегапикселях
- **four_g**: Наличие сети 4G на устройстве
- **int_memory**: Внутренняя память устройства в гигабайтах
- **m_dep**: Глубина устройства в сантиметрах
- **mobile_wt**: Вес устройства
- **n_cores**: Количество ядер процессора
- **pc**: Качество основной камеры в мегапикселях
- **px_height**: Высота разрешения экрана в пикселях
- **px_width**: Ширина разрешения экрана в пикселях
- **ram**: Оперативная память (RAM) в мегабайтах
- **sc_h**: Высота экрана устройства в сантиметрах
- **sc_w**: Ширина экрана устройства в сантиметрах
- **talk_time**: Максимальное время разговора при полностью заряженной батарее
- **three_g**: Наличие сети 3G на устройстве
- **touch_screen**: Наличие сенсорного экрана на устройстве
- **wifi**: Наличие Wi-Fi на устройстве
- **price_range**: Категоризированная цена устройства

6 Ход работы

Для начала проверим наш датасет, есть ли в нём незаполненные ячейки или ячейки с неправильными значениями. При их нахождении мы либо удаляем устройство полностью, либо заполняем их средними значениями, модами или медианами. В нашем случае, все данные были корректны.

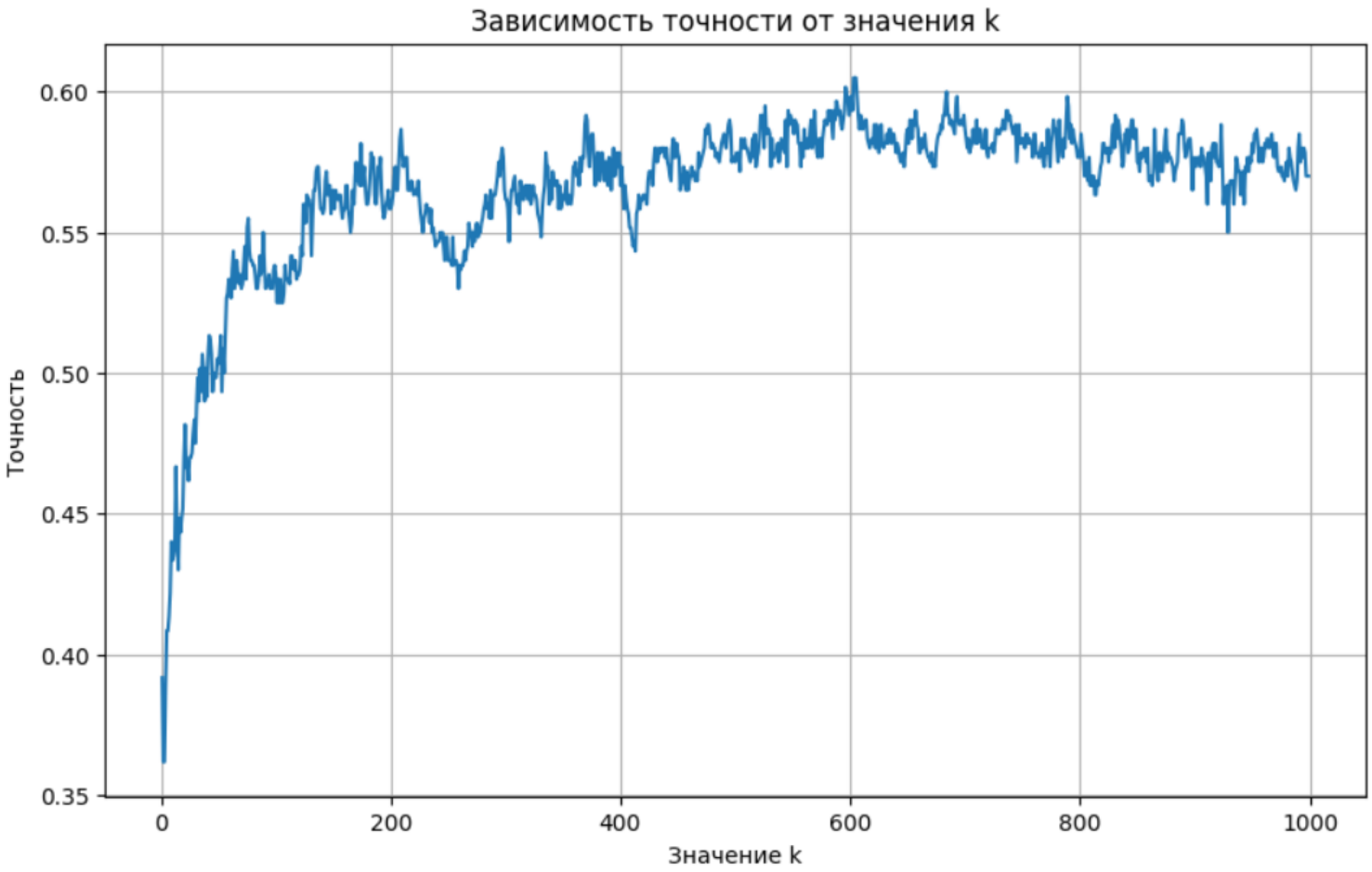
Далее нормируем все наши значения с помощью метода MinMaxScaler. Это поможет нам привести все признаки к одному масштабу и улучшить производительность наших моделей.

Определим категориальные признаки на основе количества уникальных значений и преобразуем их с помощью LabelEncoder. Это позволит нам преобразовать текстовые или категориальные признаки в числовые значения, которые могут быть использованы алгоритмами машинного обучения.

После этого разделим наши данные на обучающий и тестовый наборы. Обучим соответствующие модели и сравним их значения. Для улучшения результатов в решающих деревьях проверим значения на нескольких максимальных глубинах, а в методе ближайших соседей будем перебирать k, где k - гиперпараметр, определяющий количество соседей, используемых для классификации нового объекта. Для логистической регрессии будем перебирать максимальное количество итераций.

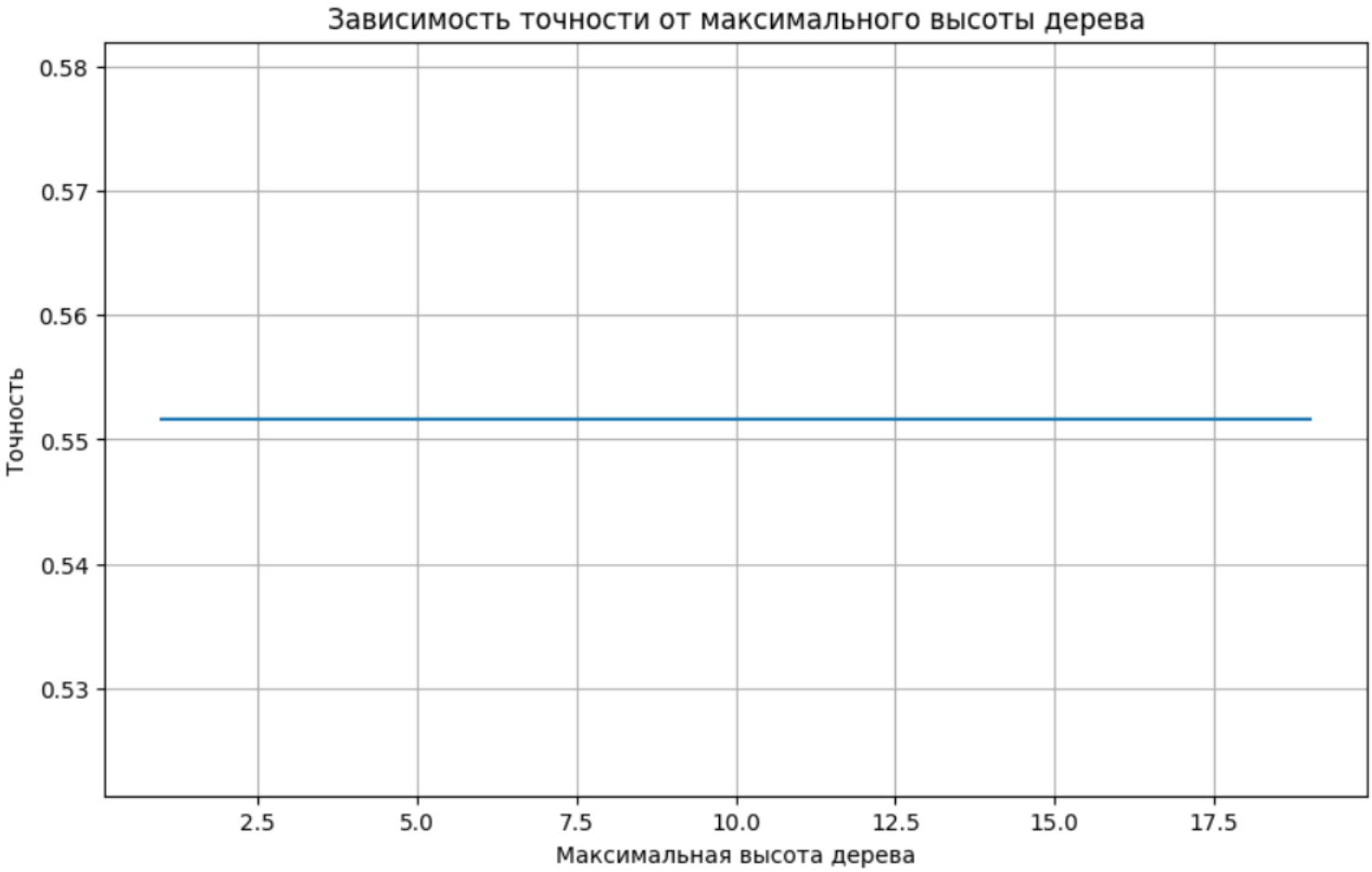
7 Заключение

Из графика зависимости точности от количесва соседей, видно что в области k = 600 достигается максимум, а после график начинает убывать



Из графика зависимости от максимального количества итераций к точности. Можно заметить, что до 43 итераций он увеличивается, а после является константой.

Из графика зависимости от максимальной высоты дерева к точности. Можно заметить, что оно не зависит от высоты дерева.



В итоге получаем, что:

- В методе ближайших соседей точность достигает 0.605, при гиперпараметре равном 603.
- В методе логистической регрессии точность достигает 0.911, при гиперпараметре равном 37.
- В методе решающих деревьев точность достигает 0.551 и остается постоянной при любом значении гиперпараметра.

В заключение можно сказать, что метод логистической регрессии лучше всего справился с нашими данными.

Список литературы

[1] Dataset cell phone price, 2024. Доступно на: `https://www.kaggle.com/datasets/atefehmirnaseri/cell-phone-price`.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[3] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on Information Theory*, 13(1):21–27, 1967.

[4] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.