



Формирование модели машинного обучения для определения оптимальных параметров приложений
обработки больших данных на базе экосистемы Hadoop

Студент: Бобряков Александр Сергеевич

Научный руководитель: к.ф.-м.н., доцент Осипова Виктория Аркадьевна



Актуальность работы. В настоящее время с увеличением объема хранимых данных возникает потребность их обрабатывать для получения статистик и новой информации, необходимых для дальнейшего анализа работающих систем. Для решения этой задачи применяются системы обработки больших данных: Hadoop, Spark и другие. В связи с этим становится актуальным вопрос по их оптимальной настройке под выполняемые задачи.

Новизна подхода. Использование нейросетевого моделирования вместо стандартных алгоритмов машинного обучения в задаче определения оптимальных параметров.

Постановка задачи. Разработать подход к построению комплекса программного обеспечения, позволяющего производить автоматическую оптимальную настройку системы Hadoop под выполняемые на ней задачи.

Формальная постановка задачи

$$D = \text{duration} = F(\text{hdfsParams}, \text{yarnParams}, \text{hbaseParams}, \text{javaParams}, \text{ownParams}) = F(\text{Params})$$

где: *hdfsParams* – параметры настройки распределенной файловой системы HDFS,
yarnParams – параметры настройки системы для планирования заданий Yarn,
hbaseParams – параметры настройки нереляционной распределенной базы данных Hbase,
javaParams – параметры настройки java,
ownParams – параметры, присущие нашей задаче.

$$w^* = w_0^*, \dots, w_n^* = \underset{w_0, \dots, w_n}{\operatorname{argmin}} (\text{model} - D)^2$$

где: *model* – множество выходных значений модели машинного обучения,
D – множество обучающих значений для целевого показателя,
*w** – обученные веса модели

$$p_0^*, \dots, p_n^* = \underset{p_0, \dots, p_n}{\operatorname{argmin}} F_{\text{model}}(\text{Params})$$

где: F_{model} - функция предсказания длительности выполнения задачи моделью машинного обучения
 p^* – множество оптимальных параметров конфигурации,

Особенности и проблемы

1. Необходимо автоматически настраивать систему для минимизации различных целевых метрик, например, времени выполнения задачи либо потребления памяти
2. Решение должно быть встроено в процесс запуска задач системы Hadoop
3. Должна быть предоставлена возможность переобучать используемую модель в процессе использования
4. Необходимо учитывать ограничения ресурсов кластера
5. Должны учитываться параметры конфигурации, присущие самой выполняемой задаче

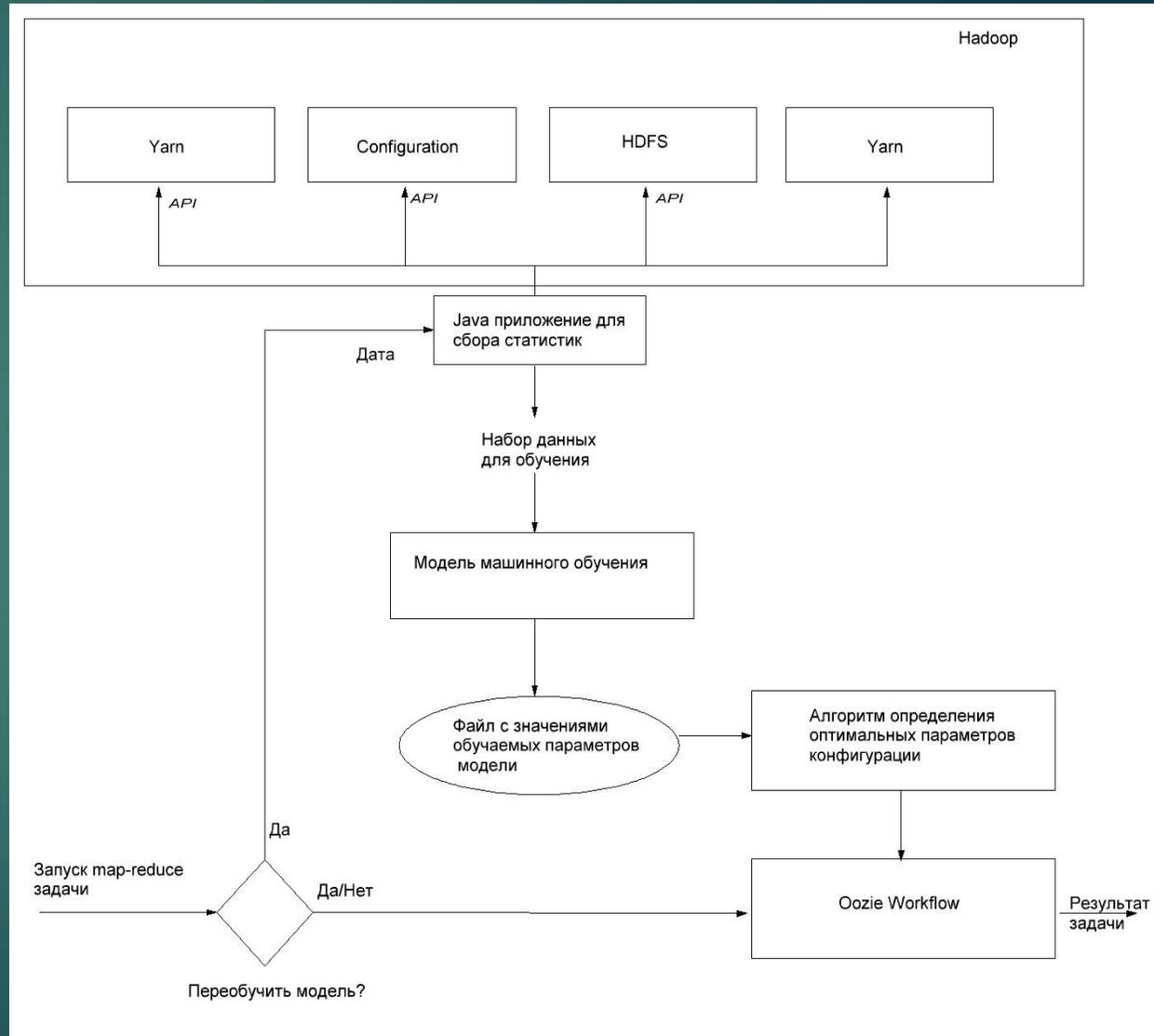
Опыт исследователей по данной тематике

1. *A Survey on Automatic Parameter Tuning for Big Data Processing Systems, 2020. Herodotos Herodotou, Yuxing Chen, Jiaheng Lu;*
2. *Using Machine Learning to Optimize Parallelism in Big Data Applications, 2017. Alvaro Brandon Hernandez, Maria S. Perez, Smrati Gupta, Victor Munteș-Mulero;*
3. *Learning-based Automatic Parameter Tuning for Big Data Analytics Frameworks, 2018. Liang Bao, Xin Liu, Weizhao Chen;*
4. *Quality Assurance for Big Data Application— Issues, Challenges, and Needs, 2016. Chuanqi Tao, Jerry Gao;*
5. *Machine Learning and Big Data Processing: A Technological Perspective and Review, 2018. Roheet Bhatnagar*

Метод решения

Предлагается выполнить следующие этапы для решения поставленной задачи:

1. Реализовать приложение для сбора статистик, на основе которых будет получен набор данных для обучения
2. Построить и обучить модель машинного обучения для предсказания целевой метрики (использовать методы для уменьшения размерности данных)
3. На основе обученной сети разработать алгоритм расчета оптимальных параметров конфигурации
4. Провести анализ проблемы ограниченности ресурсов и модифицировать алгоритм
5. Встроить полученный алгоритм в процесс запуска задач в системе Hadoop



Java приложение для сбора статистики

Использует:

1. REST API компонента Yarn:

`http://<address>:<port>/ws/v1/cluster/apps/{appid}`

`http://<address>:<port>/ws/v1/cluster/apps/?startTimeBegin=<time>`

2. Библиотека доступа к Hbase, Configuration:

`<dependency>`

`<groupId>org.apache.hadoop</groupId>`

`<artifactId>hadoop-common</artifactId>`

`<version>3.3.0</version>`

`</dependency>`

Результат:

Файл с расширением csv, содержащий набор обучающих данных

Параметр	Описание
<code>mapreduce.job.reduces</code>	Максимальное количество задач reduce
<code>mapreduce.job.maps</code>	Максимальное количество задач map
<code>mapreduce.map.cpu.vcores</code>	Количество ядер на задачу map
<code>mapreduce.reduce.cpu.vcores</code>	Количество ядер на задачу reduce
<code>mapreduce.map.memory.mb</code>	Максимальное количество памяти на задачу map
<code>mapreduce.reduce.memory.mb</code>	Максимальное количество памяти на задачу reduce

```
1 |blocksize;replication;javaMemory;reduce.memory.mb;reduce.cpu.vcores;tableNumber;job.maps;parallelJobsLimit;
2 |134217728;2;7168;3072;267;230;1073;2;109;20;2048;637083;1086;71;256;13;90;288;10099
3 |402653184;3;5120;3072;274;237;1105;1;291;23;3072;297303;1161;65;256;14;90;258;2643
4 |268435456;3;7168;2048;252;238;1149;3;143;22;4096;575375;1140;80;512;14;80;290;5545
5 |134217728;2;5120;4096;265;249;1039;2;207;24;2048;674228;1012;72;256;13;90;176;10490
6 |268435456;3;5120;3072;235;234;1125;1;181;27;4096;483084;1094;67;256;11;80;230;7010
7 |134217728;3;7168;3072;137;245;1102;1;134;30;4096;454830;1125;68;512;10;90;135;7760
8 |134217728;3;5120;2048;229;247;1057;1;190;30;2048;329425;1128;71;512;14;90;203;10225
9 |402653184;3;7168;2048;275;233;1197;3;232;20;3072;701804;1053;74;256;11;90;186;7600
10 |402653184;2;7168;4096;279;245;1096;3;124;28;3072;778179;1006;68;512;15;90;252;10153
11 |268435456;3;7168;4096;215;234;1130;1;203;27;4096;111673;1076;66;256;10;90;127;2984
12 |134217728;3;5120;4096;251;242;1026;2;149;23;2048;656802;1166;68;256;12;90;273;8622
13 |268435456;2;7168;2048;102;245;1143;3;219;28;4096;382457;1108;76;512;12;80;163;10357
14 |268435456;3;7168;4096;116;242;1119;3;116;26;2048;184787;1139;65;512;14;90;143;4449
15 |402653184;2;5120;2048;258;249;1196;2;278;20;3072;194584;1107;79;512;10;90;193;2856
16 |402653184;2;5120;4096;105;233;1147;3;213;22;3072;278682;1115;70;256;10;80;252;3534
17 |268435456;2;5120;2048;115;250;1185;1;234;30;4096;91219;1144;65;512;11;80;250;1887
18 |134217728;3;7168;3072;133;232;1051;3;158;24;4096;438618;1054;66;256;11;90;286;10393
19 |402653184;2;7168;4096;293;249;1098;1;232;25;2048;231611;1025;75;256;13;90;255;2709
20 |268435456;3;7168;4096;138;234;1008;3;201;23;4096;158769;1012;69;512;10;90;157;3280
21 |402653184;2;5120;4096;140;236;1189;2;272;21;4096;178612;1018;70;256;14;90;107;5337
22 |134217728;2;5120;4096;268;230;1088;1;168;25;3072;741589;1073;64;512;13;90;228;8889
23 |134217728;3;5120;4096;230;238;1078;3;132;29;2048;318993;1094;78;256;12;90;180;10164
24 |134217728;3;5120;4096;231;243;1139;1;118;20;4096;83227;1134;80;256;15;90;202;2309
25 |268435456;3;7168;2048;245;235;1151;1;299;29;3072;202684;1064;67;512;15;80;131;5167
26 |268435456;3;7168;3072;254;242;1005;3;169;26;3072;185018;1054;70;512;12;80;149;3970
27 |268435456;2;7168;2048;203;246;1076;1;198;26;2048;524584;1142;78;512;12;90;209;10793
28 |134217728;2;5120;3072;279;238;1011;1;132;25;4096;83386;1139;75;512;13;80;144;5192
29 |134217728;3;5120;3072;207;235;1103;2;103;27;3072;660063;1169;67;256;13;80;263;9551
```

Модель машинного обучения

Вход:

Файл csv с обучающими данными

Использует:

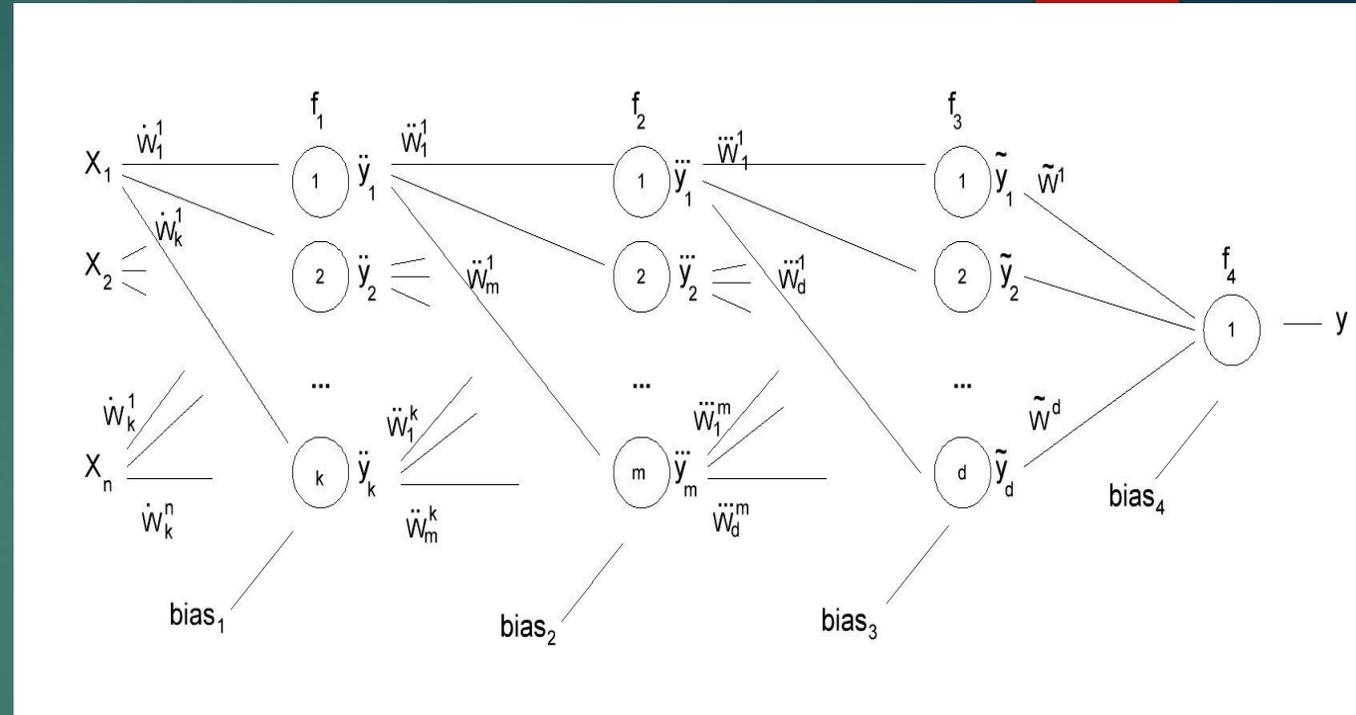
1. Библиотек Scikit-learn
2. Библиотека Keras

Этапы:

1. Подготовка данных
2. Отбор признаков по важности (алгоритм *Extremely Randomized Trees*)
3. Обучение нейронной сети

Результат:

Файл `model.py`, содержащий архитектуру нейронной сети и рассчитывающий весовые коэффициенты



$$y = f_4 \left(\sum_{r=1}^d \left(\tilde{w}^r \left[f_3 \left(\sum_{i=1}^m \left(\ddot{w}_d^i \left[f_2 \left(\sum_{j=1}^k \left(\dot{w}_i^j f_1 \left(\sum_{l=1}^n (\dot{w}_j^l x_l) + bias_1 \right) \right) + bias_2 \right] \right) + bias_3 \right] \right) + bias_4 \right) \right) \right)$$

Алгоритм определения оптимальных параметров конфигурации

Вход:

Весовые коэффициенты нейронной сети w

Использует:

1. Библиотек *Scikit-learn*

Этапы:

1. *Определение условий, задающих ограничение искомым параметром*
2. *Минимизация математической функции, описывающей результат выходного слоя (алгоритм оптимизации последовательного квадратичного программирования SQP)*

Результат:

Файл `algoritm.py`, рассчитывающий оптимальный набор конфигурационных параметров

```
# Примеры ограничений
def constraint_memory(x):
    mapper_memory = x[0]
    reducer_memory = x[1]
    # Сумма размеров памяти не превышает 1024*5
    return 1024*5 - (mapper_memory + reducer_memory)

def constraint_cores(x):
    mappers_cores = x[2]
    reducers_cores = x[3]
    # Сумма количества ядер не превышает 1500
    return 1500 - (mappers_cores + reducer_cores)
```

Внедрение в Oozie Workflow

Вход:

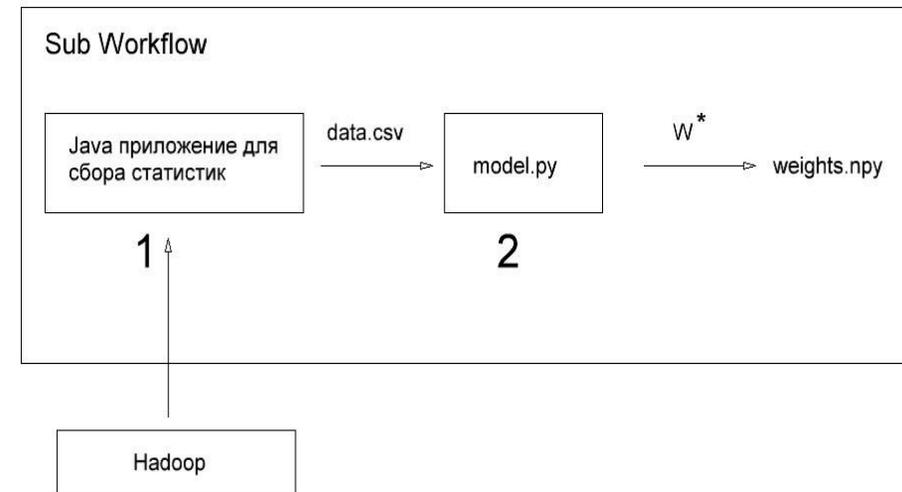
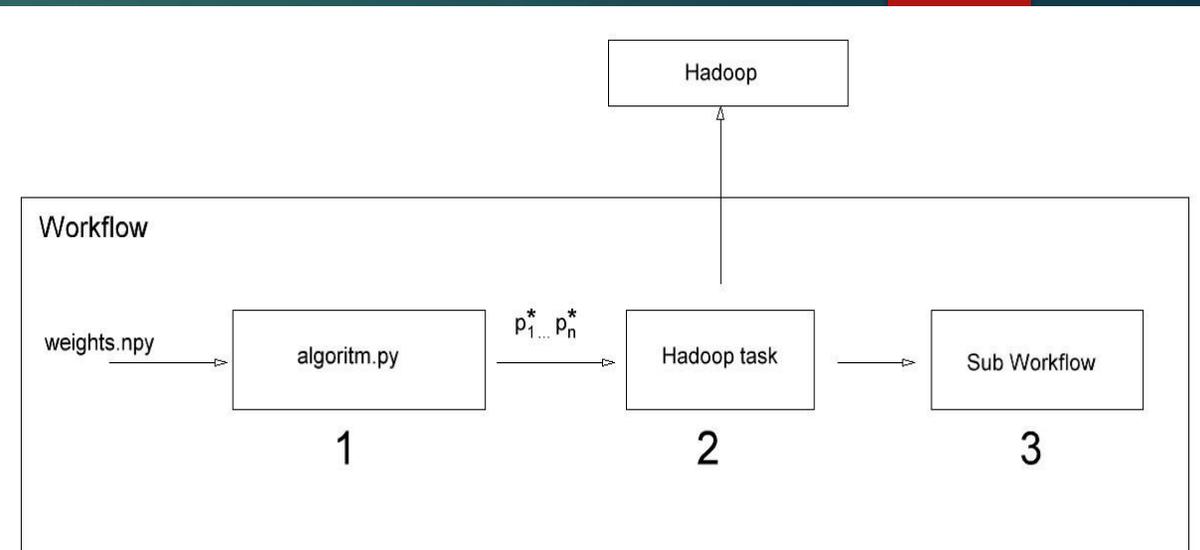
Файл *model.py*, *algorithm.py*

Использует:

1. Система Oozie

Этапы:

1. Запуск файла *workflow.xml*, включающий необходимые этапы подстановки автоматических параметров в запускаемую задачу
2. Возможный запуск файла *sub-workflow.xml* для переобучения нейронной сети



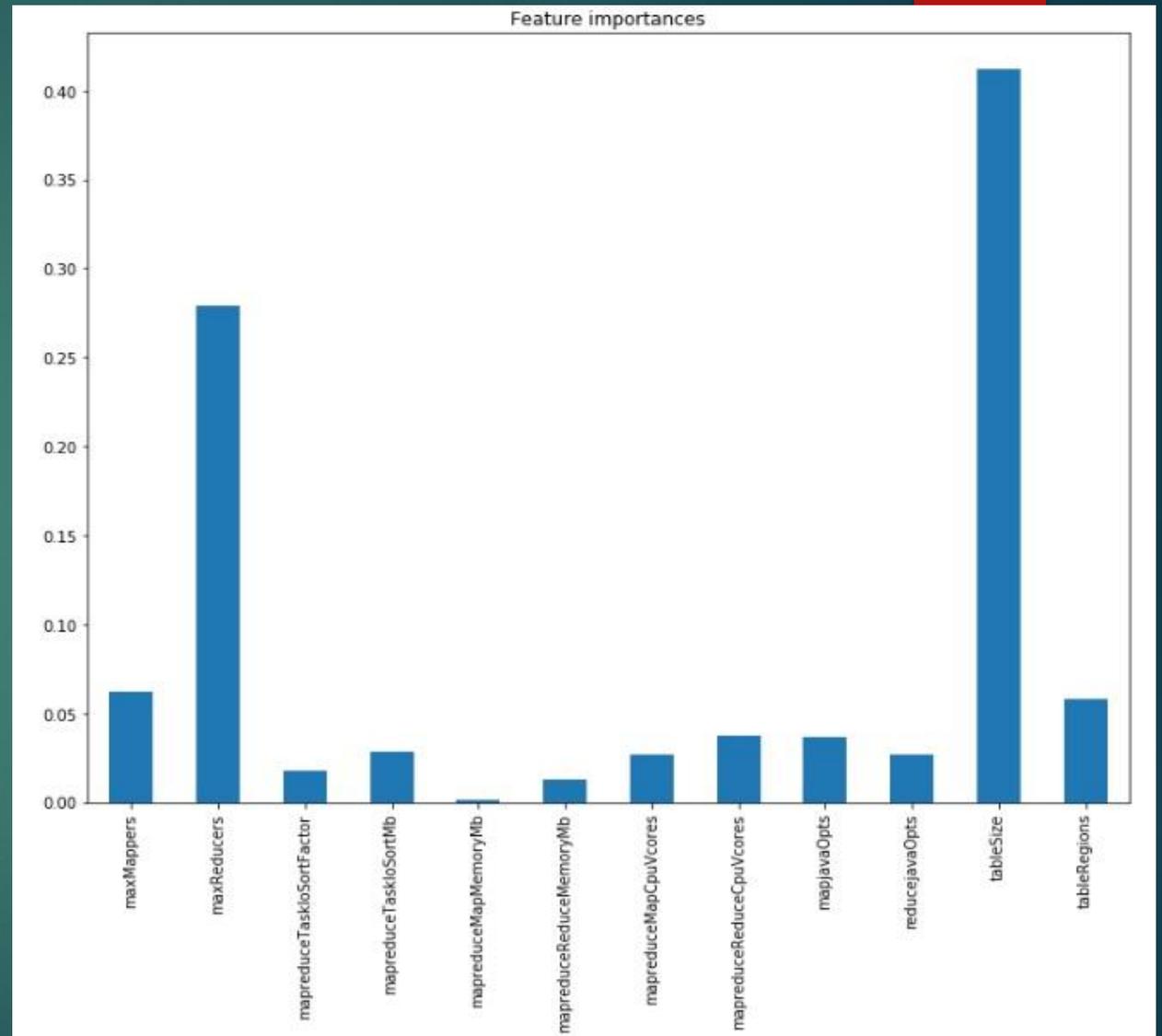
Важность параметров запускаемой задачи

10 настраиваемых параметров:

mapreduce.task.io.sort.factor,
mapreduce.task.io.sort.mb,
max.reducers,
max.mappers,
mapreduce.map.cpu.vcores,
mapreduce.reduce.cpu.vcores,
mapreduce.map.memory.mb,
mapreduce.reduce.memory.mb,
mapper.java.option,
reducer.java.option

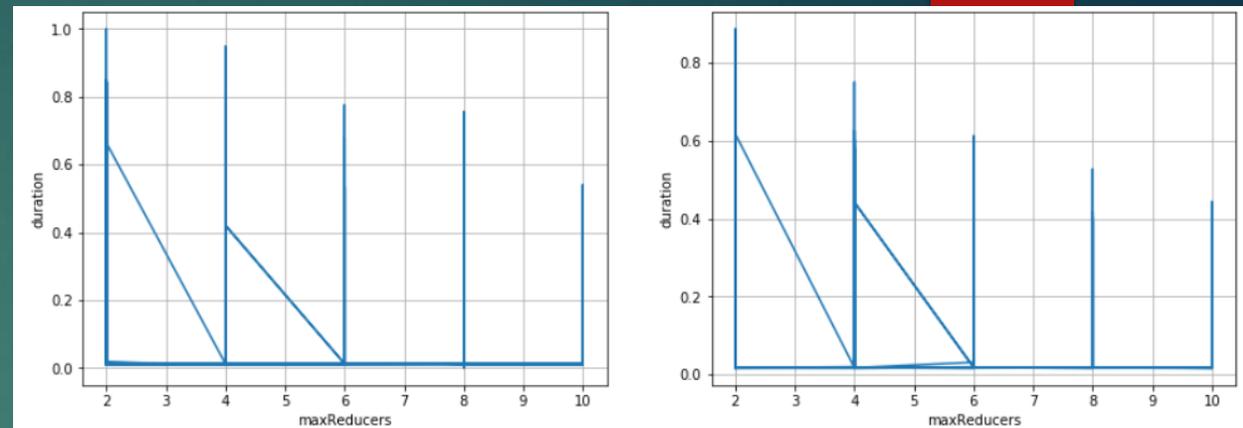
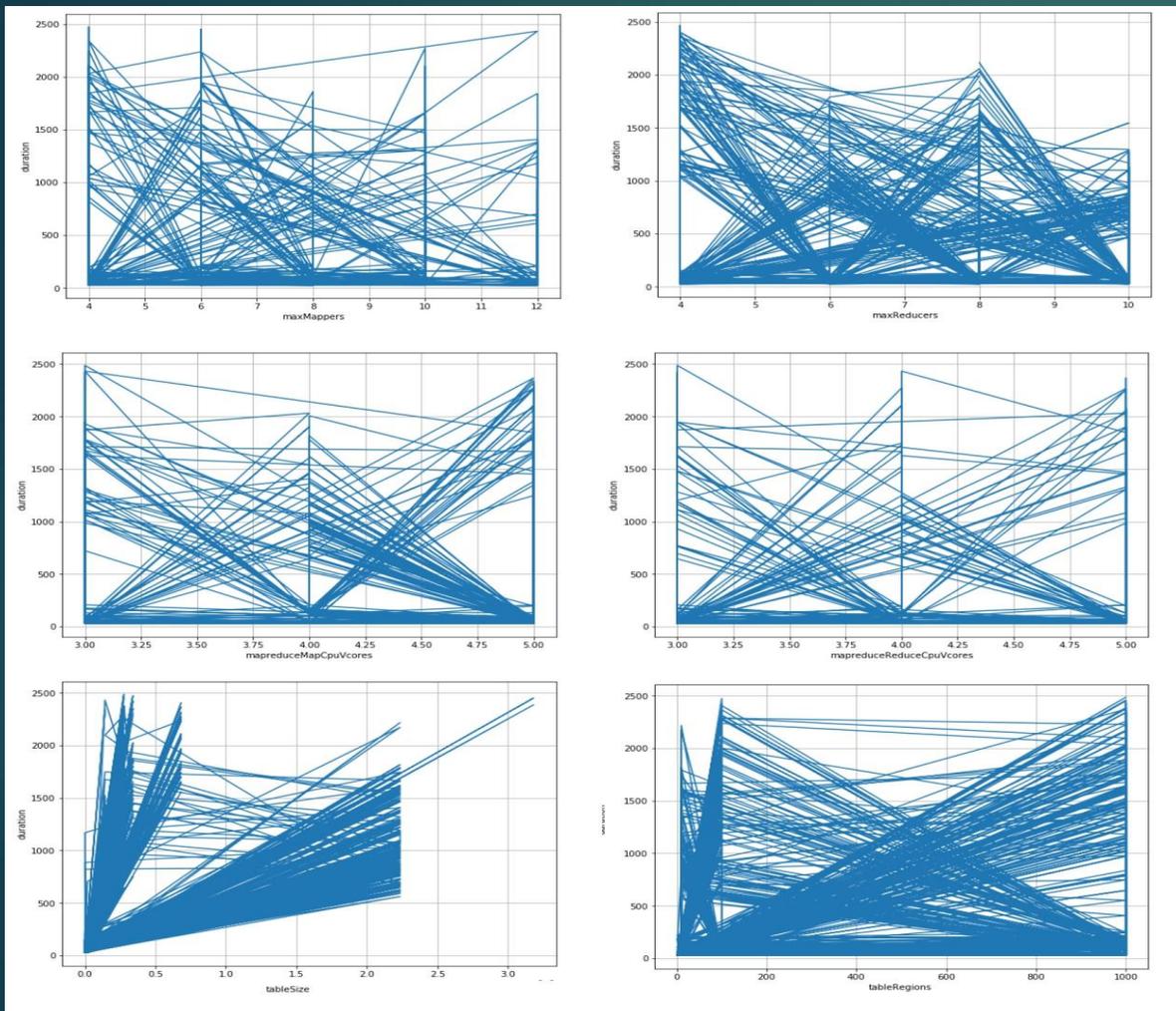
2 параметра, присущих выполняемой задаче:

table.size,
table.regions



Важность параметров по алгоритму на основе дерева решений
(Extremely Randomized Trees)

Длительность выполнения задачи в зависимости от параметров



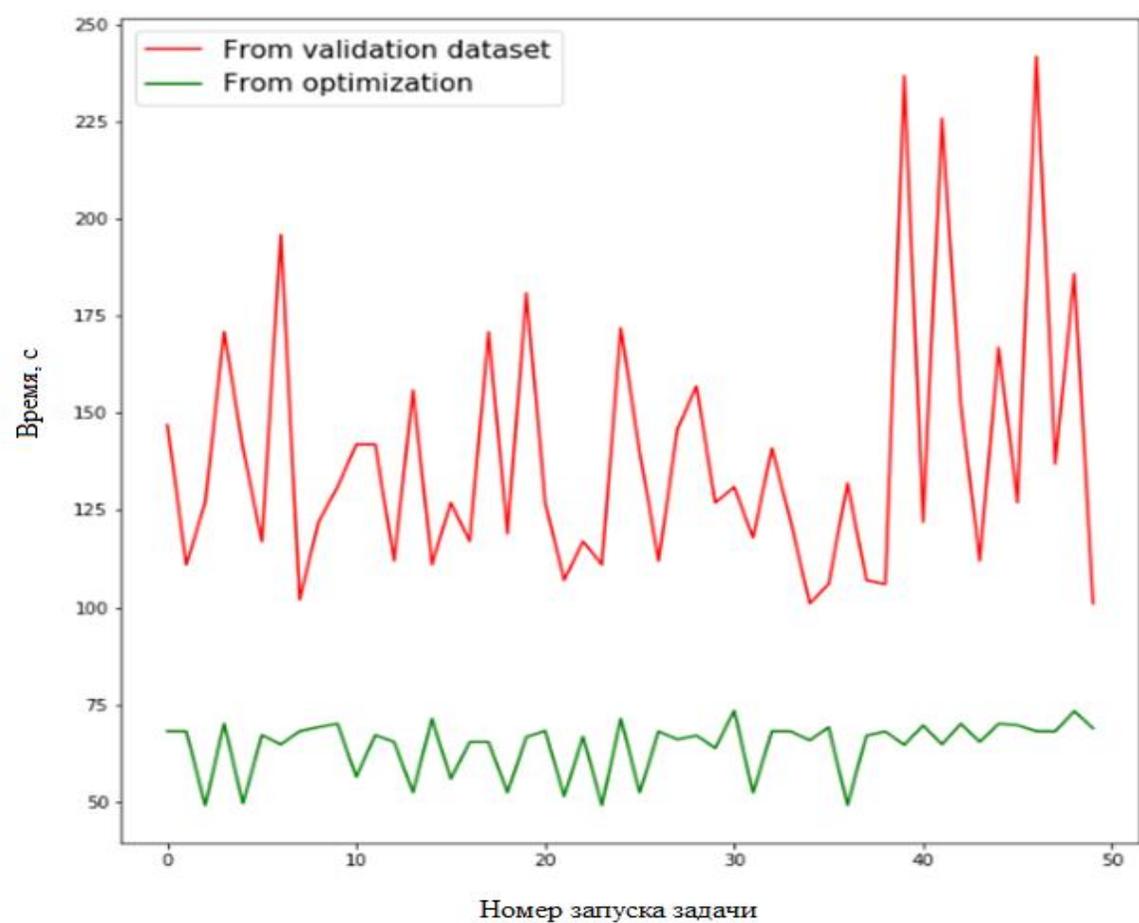
Сравнение длительности выполнения задачи для тестовой выборки исходного набора данных и предсказанных моделью значений от количества задач свертки

Длительность выполнения задачи в зависимости от входных параметров: а) от максимального количества задач отображения, б) от количества задач свертки, в) от размера таблицы, г) от количества разбиений (регионов) таблицы

Запуск задачи в условиях ограничений ресурсов

```
1: min_values = {'maxMappers': [2],\  
2:               'maxReducers': [2], \  
3:               'mapreduceTaskIoSortFactor': [10], \  
4:               'mapreduceTaskIoSortMb': [200], \  
5:               'mapreduceMapMemoryMb': [512],\  
6:               'mapreduceReduceMemoryMb': [512], \  
7:               'mapreduceMapCpuVcores': [1], \  
8:               'mapreduceReduceCpuVcores': [1], \  
9:               'mapjavaOpts': [512],\  
10:              'reducejavaOpts': [512], \  
11:              'tableSize': [500000], \  
12:              'tableRegions': [200] \  
13:            }\  
14: max_values = {'maxMappers': [15],\  
15:               'maxReducers': [15], \  
16:               'mapreduceTaskIoSortFactor': [500], \  
17:               'mapreduceTaskIoSortMb': [512], \  
18:               'mapreduceMapMemoryMb': [2048],\  
19:               'mapreduceReduceMemoryMb': [2048], \  
20:               'mapreduceMapCpuVcores': [5], \  
21:               'mapreduceReduceCpuVcores': [5], \  
22:               'mapjavaOpts': [1024],\  
23:               'reducejavaOpts': [1024], \  
24:               'tableSize': [500000], \  
25:               'tableRegions': [200] \  
26:            }
```

Пример ограничений входных параметров



Сравнение длительности выполнения задачи для тестовой выборки исходного набора данных с длительностью после оптимизации параметров для нескольких запусков

Основные результаты:

1. Разработаны архитектура и программный подход к построению системы, позволяющей производить автоматическую оптимальную настройку Nadoor под выполняемые на ней задачи.
2. Реализованы все основные компоненты системы. Каждый из них является работоспособным.
3. Произведено внедрение всех компонентов в процесс их последовательного запуска

Основной недостаток:

1. Подход применим для задач, которые запускаются периодически без изменений логики обработки данных на различных объемах

Дальнейшее развитие:

1. Определение оптимальных параметров по заданному ограничению времени
2. Независимость от конкретно запускаемой задачи
3. Автоматическая рекомендация по оптимальному количеству разбиения таблиц на составляющие регионы

1. Бобряков А.С. “Нейросетевое моделирование для автоматической настройки задач в системах обработки больших данных”, XLVII Гагаринские чтения, Москва 2021