АНТИПЛАГИАТ
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Отчет о проверке на заимствования №1

Автор: МФТИ admin@phystech.edu / ID: 211
Проверяющий: (admin@phystech.edu / ID: 211)
Организация: Московский физико-технический институт

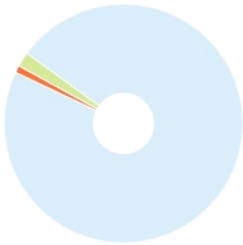
Отчет предоставлен сервисом «Антиплагиат» - http://mipt.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 14078
Начало загрузки: 20.06.2020 21:23:20
Длительность загрузки: 00:01:09
Имя исходного файла: Неизвестно
Название документа: Diplom_Lemikhov.pdf
Размер текста: 1 кБ
Символов в тексте: 30535
Слов в тексте: 4085
Число предложений: 237
Method of text extraction: OCR

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 20.06.2020 21:24:30
Длительность проверки: 00:00:13
Комментарии: не указано
Модули поиска: Модуль поиска ИПС "Адилет", Модуль выделения библиографических записей, Сводная коллекция ЭБС, Коллекция РГБ, Цитирование, Модуль поиска переводных заимствований, Модуль поиска переводных заимствований по elibrary (EnRu), Модуль поиска переводных заимствований по интернет (EnRu), Коллекция eLIBRARY.RU, Коллекция ГАРАНТ, Модуль поиска Интернет, Коллекция Медицина, Модуль поиска "МФТИ", Модуль поиска перефразирований eLIBRARY.RU, Модуль поиска перефразирований Интернет, Коллекция Патенты, Модуль поиска общеупотребительных выражений, Кольцо вузов



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
0,3%	0%	2,15%	97,55%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	1,75%	1,75%	не указано	не указано	раньше 2011	Модуль выделения библиографических записей	1	1
[02]	0,41%	0,41%	не указано	не указано	раньше 2011	Модуль поиска общеупотребительных выражений	4	4
[03]	0,3%	0,3%	Факультет Вычислительной Математи..	http://dis.podelise.ru	14 Мая 2020	Модуль поиска Интернет	1	1

Министерство общего и профессионального образования
Российской Федерации
Московский Физико-технический институт
(Государственный университет)
Факультет управления и прикладной математики
Кафедра Acronis

Выпускная квалификационная работа
"Anomaly detection"

Студента 4-го курса Лемихова Александра Алексеевича

Научный руководитель
Андрей Кулага

Москва, 2020

Содержание

1 Аннотация

В данная выпускная квалификационная работа посвящена предсказанию временных рядов. Основной упор сделан на распространении одного метода на всё многообразие дата центров компании Acronis. Результатом работы является метод предоработки входных данных, построений предсказаний и оценки результатов которые подходят для существенной части хранилищ.

2 Введение

Данная работа является вспомогательным инструментом для системы автоматического определения аномалий. Множество методов определения аномалий используют предсказания временных рядов. Ошибка в предсказаниях является нарушением установленных связей в системе и может пролить свет на неточности системы, которые невозможно уловить человеческими усилиями, ввиду огромного количества данных.

Для автоматизации обработки данных предлагается использовать базовые механизмы машинного обучения - линейную регрессию. Её преимуществом является интерпретируемость, возможность на каждом шаге отследить корректность модели.

На сегодняшний день построение предсказательных систем является важной областью машинного обучения. Разработано множество подходов к предсказанию временных рядов множества переменных. Несмотря на это, поиски литературы об универсальности методов для различных систем не принесли результатов.

Основной целью работы является выделение из класса моделе `SARMA` той, которая будет одинаково хорошо подходить под кажый instance продукта ABGW Acronis.

Работа разделена на следующие части:

- Обзор методов, которые могут быть применены в данной задаче
  - Сбор данных
  - Предобработка

  - Модель
  - Оценка модели

- Далее каждый этап модели представлен в отдельной секции более подробно

В процессе работы был исследован класс моделей `ARMA` и предложено универсальное решение, подходящее под большое множество инстансов.

3 Постановка задачи

Цель задачи - приспособить широко известный концепт - линейную регрссию для предсказания и исследования временных рядов. Как основной концепт, исопльзуемый для построения модели в работе используется касс моделей `ARMA`.

Идея исследовать линейные зависимости в системе давно развивается Андереем Кулагой и его студентами. Важное свойство - линейность, была обнаружена зорким глазом, при наблюдении метрик системы в Graphana. Линеные взаимосвязи был выявлены среди признаков, со следующей сутью:

- Производная суммарной задержки iop операций.
- Производная суммарного счётчика запросов, вызвавших задержку.
- Число подключений.
- Производная суммарной задержки, видной польователю.

На сегодняшний день Acronis собирает данные о много большем числе признаков системы. Из этого возникает возможность исследования большего числа метрик, для построения более качественной модели.

Ниже перечислены возможные подхды, которые были применены к системе

4 Подходы к решению

4.1 Сбор данных

Для построения системы, строящей предсказания на основе взаимосвязи между метриками необходима система сбора и хранения данных. В компании Acronis для этого используется Prometheus [1], предоставляющий возможность выгрузки данных с дискретность не выше 1 минуты. Как оказалось, этого вполне хватает для успешного построения модели. Сбор данных продолжается исследованием структуры данных.

4.2 Выбор данных

Acronis использует сервис мониторинга Prometheus, с помощью которого можно загрузить данные в удобном формате. В работе выводы построены на данных в период с 14/01/2020 00:01 до 01/05/2020 00:01 со следующих дата-центров:

1. au2-acs1
2. us6-acs2
3. eu3-acs1
4. us3

Необходимо выбрать ограниченный набор дата-центров, чтобы трудоёмкость вычислений и объёмы данных не мешали построению выводов. Тем не менее этот набор данных позволяет наблюдать различные нагрузки и шаблоны поведения системы.

В качестве целевой метрики, метрики для предсказаний была выбрана следующая метрика:

```
abgw_iop_latency_ms_sum{err="OK proxied="0 iop="isync"}
```
(1)

Исследуя все остальные метрики был составлен набор информативных метрик. Популярные примеры не информативны метрик - константы на довольно длинных участках.

Кроме того, следует упомянуть, что запрос принципиально возвращает матрицу. Например `abgw_client_conns_cur{dc="us3 instance="5"}` вернёт матрицу, всех возможных `abgw_client_conns_cur` для заданого инстанса по различным фильтрам. Одним из таких фильтров является `proto="v1"`, который возник неожиданно и не на всех инстансах. Было принято решение перейти от матрицы к вектору - метрике путём суммирования построчно. Такое решение не изменит сути признаков и их интерпритация не изменится.

Список рассматриваемых метрик приведён ниже:

1. `abgw_iop_latency_ms_count{err="OK", job="abgw", iop="isync", proxied="0"}`
2. `abgw_account_lookup_errs_total{err="OK"}`
3. `abgw_account_pull_errs_total{err="OK"}`
4. `abgw_accounts`
5. `abgw_append_throttle_delay_ms_total`
6. `abgw_client_conns_cur`
7. `abgw_client_conns_total`
8. `abgw_fds)`
9. `abgw_file_lookup_errs_total{err="OK"}`
10. `abgw_files`
11. `abgw_read_bufs`
12. `abgw_read_bufs_bytes`
13. `abgw_read_bytes_total{proxied="0"}`
14. `abgw_read_reqs_total`
15. `abgw_req_latency_ms_count`
16. `abgw_req_latency_ms_sum`

17. `abgw_account_lookup_errs_total`

18. `(abgw_account_pull_errs_total{err="OK"}`

4.3 Предобработка

Теория временных рядов предполагает разложение ряда на трендовую и сезонную составляющие, обуславливаемые наблюдаемыми признаками. Помимо них присутствует ошибка, которая в идеальном случае должна быть нормальной центрированной случайной величиной. И последняя составляющая - циклы, отличающиеся от сезонности непостоянностью периода.

Кроме стационарности важно распределние, соответсвующее потоку данных. Особенности в данных, такие как выбросы и пропуски должны быть вявлены и процесс вся модель, вместе с предобработкой должна учитывать их.

Задача предобработки - получить стационарные данные, удовлетворяющие этим требованиям.

4.4 Модель

Имея набор данных - метрик нужно сформировать признаковое пространство и гипотезу об ошибке. Максимизация функции правдоподобия для выбранной вероятностной модели приводит к выбору функции потерь. Как правило, такой поход требует дополнения в виде регуляризаторов - штрафа за сложность и нестабильность модели или же изменения гипотезы о вероятности ошибки.

Построение универсальной модели для семейства объектов предполагает подбор гиперпараметров - параметров, которые подбираются один раз на некоторое множество моделей. Такие параметры присутствуют, как правило, в большинстве алгоритмов. Для подбора гиперпараметра функция ошибки должна быть масштабируемой - корректно оценивать качество модели, например, не зависимо от нагрузки, количества пользователей.

4.5 Оценка модели

В рамках исследования класса моделей `ARMA` можно выделить следующие критерия качества модели:

- Гетероскедастичность
- Однородность временного ряда ошибки
- Центрированность ошибки

Общепринятный проверки качества - анализ графиков. Также можно проверять статистические гипотезы. Однако, модель применена к огромному количесвту данных и поэтому мощность критериев становится слишком велика [2].

5 Предобработка

5.1 Тренд

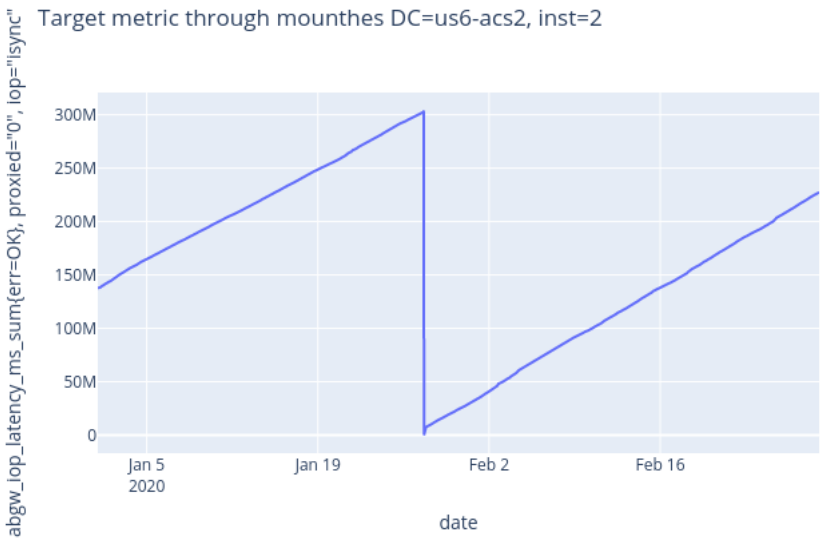


Рис. 1: Целевая метрика 1 за исследуемый период

На рис.1 показан график целевой переменной. Уменьшение целевой переменной между 19 марта и 2 февраля - явный признак перезагрузки системы. Такие участки необходимо удалить из данных для построения

зависимостей. Кроме того в данных явно присутствует тренд. Самый простой способ избавиться от тренда - перейти к приращениям, производной. Такой подход оказался достаточно универсальным.

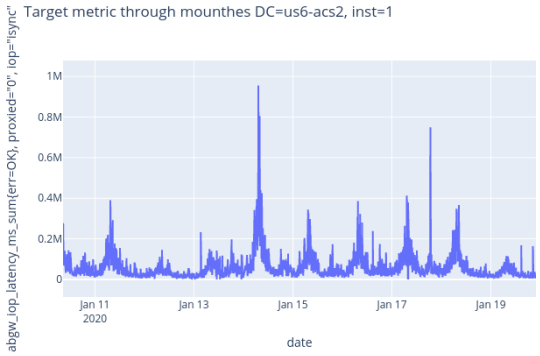


Рис. 2: Целевая метрика в пространстве производных, dc=us6-acs2, inst=23

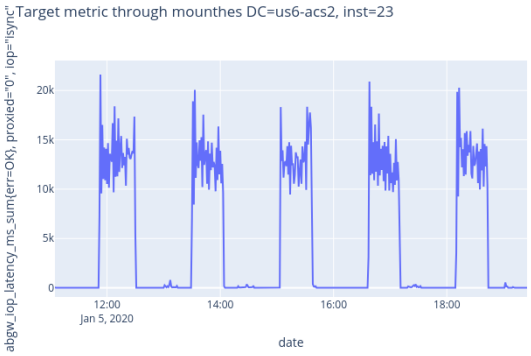


Рис. 3: Целевая метрика в пространстве производных, dc=us6-acs2, inst=1

Результатом дифференцирования суммарной задержки является накопленная за минут задержка. Её можно интерпретировать как загрузку системы.

5.2 Масштабирование

В отсутствии тренда метрика локализована на компакте. Однако стоит отметить разницу загрузки систем в десятки раз. Один из способов избавиться от такой разнице в загрузке - перейти к масштабированным переменным на $[0; 1]$. Такой переход продемонстрирована в формуле 2.

$$x_{sc} = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

Применение такого масштабирования обеспечит простую интерпретацию целевой переменной - доля максимальной загрузки.

Однин из минусов такой предобработки - неустойчивость к выбросам. Стоит упомянутб преобразование, не страдающее таким неугом - преобразование Yeo-Johnson:

$$\hat{x}_\lambda = \begin{cases} ((x+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, x \geqslant 0 \\ \log(x+1) & \text{if } \lambda = 0, x \geqslant 0 \\ -[(-x+1)^{(2-\lambda)} - 1]/(2-\lambda) & \text{if } \lambda \neq 2, x < 0 \\ -\log(-x+1) & \text{if } \lambda = 2, x < 0 \end{cases} \tag{3}$$

Парметр λ выбирается, чтобы приблизить распределение признака к нормальному. Сравнение этих двух подходов представлено в [3]. Для построения можели, в первом приближении, в работе используется `MinMaxScaler` (2).

5.3 Сезонность

На рис.2 и рис.3 легко видеть сезонность. В этой части приведено рассмотрение несколько способов предобработки сезонности. Известны следующие подходы:

1. Линейная регрессия метрики в прошлое
2. Сезонное дифференцирование.

Так-же различают сезонность аддитивную 4 и сезонность мультипликативную 5.

$$x \sim seasonal \cdot trend \tag{4}$$

$$x \sim seasonal + trend \tag{5}$$

Взятием логарифма можно перейти от мультипликативной сезонности к аддитивной, или, иначе говоря, стабилизировать дисперсию. Также это позволяет сделать преобразование Yeo–Johnson 3.

Нужно упомянуть, что возможна проверка ряда $x(t)$ на стационарность с использованием статистических критериев. В данной работе использовался критерий `KPSS`(Kwiatkowski–Phillips–Schmidt–Shin). Выделяет его среди всех остальных реализация на языке `Python`. Критерий проверяет

следующую гипотезу:

$$H_0 : \texttt{Какое бы окно ряда x(t) мы не рассматривали, распределение не изменится относительно тренда} \tag{6}$$

В анализе одномерных временных рядов зачастую используют как добавление предыдущих, с сезонным шагом, значений как признаки в модель, так и сезонное дифференцирование. В случае добавления признаков на стационарность нужно проверять временной ряд из ошибок предсказаний. В данной работе предлагается использовать подход с расширением пространства признаков, т.к. ни сезонное дифференцирование, ни преобразования Yeo-Johnson не позволили принять гипотезу о стационарности 6 на уровне значимость 0.05. Иллюстрация временного ряда до и после преобразования 3 приведена на рис.4, 5

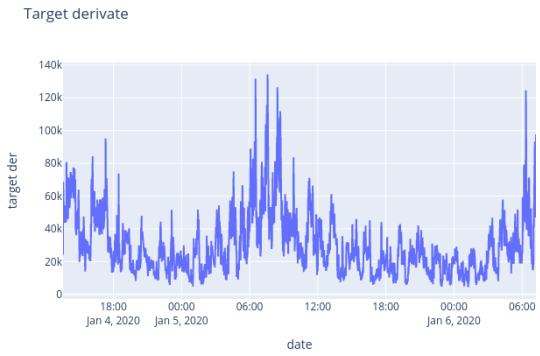


Рис. 4: Производная целевой переменной

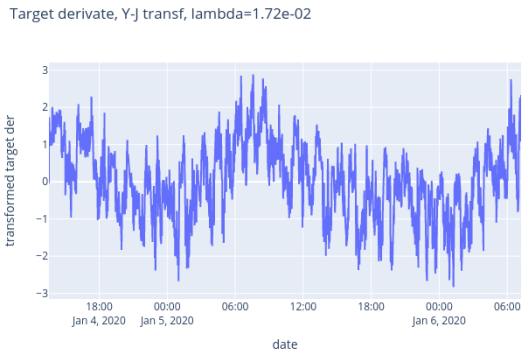


Рис. 5: Производная целевой переменной после масштабирования и преобразования Yeo-Johnson с $\lambda = 1.72e-02$

Это преобразование, действительно, стабилизирует дисперсию. Тем не менее стационарным этот ряд сделать не получилось.

Стоит отметить, что на самом деле это не сезонность. При более детальном рассмотрении, разница в ежедневных пиках загрузки может составлять и 15 минут и более. Таким образом, если и есть подход, учитывающий сезонность, он должен быть достаточно рабастым.

6 Модель

6.1 Описание классических моделей

Один из основных подходов к прогнозированию временных рядов - модели `SARMA(p, q) × (P, Q)`. Смысл аббревиатуры приведён ниже:

S - Модель учитывает сезонность длинны S

AR - Модель учитывает признаки за p предыдущих моментов времени.

MA - Модель учитывает свои ошибки за q предыдущих предсказаний

Итоговую модель можно сформулировать следующим образом:

$$y_t = \varepsilon_t + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \\ \phi_S \cdot y_{t-S} + \cdots + \phi_{PS} \cdot y_{t-SP} + \theta_S \cdot \varepsilon_{t-S} + \cdots + \theta_{QS} \cdot \varepsilon_{t-QS}$$

где $\varepsilon_\tau \sim \mathcal{N}(0, \sigma)$ - нормальный шум. Поскольку шум невозможно наблюдать, ε_τ вычисляется как ошибка предсказаний. Предсказания ряда строятся следующим образом:

$$\hat{y}_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \\ \phi_S \cdot y_{t-S} + \cdots + \phi_{PS} \cdot y_{t-SP} + \theta_S \cdot \varepsilon_{t-S} + \cdots + \theta_{QS} \cdot \varepsilon_{t-QS} \tag{7}$$

Обозначив коэффициенты ϕ_τ и θ_τ и имеющемся наборе данных можно составить серию предсказанных и известных значений, посчитать ошибку. Как правило используют `MSE`, поскольку такая ошибка даёт несмещённые оценки. Коэффициенты ϕ_τ и θ_τ находятся как решение оптимизационной задачи по минимизации ошибки.

Стоит упомянуть, что $\hat{y}$ - оценка целевой переменной, а y - признаковое описание системы, включающее лаг целевой переменной.

Расширением является класс моделей `SARIMA(p, d, q) × (P, D, Q)` - ряд предварительно дифференцируется `d` раз. В данной работе построение начинается с `SARIMA, d = 1`. Впоследствии возможен перебор и этого гиперпараметра.

Описание важных особенностей системы в данной работе приведено через поэтапное построение модели.

6.2 AR(p)

Строить `SARMA` - схожие модели мы будем используя линейную регрессию. Предсказывать мы будем y_t, а в качестве признаков будем использовать не только значения y_{t-i}, как в 7, но и другие метрики, которые наблюдает Acronis, и их значения в предыдущие моменты. В итоге задача сводится к простой линейной регрессии.

Для начала положим `p=1`. Как оказалось, рассмотрение даже такого небольшого класса позволит увидеть некоторые закономерности в данных. Для начала сформируем выборки для теста и обучения. Варьируя размер обучающей выборки построим график ошибки от размера обучающего набора данных. Он представлен на рис.6. Оценкой качества модели считается `MAE`. Внезапный скачок ошибки обусловлен единичным выбросом, детальней можно пронаблюдать эту ситуацию на рис.8 и рис.9.

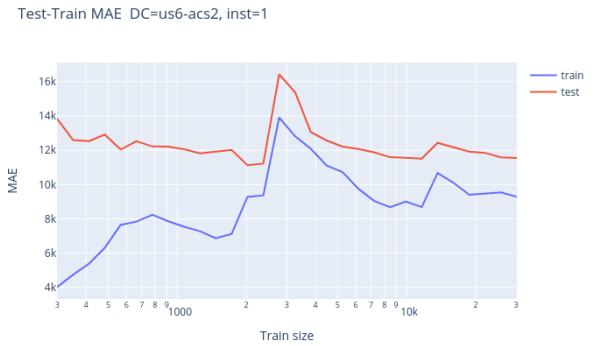


Рис. 6: Зависимость ошибки на тестовой и обучающей выборке от размера обучения. В качестве ошибки для минимизации использовалось MSE. dc=us6-acs2, inst=23

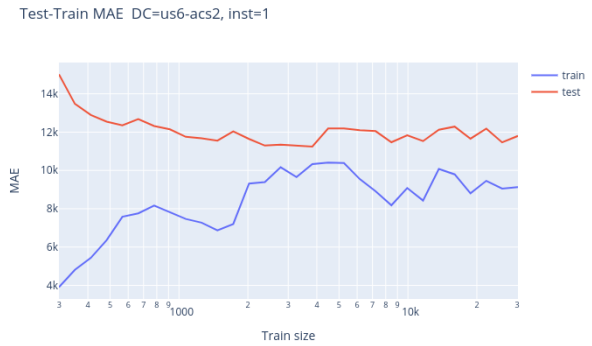


Рис. 7: Зависимость ошибки на тестовой и обучающей выборке от размера обучения. В качестве ошибки для минимизации использовался Hubert Loss. dc=us6-acs2, inst=1

Как видно на рис. 6, выброс портит не только ошибку на обучении, но и на отложенной выборке. Это показывает необходимость либо сменить метрику, либо удалить выбросы из обучающей выборки. В 7 нужны предыдущие значения и удаление одной точки приведёт к невозможности осуществить качественные расчёты для множества других. Другой интересный подход к такой проблеме - использование не `MSE`, а `Hubert Loss`, что представляет из себя параболу, продолженную линейно после некоторого ε. Решение оптимизационной задачи в сформированном признаковом

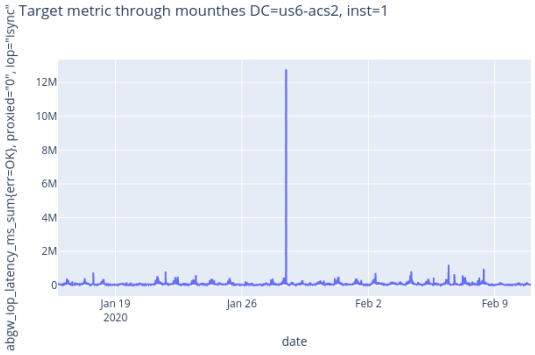


Рис. 8: Единичный выброс, dc=us6-acs2, inst=23

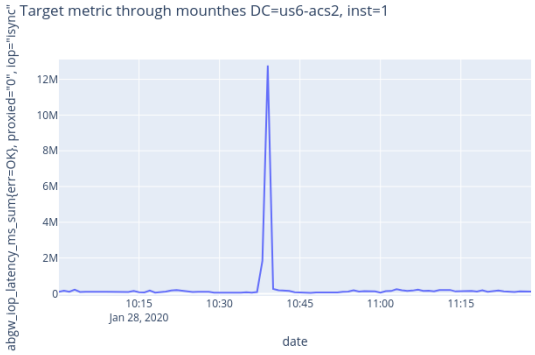


Рис. 9: Единичный выброс, увеличенный dc=us6-acs2, inst=1

пространстве `X` в 8, где w - обобщённый вектор параметров ϕ и θ, $alpha$ - коэффициент регуляризации. y - целевая переменная. Параметр $\varepsilon = 1.35$ устанавливается по умолчанию.

$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_\epsilon \left(\frac{X_i w - y_i}{\sigma} \right) \sigma \right) + \alpha ||w||_2{}^2 \tag{8}$$

$$H_\varepsilon(z) = \begin{cases} z^2, & \text{if } z \leqslant \varepsilon \\ 2\varepsilon|z| - \varepsilon^2, & \text{otherwise} \end{cases}$$

Результат использования такой оптимизационной задачи можно видеть на рис.7 Далее в работе коэффициенты для модели находятся решением 8. Возвращаясь к размеру обучающей выборки, стоит отметить, что значимое уменьшение перестаёт наблюдаться при размере обучения около полутора дней. Скорость вычислений и представленный объём данных позволяют выбрать этот размер обучения равный трём дням.

Следующим шагом логично поставить рассмотрения использования `p > 1`. Качество модели в при подборе гиперпараметра, в этой работе, предлагается оценивать с помощью валидации. Для временных рядов в процессе валидации не должно происходить тестирование в момент, предшествующий обучению. Предложена следующая техника:

1. Зафиксировать размер обучения и тестирования
2. Тестовую выборку выбрать как продолжение обучающей.
3. Обучить модель и протестировать. Оценивать качество при помощи

MAE.

4. Сдвинуть тестовую и обучающую выборку на минимум из размера обучения и теста.

Результат такого перебора для p=3 представлен на рис.10. Имея e_1 ошибку при p=1 и вычислив e_2 ошибку при другом p на графике отображёно $\frac{e_2-e_1}{e_1}$.

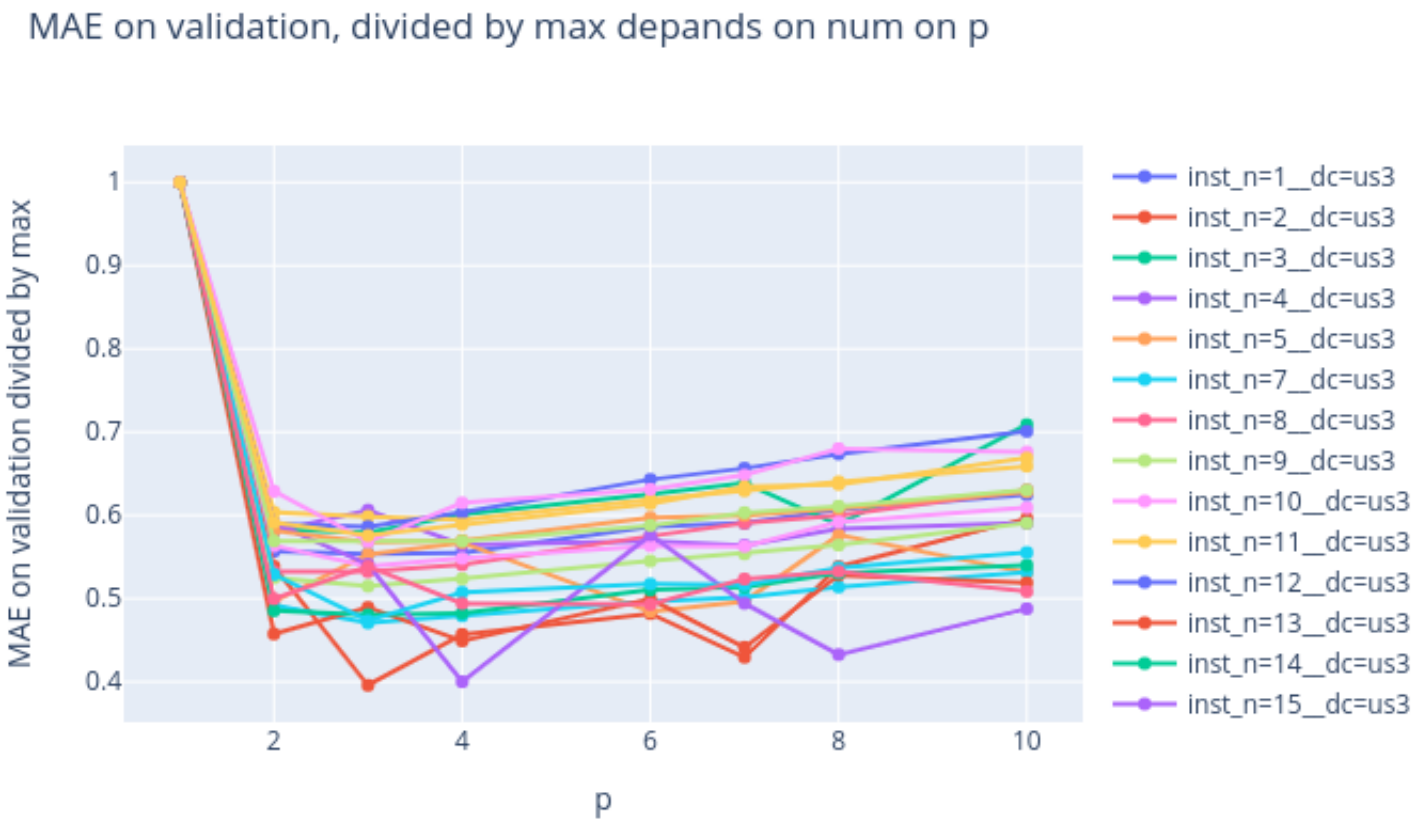


Рис. 10: Доля уменьшения MAE на валидации для us3

Почти для всех интсансов характерно следующее поведение - резкое улучшение качества про добавлении одного лага. Все дальнейшие добавления не несут пользы, а лишь зашумляют модель. Выбор p=2 позволяет допустить значимое увеличение ошибки и не допустить переобучения.

6.3 Отбор метрик

После получения первого приближения качественной модели можно преступить к отбору метрик. Это позволит побороть переобучение в модели и, возможно, использовать больше лагов.

Основной подход предполагает определение универсального набора метрик для всех инстансов. Полная процедура следующая:

1. Для каждого инстанса с помощью Lasso регуляризатора отбирается число признаков. Критерий выбора числа - отсутствие значимого увеличения точности при добавлении ещё одного признака.

2. Выбор числа признаков универсально для всех инстансов. Оказывается, что искомое число признаков практически совпадает.

3. Для каждого инстанса строится модель на выбранных метриках.

4. По полученным коэффициентам регрессии происходит отбор метрик - выбираются метрики с наибольшей по модулю границей разброса. Если $coeffs$ - это набор коэффициентов регрессии для выбранного признака, то сравниваются $abs(coeffs.mean()) + coeffs.std()$

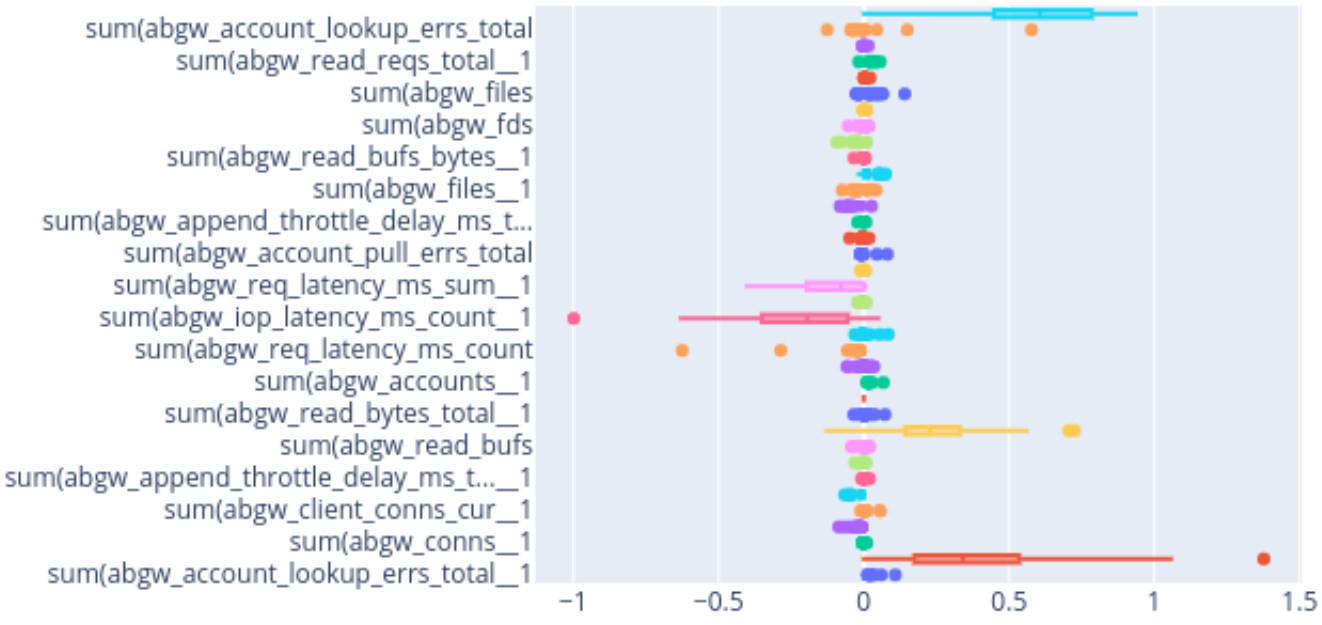


Рис. 11: BoxPlot для коэффициентов регрессий

На рис.11 представлен разброс метрик и легко можно выделить самые значимые. Тем не менее представленные результаты не совсем результат отбора. Для начала коэффициенты получены при построении модели с Lasso регуляризатором и ошибкой MSE, в то время как предложено было использовать HubertLoss. Кроме того Lasso отбирала признаки, а не метрики. Для полной уверенности в результатах необходимо посмотреть как меняется точность модели при различном числе метрик. Результаты уменьшения ошибки представлены на рис.12.

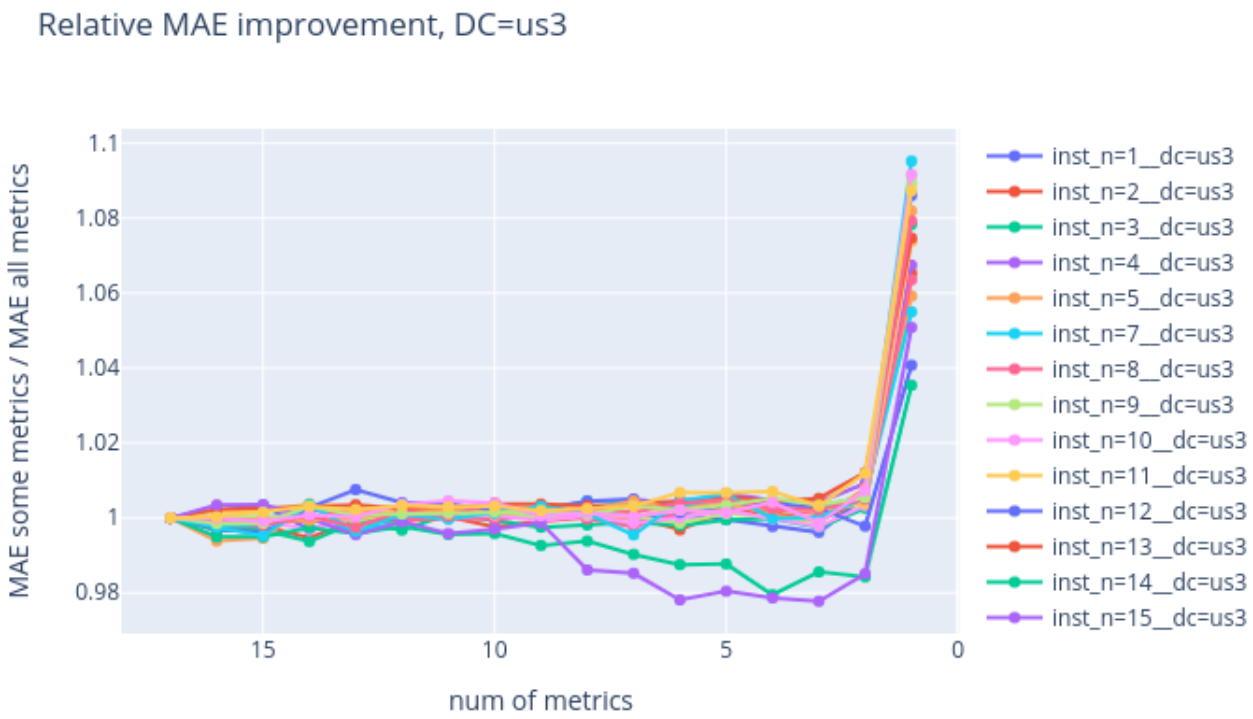


Рис. 12: Результаты уменьшения ошибки для различного числа метрик на us3

Видно, что изменение качество происходит совсем незначительное, поэтому имеет смысл оставить только лишь 4 метрики.

Хотелось бы отметить, что принципиально важно провести отбор признаков до построения `ARMA` модели. Уменьшение размерности пространстве позволяет не сильно переопределить систему. При этом вместо не информативных признаков и их лагов в модели могут использоваться большие лаги информативных признаков и лагов ошибок.

На рис.13, 14 можно наблюдать разницу влияния параметра `p` до и

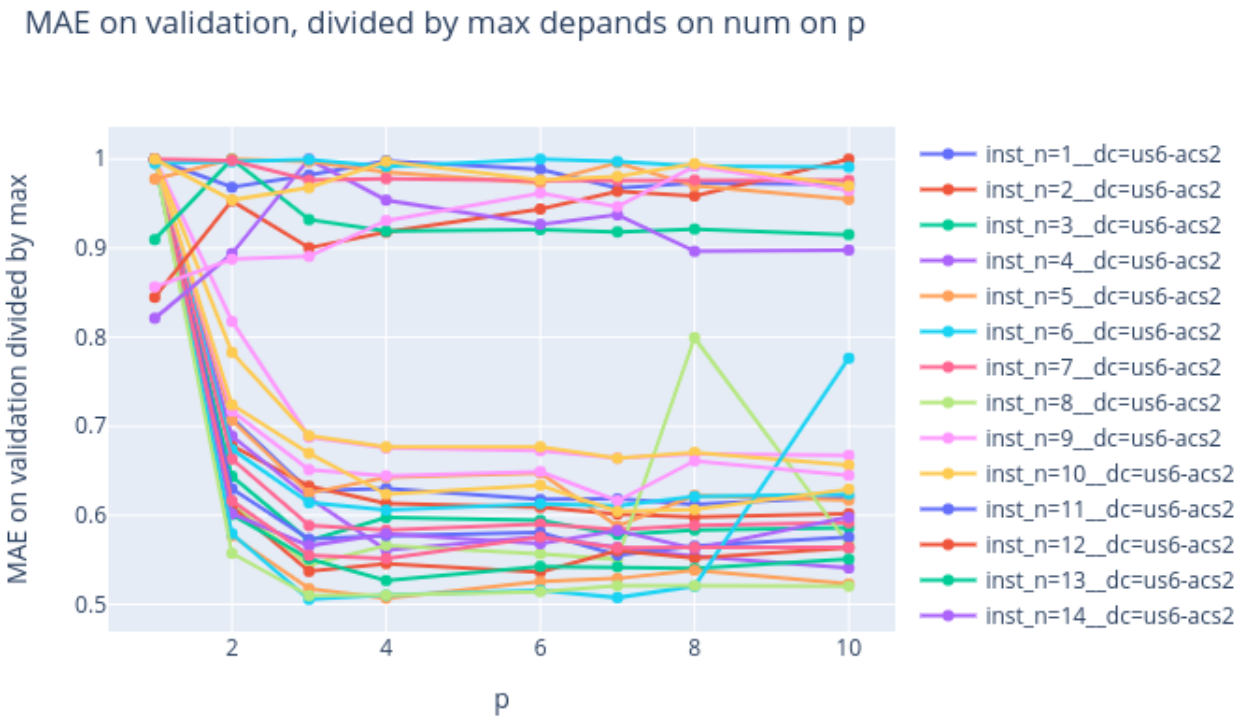


Рис. 13: Уменьшение ошибки в зависимости от параметра `p` после отбора признаков

после отбора признаков.

6.4 ARMA(p, q)

В этой части предлагается провести исследование `ARMA(p,q)` 9 модели.

$$\hat{y}_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{9}$$

Важно обсудить оценку обучение модели и применение.

При построении модели нужно учитывать шум ε на предыдущих участках. Он не наблюдается, можно оценить построенной AR моделью на первом шаге, а затем несколько раз уточнить. Подобный подход, как самый наивные, описывается в [4]. Количество уточняющих повторений обозначим за k.

Ниже сформулирован подход к обучению модели ARMA:

1. Построить AR модель, с её помощью оценить коэффициенты для мет-

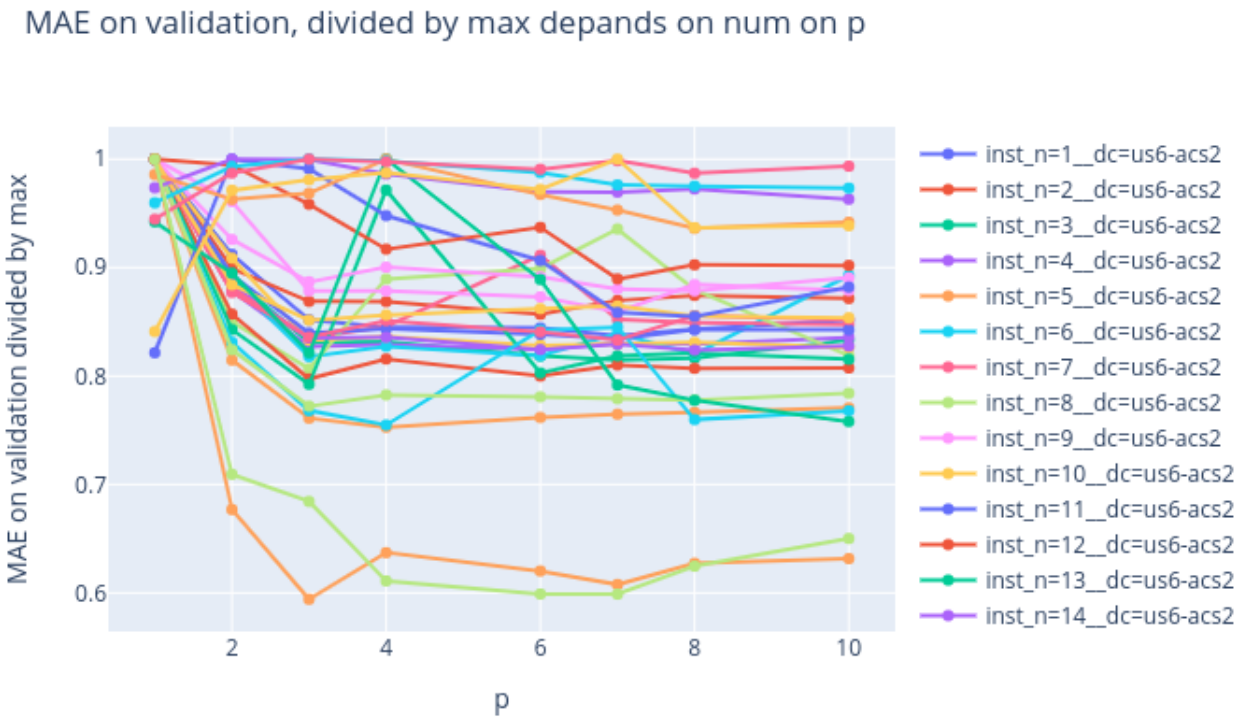


Рис. 14: Уменьшение ошибки в зависимости от параметра `p` до отбора признаков

рик их лагов.

2. Оценить ошибку при помощи AR модели и добавить её и её лаги к признаковому описанию.

3. Построить новую модель, учитывающую ошибку. Подсчитать ошибку и обновить её.

4. Повторить предыдущий шаг k_train раз.

Ниже сформулирован подход к применению обученной модели ARMA:

1. К признаковому описанию добавить q признаков, обозначающих ошибку. Их определить нулями.

2. Построить регрессионную модель и оценить ошибку и её лаги.

3. Построить новую модель, учитывающую ошибку. Подсчитать ошибку и обновить её.

4. Повторить предыдущий шаг k_test раз.

Осуществить подбор k_{test} и k_{train} можно, проанализировав изменения коэффициентов модели и ошибки.

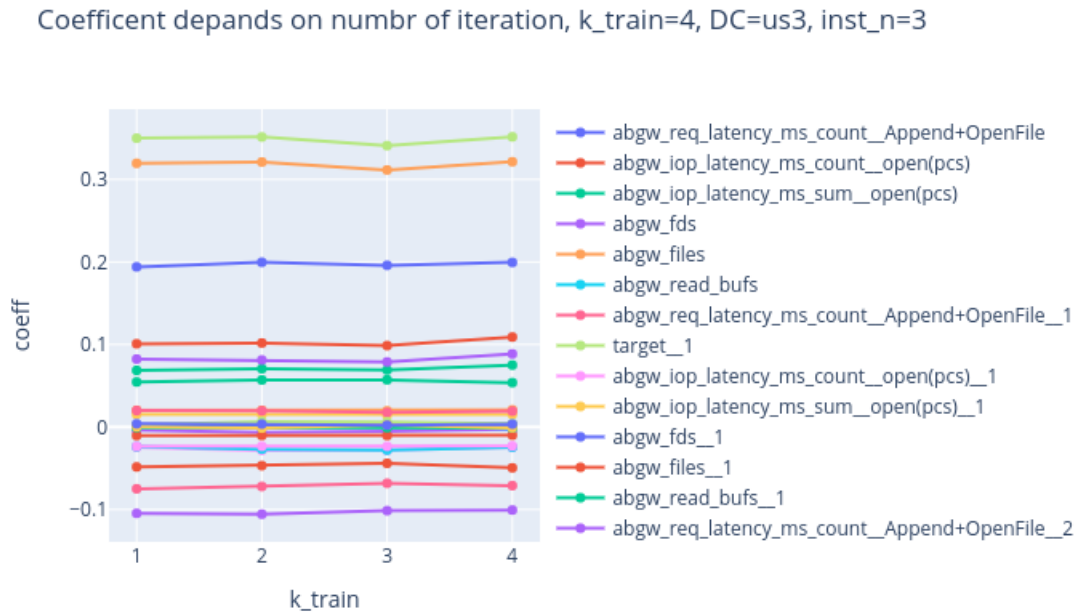


Рис. 15: Зависимость коэффициентов регрессиии от числа итераций при построении ARMA

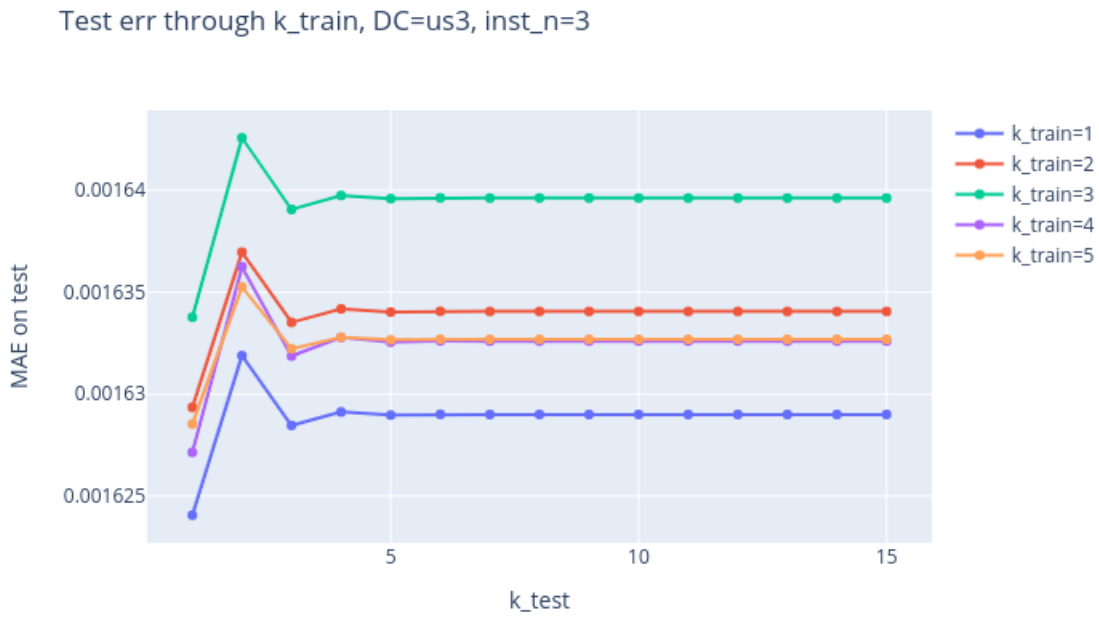


Рис. 16: Зависимость точности от числа итераций ARMA при применении

Такой подбор представлен на рис.15, 16. Применение ARMA не требует большого числа итераций - хватит 4 для применения и 2 для прмиенения. Стоит отметить, что предложеные изменения не дают большого уеличения точности, поэтому следующим шагом предлагаю сравнить модель с подобранным `q`, k_{test}, k_{train} с первой моделью `AR`.

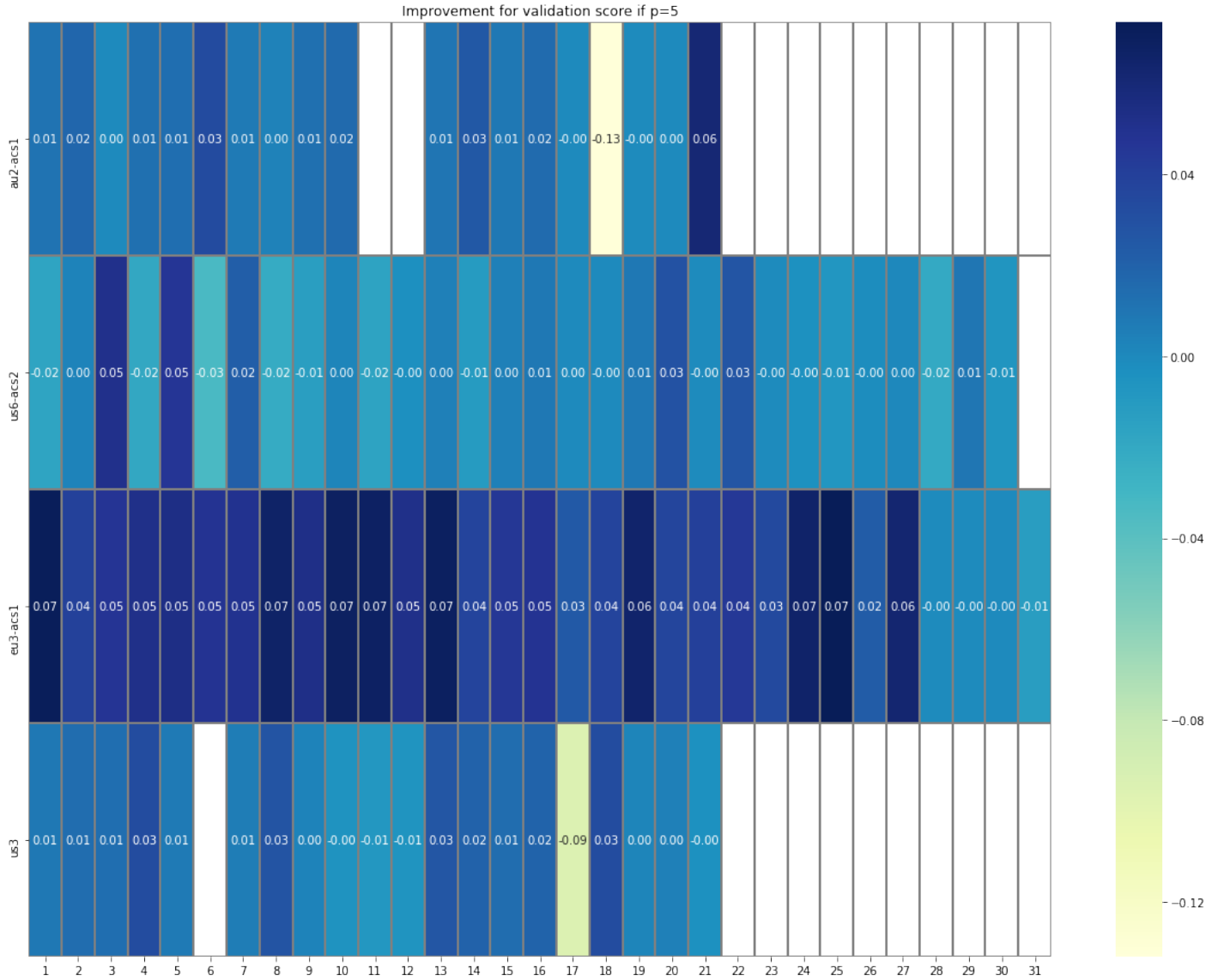


Рис. 17: Относительное умнтшение MAE при испоьзованиие ARMA модели в сравенинии с AR для каждого инстанса

На рис. 17, 18 представлено относительное уменьшение `MAE` на валидации при выборе `ARMA` модели. Таким образом, можно отметить, что значимого улучшения не наблюдается на большинсве инстансах, однако на некоторй группе оно есть значительное. Поэтому имеется смысл в исполь-зовании `ARMA` модели.

На текущем этапе исследования модели также имеется возможность

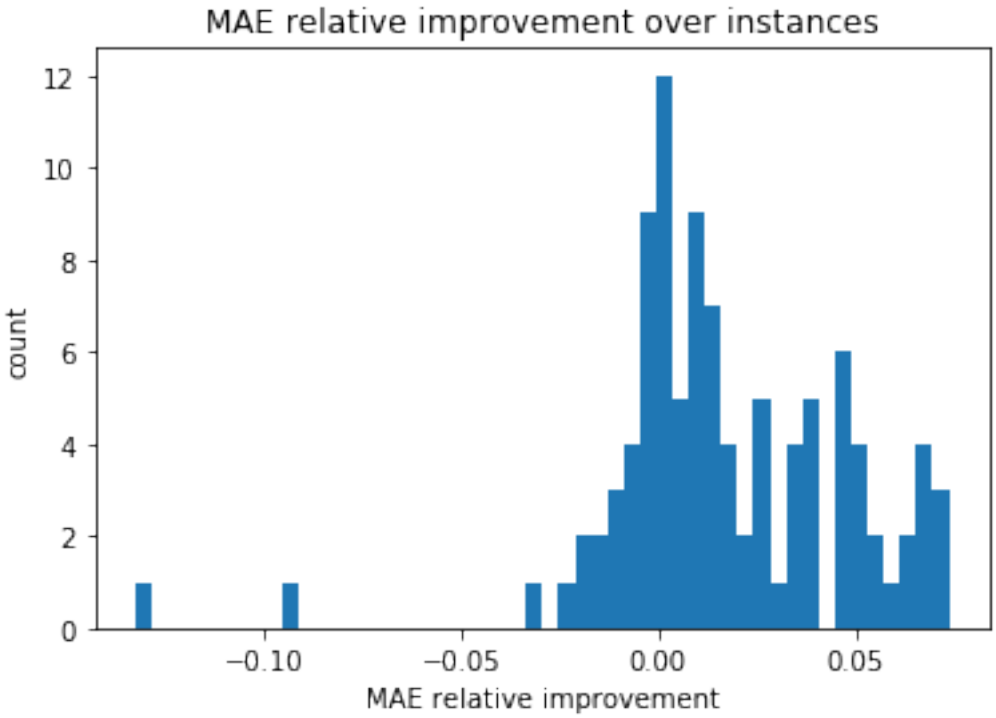


Рис. 18: Относительное умнтшение MAE при испоьзованиие ARMA модели в сравенинии с AR

рассматривать модель при трёх важных гипрепараметрах: количество признаков, p, q.

6.5 Корректность модели

После построения предсказательной модели овзникает вопрос о том, корректна ли эта модель. Для этого нужно проверить слудеющие характеристики: распределние ошибки и её гетероскедастичность. Также нужно рассмотреть стационарность временного ряда ошибки. Для начала обсудим гетероскичность - независимость ошибки от целевой переменной. Самоё простое что можно сделать - посмотреть на ошибку в зависимости от целевой переменной. Стоит отметить наличие двух принципиально рзных случаев - AR(p$\neq$ 0) и AR(0).

Интересно, что на рис. 19 есть участок, соответствующий отрицательным ошибкам на участках с низкими значениями целевой переменной. На рис.20 такого участка не наблюдается. Это может быть обусловлено учётом лагов. Наличие выброса, кроме большой ошибки в текущий момент повлечёт ошибку при учёте выброса, как лага.

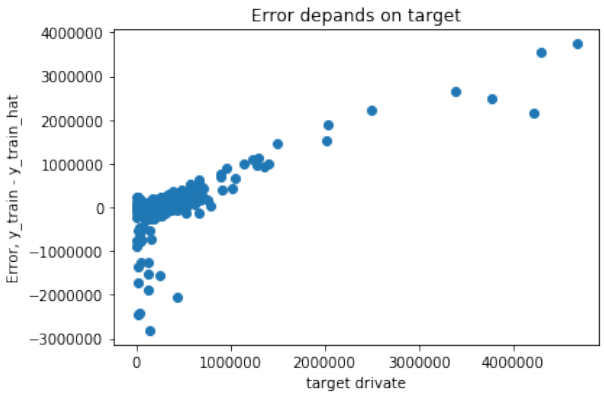


Рис. 19: Зависимость ошибки от целевой переменной в модели с учётом лагов

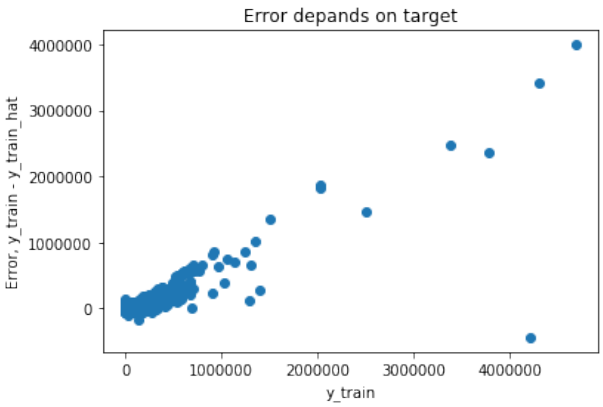


Рис. 20: Зависимость ошибки от целевой переменной в модели с без учёта лагов

Тем не менее признак некачественной модели - зависимость ошибки от цлелевой переменной. Одной из причин такого поведения может быть неверное масштабирование, ведь при маштабировании по формуле (2) основная масса значений оказывается около нуля из-за наличия выбросов. Таки образом ошибка `Hubert` почти не настраивается на большие значения ошибок и целевой переменной и при модель получается в некоторм смысле смещённой. Это происходит изза того, что распределение целевой переменной не обладает хоть какой-нибудь симметрией. Один из сопсобов расрпеделить влеичины более равномерно - исопльзовать преобразование Yeo-Johnson (3).

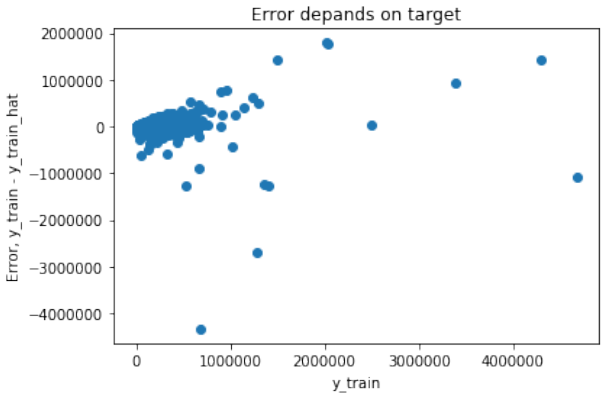


Рис. 21: Зависимость ошибки от целевой переменной в модели с учётом лагов и преобразование (3)

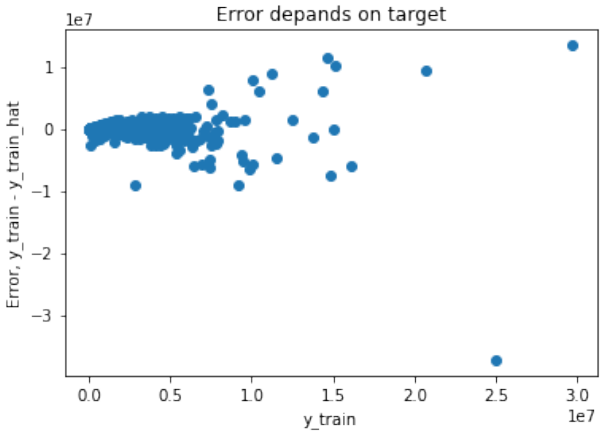


Рис. 22: Зависимость ошибки от целевой переменной в модели с учётом лагов, усреднённая за 5 мин

На рис.21 видно, что после применения другого способа масштабирова-

ния ошибка стала меньше зависеть от цлеелвой переменной. Тем не менее заметна небольшая тенденция к возрастанию. Пока что такое поведение можно объяснить высокой дискретностью данных и резкимими перепадами. Если же избавиться от резких перепадов, нарпример усреднением за 5 минут, то такое возрастанние исчезает, как на рис. 22. Кроме того стоит отметить, что разброс ошибки сильно уменьшился в особые, систематически более загруженные периоды времени.

7 Заключение

В представленной работе произведё качественное и количественное исследование класса мделей `ARMA` для продукта Acronis Storage. Результатом исследования стало:

- Схемы выбора важных для описания системы метрик
- Линейная модель, предназначение котрой - быть инструментом для определения аномального поведения.

В ходе исследования были применены:

- Различные подходы к масштабированию
- `LASSO` регуляризатор для отбора признаков
- Различные функции ошибки

Ни сезонностью, ни преобразовниями Yeo-Johnson добиться стационарности ряда про критерию `KPSS` не удалось. Также не удалось добиться достаточного уровня значимости для принятия гипотезы о стационарности ошибки. Тем не менее гистограммы и графики ошибки от времени показывают воодушевляющие результаты.

Список литературы

[1] *Prometheus - an open-source systems monitoring and alerting toolkit.* URL: https://prometheus.io/.

[2] *How do we know which test to apply for testing normality*
URL: https://www.researchgate.net

[3] *Sk Learn - Compare the effect of different scalers on data with outliers*
URL: https://scikit-learn.org

[4] *Lesson 12, Estimation of the parameters of an ARMA model*
URL: http://www.phdeconomics.sssup.it