Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

Факультет *информационных технологий и компьютерных систем*

Кафедра *«Автоматизированные системы обработки информации и управления»*

КУРСОВОЙ ПРОЕКТ

по дисциплине ***Динамические языки программирования***

на тему ***Исследование методов машинного обучения***

Пояснительная записка

Шифр проекта **043-КП-09.03.04-№08-ПЗ**

Студента (ки) *Киселевой Майи Андреевной*
фамилия, имя, отчество полностью

Курс ***4*** Группа ***ПИН-201***

Направление (специальность) 09.03.04 – Программная инженерия
код, наименование

Руководитель ***ст.пр***
ученая степень, звание

Кабанов.А.А
фамилия, инициалы

Выполнил (а) _____
дата, подпись студента (ки)

К защите _____
дата, подпись руководителя

Проект (работа) защищен (а) с оценкой _____

Набранные баллы: _____ _____ _____
в семестре на защите Итого

Омск 2024

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

Задача 1. Найти набор данных (датасет) для классификации удовлетворяющий следующим условиям:

- более 10 000 строк,
- более 20 столбцов,
- разные типы в столбцах,
- обязательно наличие целевого признака (таргета).

Задача 2. Провести классификацию найденного датасета, методом к-ближайших соседей. В формате Markdown писать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

Задача 3. Провести классификацию найденного датасета методом машины опорных векторов. В формате Markdown записать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

Задача 4. Провести классификацию найденного датасета методами линеной и логистической регрессий. В формате Markdown записать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

Задача 5. Провести классификацию найденного датасета методами наивного Байеса. В формате Markdown записать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

Задача 6. Провести классификацию найденного датасета методами решающего дерева и случайного леса. В формате Markdown записать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

Задача 7. Провести классификацию найденного датасета методами CatBoost. В формате Markdown записать пояснения. Объяснить, почему были выбраны именно такие гиперпараметры, была ли перекрёстная проверка, и т.д.

РЕФЕРАТ

Пояснительная записка к курсовому проекту.

МАШИННОЕ ОБУЧЕНИЕ, КЛАССИФИКАЦИЯ, CAT BOOST, ТОЧНОСТЬ, ВРЕМЯ ОБУЧЕНИЯ, МЕТОДЫ НАИВНОГО БАЙЕСА, МЕТОД К-БЛИЖАЙШИХ СОСЕДЕЙ, МЕТОД ЛИНЕЙНОЙ И ЛОГИСТИЧЕСКОЙ РЕГРЕСИИ, МЕТОД ОПОРНЫХ ВЕКТОРОВ

Предметная область исследования курсового проекта - "Исследование методов машинного обучения".

Основная цель данного исследования - разработка, изучение и оформление требований к использованию методов машинного обучения в разнообразных сферах.

В ходе выполнения проекта были осуществлены следующие этапы:

- определение цели,
- выделение предметной области,
- проведение анализа существующих методов машинного обучения,
- разработка и реализация этих методов,
- проведение сравнительного анализа с целью выбора наилучшего метода машинного обучения.

Был проведён анализ основных методов машинного обучения, таких как:

- метод к-ближайших соседей,
- метод машины опорных векторов,
- линейная регрессия,
- логистическая регрессия,
- наивный Байес,
- решающие деревья,
- случайные леса,
- CatBoost.

Каждый из методов был подвергнут оценке по точности и времени поиска лучших параметров.

Оглавление

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В данном курсовом проекте применяют следующие термины и определения:

1. Машинное обучение - область исследования, где компьютерные системы автоматически обучаются на основе данных и алгоритмов, без явного программирования.
2. Метод к-ближайших соседей - алгоритм классификации, основанный на поиске k ближайших объектов обучающей выборки к новому объекту и определении его класса на основе классов ближайших соседей.
3. Машина опорных векторов - алгоритм машинного обучения, строящий гиперплоскость в пространстве признаков для максимального разделения объектов разных классов.
4. Логистическая регрессия: метод машинного обучения для бинарной и многоклассовой классификации, использующий логистическую функцию для предсказания вероятности принадлежности к определенному классу.
5. Наивный байес с нормальным распределением (GaussianNB) - Байесовский классификатор, основанный на предположении о нормальном (гауссовом) распределении признаков.
6. Решающие деревья - алгоритмы машинного обучения, строящие структуру дерева для принятия решений на основе признаков объектов.
7. Случайный лес - ансамбль решающих деревьев, объединенных для улучшения обобщающей способности и уменьшения переобучения.
8. CatBoost - высокопроизводительная библиотека градиентного бустинга, специально разработанная для эффективной работы с категориальными признаками в данных.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В данном курсовом проекте применяют следующие сокращения и обозначения:

1. МО - Машинное обучение
2. k-NN - Метод к-ближайших соседей
3. SVM - Машина опорных векторов

ВВЕДЕНИЕ

В современном информационном обществе важность и применение методов машинного обучения в различных областях знаний становятся всё более значимыми. Расширение доступа к данным и возможность эффективно их анализировать и использовать в различных задачах делают машинное обучение ключевым инструментом для прогнозирования, классификации и обработки информации.

Цель курсовой работы - провести анализ и сравнительный обзор различных моделей машинного обучения на примере конкретного набора данных. Работа включает в себя исследование и оценку производительности таких моделей, как *k*-ближайших соседей, метода опорных векторов, логистической регрессии, наивного байесовского классификатора, модели решающего дерева, случайного леса и CatBoost, а также сопоставление их результатов.

Работа включает в себя сравнительный анализ преимуществ и недостатков каждой модели, а также итоговый вывод о наиболее подходящей модели для конкретного датасета. Полученные результаты могут быть использованы в различных областях, требующих прогнозирования или классификации на основе имеющихся данных.

ОПИСАНИЕ ДАТАСЕТА

Представленный набор данных информацию о кибератаках, включая детали атак, предупреждения, действия, протоколы и информацию об устройствах и пользовательской активности:

1. Timestamp: Временная метка события.
2. Source IP Address: IP-адрес отправителя.
3. Destination IP Address: IP-адрес получателя.
4. Source Port: Порт отправителя.
5. Destination Port: Порт получателя.
6. Protocol: Используемый протокол (например, TCP, UDP).
7. Packet Length: Длина пакета.
8. Packet Type: Тип пакета (например, запрос, ответ).
9. Traffic Type: Тип трафика (например, HTTP, DNS).
10. Payload Data: Данные пакета.
11. Malware Indicators: Признаки наличия вредоносного ПО.
12. Anomaly Scores: Оценка аномальности события.
13. Alerts/Warnings: Предупреждения и оповещения.
14. Attack Type: Тип атаки.
15. Attack Signature: Сигнатура атаки.
16. Action Taken: Предпринятые меры по атаке.
17. Severity Level: Уровень серьезности инцидента.
18. User Information: Информация о пользователе.
19. Device Information: Информация об устройстве.
20. Network Segment: Сегмент сети.
21. Geo-location Data: Географические данные.
22. Proxy Information: Информация о прокси.
23. Firewall Logs: Логи брандмауэра.
24. IDS/IPS Alerts: Оповещения системы обнаружения вторжений/предотвращения вторжений.
25. Log Source: Источник журнала или логов.

Этот датасет включает в себя 25 различных метрик и содержит 40,000 записей.

ОБРАБОТКА ДАТАСЕТА

Перед сравнением методов машинного обучения, для каждого метода проводится обработка датасета, следующими способами:

1. Удаление строк с отсутствующими значениями
2. Кодирование текстовых данных:

   label_encoder = LabelEncoder()

   Dataset[‘Attack Type Encoded’] = label_encoder.fit_transform(Dataset[‘Attack Type’])

	Attack Type	Attack Type Encoded
0	Malware	2
1	Malware	2
2	DDoS	0
3	Malware	2
4	DDoS	0
5	Malware	2
6	DDoS	0
7	Intrusion	1
8	Intrusion	1
9	Malware	2

Рис.1 – Результат кодирования

3. Подсчёт уникальных классов ‘Attack Type Encoded’
4. Удаление текстовых значений и заполнение пропусков:

```
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Source Port          9404 non-null   int64
 1   Destination Port     9404 non-null   int64
 2   Packet Length        9404 non-null   int64
 3   Anomaly Scores       9404 non-null   float64
 4   Attack Type Encoded  9404 non-null   int64
dtypes: float64(1), int64(4)
memory usage: 367.5 KB
```

Рис.2 – Результат очистки нетекстовых значений

5. Определение и удаление пропущенных значений:

Выводится процентное соотношение пропущенных значений для каждого столбца. Пропущенные значения удаляются, обеспечивая чистоту данных для дальнейшего использования.

ИТОГОВЫЕ ДАННЫЕ

Обработанные обучающие данные, прошедшие все этапы предварительной очистки, готовы для внедрения в модели машинного обучения. Теперь датасет представляет собой сбалансированную обучающую выборку, готовую для последующего обучения моделей.

1. МЕТОД K-БЛИЖАЙШИХ СОСЕДЕЙ

Метод k-ближайших соседей представляет собой метрический алгоритм, применяемый для автоматической классификации объектов или выполнения регрессии. При использовании данного метода для классификации, каждый объект получает присвоенный ему класс, который определяется на основе наиболее часто встречающегося класса среди k ближайших соседей данного элемента. В данном контексте, параметр k представляет собой заданное количество соседей, классы которых предварительно известны.

Гиперпараметры метода k-ближайших соседей:

- k (число соседей):

  Одним из главных гиперпараметров является количество соседей (k), которое определяет, на сколько далеко в пространстве признаков смотрит алгоритм.
- Метрика расстояния (евклидова, манхэттанская, чебышева, минковски).

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- Количество соседей – от 3 до 40
- Метрики (евклидова, манхэтанская, чебышева, минковски)

Поиск лучших параметров занял 90.05 секунд. После поиска лучших гиперпараметров были получены следующие результаты:

- Количество соседей – 3
- Лучшая метрика – manhattan

На основе лучших параметром была получен следующая точность:

```
Точность модели с 37 соседями и метрикой euclidean: 0.3272195640616693
              precision    recall  f1-score   support

           0       0.34      0.45      0.38      1267
           1       0.33      0.31      0.32      1286
           2       0.31      0.22      0.26      1209

    accuracy                           0.33      3762
   macro avg       0.32      0.33      0.32      3762
weighted avg       0.32      0.33      0.32      3762
```

Рис.4 – Обучение на лучших параметрах.

На основе этого можно сделать вывод, что при 3 соседях и метрике manhattan была получена точность ~0.3328.

2. МЕТОД МАШИНЫ ОПОРНЫХ ВЕКТОРОВ

Метод машины опорных векторов представляет собой алгоритм

машинного обучения, применяемый для решения задач классификации и регрессии. Суть метода заключается в поиске оптимальной гиперплоскости в многомерном пространстве признаков, которая наилучшим образом разделяет объекты разных классов. Основной стратегией метода SVM является максимизация расстояния между гиперплоскостью и объектами различных классов, что придает ему эффективность даже в случаях с нелинейными разделяющими поверхностями.

Гиперпараметры:

- Ядро (Kernel):

  Выбор ядра влияет на способность модели обрабатывать сложные нелинейные зависимости в данных. Часто используются линейное (linear), полиномиальное (poly) и радиально-базисное (rbf) ядра, сигмоид (sigmoid).

- Параметр регуляризации (C):

  Параметр C контролирует уровень штрафа за неверную классификацию. Большие значения C приводят к более жесткой классификации, а малые значения позволяют допустить некоторые ошибки.

- Параметры ядра (Kernel-specific parameters):

  Некоторые ядра, такие как полиномиальное, могут иметь дополнительные параметры, такие как степень полинома (degree) и коэффициенты.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- Ядра (kernel) – linear, rbf, sigmoid, poly
- C (параметр регуляризации): 0.001, 0.01, 0.1, 1, 10
- Степень полинома (degree): 1, 2, 3, 4, 5, 6,

Поиск лучших параметров занял 1094.3 секунды.

После поиска лучших гиперпараметров были получены следующие результаты:

- Лучшее ядро – poly
- Лучший параметр регуляризации – 0.1
- Степень полинома - 6

На основе лучших параметров была полученная следующая точность:

```
Точность модели: 0.3301875
              precision    recall  f1-score   support

           0       0.33      0.90      0.48      5306
           1       0.31      0.02      0.04      5416
           2       0.33      0.07      0.12      5278

    accuracy                           0.33     16000
   macro avg       0.32      0.33      0.21     16000
weighted avg       0.32      0.33      0.21     16000

Средняя точность перекрестной проверки: 0.33487500000000003
```

Рис. 5 – Обучение на лучших параметрах.

На основе этого можно сделать вывод, что при ядре poly, параметре регуляризации 0.1 и степени полинома 6 получена точность ~0.3348.

3. МЕТОД ЛИНЕЙНОЙ И ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Линейная регрессия представляет собой метод, который применяется не только для классификации, но также для прогнозирования вероятности принадлежности к определенному классу.

Гиперпараметры:

- fit_intercept (включение свободного члена):

  Определяет, следует ли включать свободный член в уравнение регрессии. Если установлен в False, модель проходит через начало координат.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- fit_intercept: true

Поиск лучших параметров занял 0.05 секунды.

После поиска лучших гиперпараметров были получены следующие результаты:

- fit_intercept: true

На основе лучших параметров была полученая следующая точность:

	precision	recall	f1-score	support
0	1.00	0.00	0.00	5224
1	0.33	1.00	0.50	5253
2	1.00	0.00	0.00	5253
accuracy			0.33	15730
macro avg	0.78	0.33	0.17	15730
weighted avg	0.78	0.33	0.17	15730

Рис.6 – Обучение на лучших параметрах

На основе этого можно сделать вывод, что при fit_intercept: true точность 0.33

Логистическая регрессия представляет собой метод классификации, применяемый для решения задач бинарной или многоклассовой классификации. Она базируется на использовании логистической функции, которая преобразует линейную комбинацию признаков в вероятность принадлежности к определенному классу.

Гиперпараметры:

- penalty (штраф):

  Определяет тип регуляризации, который применяется к модели. Может быть 'l1', 'l2', 'elasticnet' или 'none'.

- C (обратная сила регуляризации):

  Обратная сила регуляризации, которая контролирует величину штрафа за сложность модели. Меньшие значения C увеличивают штраф и могут привести к более простым моделям.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- penalty (штраф): l1, l2
- C (обратная сила регуляризации): 0.001, 0.01, 0.1, 1, 10

Поиск лучших параметров занял 3.59 секунд.

После поиска лучших гиперпараметров были получены следующие результаты:

- penalty (штраф): l1
- C (обратная сила регуляризации): 0.01

На основе лучших параметров была полученая следующая точность:

```
Лучшие параметры для логистической регрессии: {'C': 0.01, 'penalty': 'l1'}
Отчет о классификации для логистической регрессии:
              precision    recall  f1-score   support

           0       0.33      0.67      0.45      5224
           1       0.33      0.10      0.15      5253
           2       0.34      0.23      0.27      5253

    accuracy                           0.33     15730
   macro avg       0.33      0.33      0.29     15730
weighted avg       0.33      0.33      0.29     15730
```

Рис.7 – Обучение на лучших параметрах

На основе этого можно сделать вывод, что при penalty: l1 и C:0.01, точность равна 0.33

4. МЕТОДЫ НАИВНОГО БАЙЕСА

Классификатор наивного Байеса — это группа алгоритмов, построенных на принципе байесовской классификации с предположением о независимости между признаками (наивное предположение).

Основные методы наивного байеса и их гиперпараметры:

- Наивный байес с распределением Бернулли (BernoulliNB):

  Гиперпараметры:

  alpha: Параметр аддитивного сглаживания (Laplace smoothing) для предотвращения проблемы с нулевыми вероятностями. Может принимать значения от 0 до 1.

  binarize: параметр задает порог для бинаризации входных данных. Все значения признаков, большие этого порога, становятся 1, а все значения, меньшие или равные порогу, становятся 0.

- Наивный байес с мультиномиальным распределением (MultinomialNB):

  Гиперпараметры:

  alpha: Параметр аддитивного сглаживания (Laplace smoothing) для предотвращения проблемы с нулевыми вероятностями. Может принимать значения от 0 до 1.

- Наивный байес с нормальным распределением (GaussianNB):

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- BernoulliNB: alpha: 0.1, 0.5, 1.0; binarize: 0.0, 0.1, 0.2
- MultinomialNB: aplha: 0.1, 0.5, 1.0
- GaussianNB

Поиск лучших параметров занял 0.41 секунды.

После поиска лучших гиперпараметров были получены следующие результаты:

- Лучшая модель: GaussianNB

На основе лучших параметров была полученая следующая точность:

```
Точность модели: 0.3277813095994914
              precision    recall  f1-score   support

           0       0.32      0.32      0.32      5224
           1       0.32      0.26      0.29      5253
           2       0.33      0.40      0.37      5253

    accuracy                           0.33     15730
   macro avg       0.33      0.33      0.32     15730
weighted avg       0.33      0.33      0.32     15730
```

Рис.8 – Обучение на лучших параметрах

На основе этого можно сделать вывод, что при GaussianNB точность ~0.3277

5. МЕТОДЫ РЕШАЮЩЕГО ДЕРЕВА И СЛУЧАЙНОГО ЛЕСА

Модель решающего дерева представляет собой метод машинного обучения, основанный на структуре древа для принятия решений. Каждый узел в дереве представляет собой решение, связанное с определенным признаком, а каждый лист дерева представляет собой окончательный ответ в виде класса или значения.

Гиперпараметры:

- Max Depth (максимальная глубина):
  Ограничивает максимальную глубину дерева. Это помогает предотвратить переобучение.
- Min Samples Split (минимальное количество объектов для разделения):
  Устанавливает минимальное количество образцов, необходимых для разделения внутреннего узла.
- Min Samples Leaf (минимальное количество объектов в листе):
  Определяет минимальное количество образцов, которые должны находиться в листе.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- Max Depth: None, 5, 10, 15, 20
- Min Samples Split: 2, 5, 10
- Min Samples Leaf: 1, 2, 4

Поиск лучших параметров занял 16.85 секунд.

После поиска лучших гиперпараметров были получены следующие результаты:

- Max Depth: 20
- Min Samples Split: 2
- Min Samples Leaf: 5

На основе лучших параметров была полученая следующая точность:

```
Decision Tree с лучшими параметрами - Точность на тесте: 0.3355625
Classification Report для Decision Tree:
              precision    recall  f1-score   support

           0       0.33      0.40      0.36      5306
           1       0.34      0.35      0.34      5416
           2       0.35      0.26      0.30      5278

    accuracy                           0.34     16000
   macro avg       0.34      0.34      0.33     16000
weighted avg       0.34      0.34      0.33     16000
```

Рис.9 – Обучение на лучших параметрах

На основе этого можно сделать вывод, что при Max Depth: 5, Min Samples Split: 4, Min Samples Leaf: 2 точность ~0.336

Случайный лес представляет собой совокупность деревьев решений, где каждое дерево обучается на случайной подвыборке данных, а затем прогнозы объединяются или усредняются для формирования окончательного результата.

Гиперпараметры:

- N Estimators (количество деревьев):
  Определяет количество деревьев в ансамбле.
- Max Depth (максимальная глубина деревьев):
  Ограничивает максимальную глубину каждого дерева в ансамбле.
- Min Samples Split (минимальное количество объектов для разделения):
  Определяет минимальное количество образцов, необходимых для разделения внутреннего узла.
- Min Samples Leaf (минимальное количество объектов в листе):
  Определяет минимальное количество образцов, которые должны находиться в листе.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- Max Depth: None, 5, 10, 15, 20
- N Estimators: 50, 100, 150, 200
- Min Samples Split: 2, 5, 10
- Min Samples Leaf: 1, 2, 4

Поиск лучших параметров занял 2955.47 секунд.

После поиска лучших гиперпараметров были получены следующие результаты:

- Max Depth: 5
- Min Samples Split: 5
- Min Samples Leaf: 4
- N Estimators: 50

На основе лучших параметров была полученая следующая точность:

```
Random Forest с лучшими параметрами - Точность на тесте: 0.3345625
Classification Report для Random Forest:
              precision    recall  f1-score   support

           0       0.33      0.38      0.35      5306
           1       0.34      0.31      0.32      5416
           2       0.33      0.32      0.32      5278

    accuracy                           0.33     16000
   macro avg       0.33      0.33      0.33     16000
weighted avg       0.33      0.33      0.33     16000
```

Рис.10 – Обучение на лучших параметрах

На основе этого можно сделать вывод, что при Max Depth: 5, Min Samples Split: 5, Min Samples Leaf: 4, N Estimators: 50 точность ~0.3345

6. МЕТОД CATBOOST

CatBoost представляет собой метод машинного обучения, который формирует прогностическую модель в виде ансамбля слабых моделей, чаще всего деревьев решений. Этот метод пошагово создает модель, что позволяет оптимизировать произвольную дифференцируемую функцию потерь.

Гиперпараметры:

- Iterations (количество итераций):

  Определяет количество базовых моделей (деревьев), которые будут обучены в ансамбле.
- Learning Rate (скорость обучения):

  Контролирует величину шага градиентного спуска при обновлении весов.
- Depth (глубина деревьев):

  Определяет максимальную глубину деревьев.

В приведенном методы были использование следующие гиперпараметры для поиска лучших параметров:

- Iterations: 100, 200, 300
- Learning Rate: 0.01, 0.05, 0.1
- Depth: 4, 6, 8

Поиск лучших параметров занял 197.5 секунды.

После поиска лучших гиперпараметров были получены следующие результаты:

- Iterations: 200
- Learning Rate: 0.05
- Depth: 4

На основе лучших параметров была полученая следующая точность:

```
Точность CatBoost на тестовых данных: 0.32875
Classification Report для CatBoost:
              precision    recall  f1-score   support

           0       0.33      0.40      0.36      5306
           1       0.34      0.22      0.26      5416
           2       0.33      0.37      0.35      5278

    accuracy                           0.33     16000
   macro avg       0.33      0.33      0.32     16000
weighted avg       0.33      0.33      0.32     16000
```

Рис. 11 – Обучение на лучших параметрах.

На основе этого можно сделать вывод, что при Iterations: 200, Learning Rate: 0.05, Depth: 4 точность ~0.3287

7. СРАВНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

В приведенной таблице 1 показано сравнение всех используемых методом машинного обучения.

Таблица 1 – Сравнение методов машинного обучения

Название	**Точность (%)**	**Время поиска лучших параметров (сек)**
Методом к- ближайших соседей	33.28	90.05
Метод машины опорных векторов	33.48	1094.33
Метод линейной регресии	33	0.05
Метод логистической регрессии	33	3.59
Метод наивный байес с нормальным распределением (GaussianNB)	33.33	0.41
Метод решающего дерева	33.6	16.85
Метод случайного леса	33.18	2955.96
Метод CatBoost	32.87	197.5

Среди всех методов выделяются метод решающего древа с точностью около 33.6% и временем выполнения ~16.85 секунд и метод линейной регрессии с точностью около 33% и временем выполнения ~0,05 секунды. Более точные результаты получатся решающего древа.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта был подготовлен набор данных, после чего проведены эксперименты с различными методами машинного обучения для решения задачи классификации. В рамках анализа моделей оценивались их точность и время, затраченное на оптимизацию параметров. В результате выделяются два метода, привлекающих внимание. Метод решающего древа проявил себя с точностью около 33.6% и затратами времени на выполнение порядка 16.85 секунд. Также стоит отметить метод линейной регрессии, демонстрирующий точность на уровне 33% и невысокое время выполнения, составляющее приблизительно 0.05 секунды. Для более точных результатов следует выбрать метод решающего древа.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Проглиб. Метод k-ближайших соседей (k-nearest neighbour) – URL: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19
2. Loginom. Классификация данных методом k-ближайших соседей – URL: https://loginom.ru/blog/knn
3. Habr. Классификация данных методом опорных векторов – URL: https://habr.com/ru/articles/105220/
4. ИТМО. Метод классификации CatBoost – URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost
5. Habr. Как легко понять логистическую регрессию – URL: https://habr.com/ru/companies/io/articles/265007/
6. Scikit-learn. Наивные методы Байеса – URL: https://scikit-learn.ru/1-9-naive-bayes/
7. Проглиб. Наивный байесовский алгоритм классификации: преимущества и недостатки – URL: https://proglib.io/p/izuchaem-naivnyy-bayesovskiy-algoritm-klassifikacii-dlya-mashinnogo-obucheniya-2021-11-12
8. Проглиб. Машинное обучение для начинающих: алгоритм случайного леса (Random Forest) – URL: https://proglib.io/p/mashinnoe-obuchenie-dlya-nachinayushchih-algoritm-sluchaynogo-lesa-random-forest-2021-08-12
9. Анализ малых данных. Случайный лес (Random Forest) – URL: https://alexanderdyakonov.wordpress.com/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/
10. Habr. Краткий обзор алгоритма машинного обучения Метод Опорных Векторов (SVM) – URL: https://habr.com/ru/articles/428503/