**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал научно-исследовательской работы**

Студент  **Бобряков Александр Сергеевич**

##### Институт №8 Информационные технологии и прикладная математика

###### Кафедра 810Б «Информационные технологии в моделировании и управлении»

##### Учебная группа М8О-203М-19

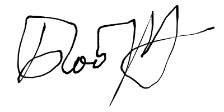
Направление подготовки (специальность)

**02.04.02 «Фундаментальная информатика и информационные технологии»**

Руководитель практики от МАИ

**Ревизников Д.Л. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Подпись руководителя

Студент

**Бобряков А.С.**  « 03 » февраля 2020г.

Подпись

**Москва 2021**

1. **Индивидуальное задание студенту**

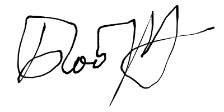
Анализ существующих подходов к отбору признаков на основе их важности.

Определение максимально важных признаков для настройки запуска задач в системе Hadoop в научной литературе.

Анализ максимально важных признаков на основе экспериментальных запусков map-reduce задач в Hadoop.

1. **План выполнения индивидуального задания**

|  |  |  |
| --- | --- | --- |
| № пп | Наименование раздела или этапа | Срок выполнения |
| 1 | Анализ существующих подходов к отбору признаков | 05.02.2021 |
| 2 | Поиск литературы по анализу важности параметров запуска задач в Hadoop | 01.03.2021 |
| 3 | Определение максимально важных признаков для настройки запуска задач в системе Hadoop в научной литературе | 01.04.2021 |
| 4 | Проведение экспериментальных запусков для определения важности параметров | 01.05.2020 |
| 5 | Анализ максимально важных признаков на основе экспериментальных запусков | 10.05.2020 |

Студент

**Бобряков А.С.**  « 03 » февраля 2021г.

1. **Отчет студента**

В ходе научно-исследовательской работы были проанализированы существующие подходы поиску важности признаков. Был осуществлен поиск максимально важных параметров среди нескольких сотен параметров, которые можно настроить в Hadoop. Были проведены запуски map-reduce задачи для экспериментального отбора признаков на основе алгоритма Extremely Randomized Trees.

Методы отбора признаков на основе их важности:

1. *Extremely Randomized Trees.*

Алгоритм Extremely Randomized Trees является ансамблевым алгоритмом на основе дерева решений. Этот алгоритм работает путем создания большого количества необрезанных регрессионных деревьев из обучающего набора и дальнейшего усреднения их прогнозов в случае регрессионной задачи. В отличие от случайного леса, который использует жадный алгоритм для выбора оптимальной точки разделения, данный алгоритм выбирает точку разделения случайным образом, а также использует всю обучающую выборку для создания деревьев.

Таким образом, есть три основных гиперпараметра, которые нужно настроить в алгоритме: количество деревьев решений в ансамбле, количество входных функций, которые следует случайным образом выбрать и рассмотреть для каждой точки разделения и минимальное количество выборок, необходимых в узле для создания новой точки разделения. Случайный выбор точек разделения делает деревья решений в ансамбле менее коррелированными, хотя это увеличивает дисперсию алгоритма. Этому увеличению дисперсии можно противодействовать, увеличив количество деревьев, используемых в ансамбле.

1. *Отбор признаков на основе χ*2*.*

Алгоритм проверяет, есть ли значимая разница между наблюдаемой и ожидаемой частотами двух категориальных переменных. Таким образом, проверяется нулевая гипотеза об отсутствии связи между двумя переменными. Данный метод относится к **методам фильтрации**. Методы фильтрации основаны на теории вероятностей и статистических подхода. модель машинного обучения использует только те предикторы, которые соответствуют определенным критериям. Методы фильтрации хороши тем, что они достаточно быстро работают: у них низкая стоимость вычислений, которая зависит линейно от общего количества предикторов. Однако, они рассматривают каждый признак изолированно, не учитывая их взаимное влияние друг на друга в частности и на целевую переменную вообще. Поэтому точность моделирования с этими методами отбора признаков недостаточно высока.

1. *Отбор признаков на основе встроенных методов.*

Встроенные методы не разделяют отбор признаков и обучение классификатора, а выделяют предикторы во время процесса расчета модели. Эти алгоритмы требуют больше времени, чем фильтрация. Основным методом этой категории является **регуляризация** – добавление дополнительных ограничений (штрафов) к условиям задачи, чтобы построить алгоритм, минимизирующий ошибку и количество используемых переменных. При этом выделяют 2 подхода: L1 (лассо, lasso regression, регуляризация через манхэттенское расстояние) и L2 (регуляризация Тихонова или ридж-регрессия, ridge regression). Регуляризация позволяет выявить взаимовлияние переменных, но занимает больше времени, чем методы фильтрации.

Параметры конфигурации Hadoop (в обеих версиях) влияют на несколько аспектов выполнения заданий на разных этапах, таких как параллелизм задач, выделение памяти, производительность ввода-вывода и использование полосы пропускания сети. В настоящее время Hadoop имеет более 200 параметров, из которых около 30 могут существенно повлиять на производительность работы. В таблице 1 перечислены некоторые из наиболее важных параметров конфигурации (действительные до Hadoop v3.2.1), которые используются для оптимизации производительности Hadoop. Значения параметров по умолчанию либо предоставляются Hadoop, либо указываются системным администратором и используются, если пользователь явно не указывает значения параметров во время отправки задания.

Таблица 1 – Ключевые параметры конфигурации Hadoop с учетом производительности

|  |  |  |
| --- | --- | --- |
| «dfs.block.size» | Размер блока по умолчанию для файлов, хранящихся в HDFS | 128 MB |
| «mapreduce.job.maps» | Количество задач карты | 2 |
| «mapreduce.job.reduces» | Количество сокращаемых задач | 1 |
| «mapreduce.combine.class» | Функция объединения для предварительной агрегации выходных данных карты перед фазой перемешивания | null |
| «mapreduce.map.combine.minspills» | Минимальное количество выходных файлов разлива карты, присутствующих для использования функции объединения | 3 |
| «mapreduce.map.output.compress" | Флаг для включения сжатия выходных данных карты | false |
| «mapreduce.map.sort.spill.percent" | Процент буфера «mapreduce.task.io.sort.mb», который нужно заполнить перед переносом вывода карты на локальный диск | 0.8 |
| «mapreduce.output.fileoutputformat.compress» | Флаг включения сжатия выходных данных задания | false |
| «mapreduce.reduce.input.buffer.percent» | Процент памяти редуктора, выделенный для буферизации выходных данных карты при выполнении задачи сокращения | 0 |
| «mapreduce.reduce.merge.inmem.threshold» | Максимальное количество перетасованных выходных пар карты до начала объединения во время перемешивания | 1000 |
| «mapreduce.reduce.shuffle.input.buffer.percent » | Процент памяти редуктора, выделенный для буферизации выходных данных карты во время перемешивания | 0.7 |
| «mapreduce.reduce.shuffle.merge.percent» | Процент уменьшения объема памяти задачи для заполнения до начала слияния во время перемешивания | 0.66 |
| «mapreduce.reduce.shuffle.parallelcopies» | Максимальное количество параллельных потоков, которые передают данные из задач карты в задачу уменьшения | 5 |
| «mapreduce.task.io.sort.factor» | Максимальное количество потоков данных для объединения во время внешней сортировки | 10 |
| «mapreduce.task.io.sort.mb» | Размер буфера памяти, в котором хранятся выходные данные карты | 100 МБ |
| «mapreduce.tasktracker.map.tasks.maximum» | Максимальное количество задач карты, выполняемых одновременно на узле кластера (количество слотов карты) | 2 |
| «mapreduce.tasktracker.reduce.tasks.maximum» | Максимальное количество задач сокращения, выполняемых одновременно на узле кластера (количество слотов сокращения) | 2 |

Для проведения экспериментальных поисков важных параметров, перечень которых приведен на рисунке 1, был собран набор данных из 15000 запусков map-reduce задачи.

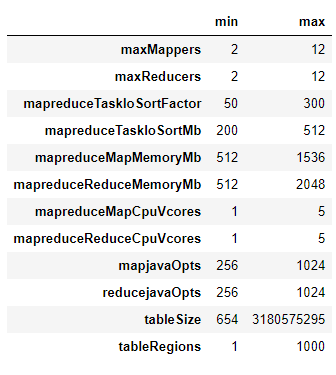


Рисунок 1 – Перечень изменяемых параметров с их min/max значениям при проведении экспериментальных запусков

Использование алгоритма на основе дерева решений Extremely Randomized Trees определило важные входные параметры, представленные на рисунке 2.

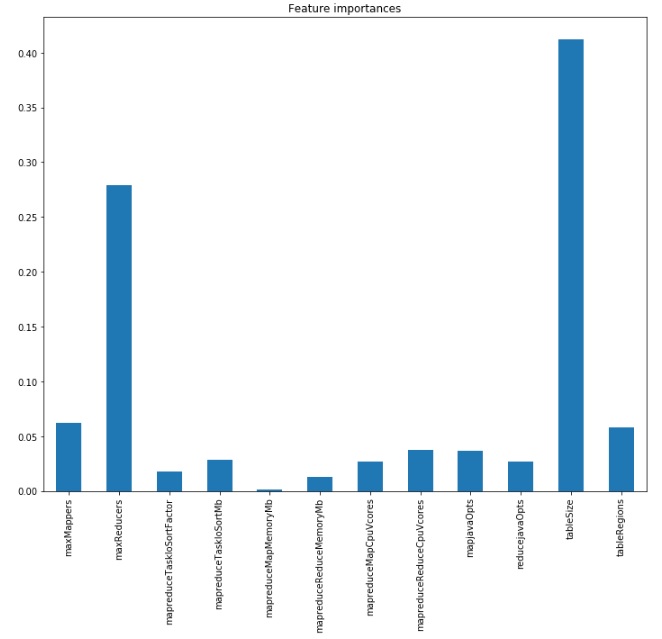
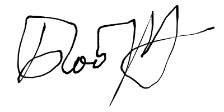


Рисунок 2 – Важность параметров алгоритмом Extremely Randomized Trees

1. **Выводы**

Проведен анализ существующих решений для отбора важнейших параметров выборки в датасете. При этом был выбран метод Extremely Randomized Trees, так как он использует жадный алгоритм для выбора оптимальной точки разделения и использует всю выборку, а также не имеет существенных недостатков, присущих другим методам. Среди изменяемых параметров при запусках задач в Hadoop наиболее важными оказались количество задач свертки, количество задач отображения, количество памяти в этих этапах, а также количество виртуальных процессоров, выделенных на каждый этап.

Студент

**Бобряков А.С.**  « 03 » февраля 2021г.