Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет

«Высшая школа экономики»

Факультет компьютерных наук

Основная образовательная программа

Прикладная математика и информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Программный проект**

**На тему**

**“** **Автоматизированный пайплайн для прогнозирования показаний для объектов с разными характеристиками”**

Выполнила студентка:

Кудрявцева Софья Павловна, гр. БПМИ-172

Руководитель КР:

Приглашенный преподаватель ФКН, Соколовский Евгений Игоревич

**Москва, 2020**

**Оглавление**

[**Аннотация** 3](#_Toc71198919)

[**Ключевые слова** 3](#_Toc71198920)

[**Введение** 3](#_Toc71198921)

[**Обзор литературы** 5](#_Toc71198922)

[**Глава 1 – Программный цикл** 6](#_Toc71198923)

[**Описание программного цикла** 6](#_Toc71198924)

[**Описание модулей** 8](#_Toc71198925)

[**Руководство пользователя** 8](#_Toc71198926)

[**Глава 2 – Работа с данными** 8](#_Toc71198927)

[**Изучение данных** 8](#_Toc71198928)

[**Предобработка данных** 9](#_Toc71198929)

[**Устранение аномалий** 9](#_Toc71198930)

[**Интерполяция** 12](#_Toc71198931)

[**Кластеризация временных рядов** 12](#_Toc71198932)

[**Глава 3 – Признаковое пространство** 13](#_Toc71198933)

[**Создание признакового пространства** 13](#_Toc71198934)

[**Отбор признаков** 15](#_Toc71198935)

[**Глава 4 – Подбор модели** 17](#_Toc71198936)

[**Сравнение методов прогнозирования временных рядов** 17](#_Toc71198937)

[**Автоматизация выбора алгоритма** 19](#_Toc71198938)

[**Перебор гиперпараметров** 19](#_Toc71198939)

[**Глава 5 – Калибровка модели** 19](#_Toc71198940)

[**Географическое масштабирование** 20](#_Toc71198941)

[**Тестирование** 20](#_Toc71198942)

[**Заключение** 20](#_Toc71198943)

[**Список источников** 21](#_Toc71198944)

[**Приложения** 23](#_Toc71198945)

[**Приложение A** 23](#_Toc71198946)

[**Приложение Б** 25](#_Toc71198947)

# **Аннотация**

Моделирование временных рядов - широко известная и востребованная в современных реалиях задача. Для каждой компании важно планировать ближайшее будущее, чтобы выстроить эффективную бизнес-политику, ошибки в моделях прогнозирования напрямую влияют на прибыль компании. Особо важна задача прогнозирования потребления тех или иных ресурсов, и уменьшение потребления энергии за счет внедрения в организацию энергосберегающих приборов и проведения энергосберегающих мероприятий.

**Annotation**

Time series modeling is a widely known and demanded task in modern realities. For each company, it is significant to plan the nearest future in order to build an effective business policy since errors in forecasting models directly affect the company's profits. The task of predicting the consumption of certain resources and reducing energy consumption by introducing energy-saving devices into the organization and implementation energy-saving measures is especially important.

# **Ключевые слова**

Временные ряды, авторегрессионные модели, стационарность, аномалии в данных, кластеризация временных рядов.

# **Введение**

Энергосбережение ресурсов является довольно острой необходимостью не только в рамках отдельной компании, но и государства в целом. Проверка эффективности энергосберегающих мероприятий возможна за счет сверки прогнозируемого и фактического потребления. В связи с цифровизацией, сбором и хранением информации в государственных системах, становится возможен анализ энергопотребления у зданий, прогнозирование и корректировка политики энергосбережения. Однако, прогнозирование и анализ потребления различных ресурсов зданий в государственном масштабе невозможен без программных модулей.

Прогнозирование временных рядов является известной задачей машинного обучения. Однако поступление новых данных для исследователя зачастую является проблемой, которую приходится решать заново. При прогнозировании потребления ресурсов новые данные приходят как в разрезе зданий, так и ресурсов. Новые здания, которые подключаются к программам цифровизации и сбора показаний поставляют новые сведения по различным ресурсам, и для них необходимо уметь автоматически подбирать хорошую модель. Мероприятия, направленные на снижение энергоэффективности могут вносить разладки во временные ряды, и требуется вмешательство программиста, для актуальной смены модели, либо требуется достаточно много времени, чтобы текущая модель подстроилась под новые данные.

Целью работы является автоматизация процессов по анализу и прогнозу поступающих данных. Данная работа решает проблему прогнозирования потребления различных ресурсов для разных зданий. Автоматизация процессов прогнозирования поставила перед нами ряд задач.

• Первым направлением нашей работы являлось изучение имеющихся данных с временной структурой и последующая автоматизация их анализа и предобработки. Это направление включает в себя интерполяцию данных, определение и замена явных ошибок в данных, произошедших из-за ручного ввода показаний либо других аномальных значений, которые будут мешать построению стабильной модели.

• Немаловажным фактором влияния для подобного рода задач является корректность и эффективность используемых методов конструирования факторов и отбора значимых факторов.

• Следующим направлением исследований был автоматический подбор модели и ее гиперпараметров.

• Также принимается во внимание возможность необходимости смены модели или внеочередной рекалибровки модели в случае возникновения разладок в прогнозируемом временном ряде, обнаружение которых необходимо реализовать.

Решение вышеописанных задач позволяет создать полностью автоматизированный пайплайн, способный без вмешательства человека в условиях стабильности временной структуры данных строить модели для прогнозирования возможного потребления ресурсов в будущем, упрощающая ведение статистики и обработку данных.

**Цель проекта:**

Создание автоматизированного пайплайна, способного спрогнозировать модель потребления различных ресурсов зданиями.

# **Обзор литературы**

На текущий момент существует множество уже реализованных подходов к решению задачи автоматизации прогнозирования. Несколько примеров таких решений были рассмотрены в рамках изучения релевантных публикаций.

На языке R реализованы библиотеки auto.arima (ARIMA) и ЭТС (экспоненциальное сглаживание). Существует также реализация auto.arima на языке Python – pyramid [1]. Для сравнения моделей обычно используются информационные критерии (AIC, BIC, AICc и т. д.). Однако, эти критерии обычно применяются для оценивания моделей только из одного семейства.

TBATS – интересный подход, способный моделировать временные ряды с несколькими сезонностями. Название является аббревиатурой для ключевых функций моделей: тригонометрическая сезонность, преобразование Бокса-Кокса, ошибки ARMA, трендовые и сезонные компоненты. Основана на методах экспоненциального сглаживания. Каждая сезонность моделируется тригонометрическим представлением, основанным на рядах Фурье. Модель сравнивает несколько моделей и выбирает лучшую критерием Акаике. На наш взгляд большим минусом этой модели является невозможность добавить экзогенные факторы, а также достаточно долгий подбор лучшей модели.

Prophet – библиотека от Facebook [2], в основе которой лежит предположение, что наиболее полезные временные ряды реального мира не содержат никакой структуры, выходящей за пределы тренда, сезонности и причинно-следственных связей, и в отличие гауссовских линейных авторегрессий (таких как ARIMA, SARIMA, SARIMAX) не пытается использовать наследственность информационного потока. Prophet часто используют для прогнозирования различных временных рядов, он был создан в попытке найти баланс между автоматизацией, гибкостью и интерпретацией.

Используя вышеописанные подходы автоматического прогнозирования, приходится соглашаться с лучшей «в среднем» моделью, но нет никаких гарантий, что в отдельно взятом случае, эта модель будет выдавать хороший прогноз без каких-либо предположений о данных. Также, в них отсутствуетобработка выбросов, которые могут значительно зашумлять данные, и пропущенных значений, что может отрицательно влиять на прогноз.

Если рассматривать не статистические методы, а методы машинного обучения, то возникает более общая задача автоматизации машинного обучения. Для данной цели существует несколько open source (light automl), а также коммерческих разработок (Google Vizier []). Эти решения обычно не акцентируют внимание на прогноз для данных с временной структурой. В разделе тестирования мы исследуем, получится ли у нас автоматизировать процесс прогнозирования лучше, чем имеющиеся решения.

# **Глава 1 – Программный цикл**

## **Описание программного цикла**

Автоматизация прогнозирования потребления подразумевает под собой автоматическое прохождение программного цикла от получения первичных данных до предоставления результата. Можно выделить несколько основных ступеней, которые необходимо реализовать:

1. Обработка полученных данных. Первичная работа с данными включает в себя: выявление выбросов и их корректировка, при необходимости дополнение пропущенных значений, выявление значимых свойств полученного временного ряда.
2. Работа с признаковым пространством. На этом этапе мы формируем набор признаков, которые будут необходимы для прогноза в 2 этапа: выявляем всевозможные факторы, которые могут влиять на потребление ресурсов и затем отбираем из них наиболее значимые.
3. Выбор модели и гиперпараметров для прогнозирования.
4. Калибровка модели. В связи с постоянно меняющимися и дополняющимися данными, чтобы избежать его устаревания, необходима периодическая донастройка пайплайна. Она может быть плановой и внеплановой при выявлении разладки в данных.

На вход пайплайна подаются внешние факторы, данные по всем имеющимся рядам и по ряду, по которому необходим прогноз, период для прогноза. Пайплайн выдает прогнозные значения ряда на заданный период.



Рисунок 1. Схема пайплайна

Наш программный модуль должен уметь подбирать алгоритм для каждого временного ряда достаточно быстро, не перегружая основную программу. При большом объеме временных рядов подбор модели для каждого отдельного ряда становится вычислительно неэффективным. Зачастую, это решается выбором одной достаточно гибкой модели, которая в среднем дает удовлетворительный результат (Например, Prophet [2]). Однако в нашем случае, мы знаем примерные данные, с которыми придется работать модели, и можно разумнее подойти к выбору подходящей прогнозной модели. Так как поступающие данные имеют различную структуру – различные ресурсы и здания, то есть смысл разделять временные ряды на кластеры. Цель этого разделения – собрать внутри одного кластера максимально похожие временные ряды, и для каждого кластера подбирать свою прогнозную модель.

Для успешного прогнозирования необходимо иметь как можно больше показаний в прошлом. При получении ряда, за период ниже определенного порога, не получится выдать корректное предсказание. Однако, такие ряды будут также требовать какого-то прогноза. Для решения этой проблемы предлагается следующий алгоритм:

1. Разделить показания приборов учета ряды на 3 группы:

* приборы учета, по которым есть хотя бы суточные показания за год с пропущенными значениями не более 5%
* приборы учета, у которых данных для построения прогноза не менее 100 дней
* приборы учета, у которых менее 100 дней с известными показаниями

1. Первую группу временных рядов мы кластеризуем. Для каждого кластера будет подбираться своя модель прогнозирования.
2. Для временных рядов из второй группы мы подбираем максимальных похожий ряд и дополняем недостающие значения первого ряда.
3. . Временные ряда из третьей группы мы не прогнозировали из-за недостаточности данных.

## **Описание модулей**

Исходный код проекта описывается следующей схемой:

\conf

\conf\example.json.example – пример настроек доступа к базе данных.

\libs

\libs\[data.py](https://github.com/sonya123454/Automated-Pipeline-for-Predicting-Readings-for-Objects-with-Different-Characteristics/blob/main/libs/data.py) -

\libs\[libs.py](https://github.com/sonya123454/Automated-Pipeline-for-Predicting-Readings-for-Objects-with-Different-Characteristics/blob/main/libs/libs.py) – функции, необходимые для прогнозирования.

\libs\[my\_db.py](https://github.com/sonya123454/Automated-Pipeline-for-Predicting-Readings-for-Objects-with-Different-Characteristics/blob/main/libs/my_db.py) – функции для получения и записи данных в базу данных.

\experiments – файлы с расширением .ipynb, в которых можно увидеть код экспериментов, описанных ниже. Не обязательны для функционирования пайплайна.

\helpers – файлы с реализованными функциями для получения и сохранения различных дополнительных данных в базу данных. Не обязательны для функционирования пайплайна.

get\_pred.py - функции для получения и записи данных в базу данных.

start.py – файл, с функциями запуска пайплайна.

## **Руководство пользователя**

Для начала работы понадобится:

* Для запуска пайплайна необходимо запустить start.py файл.
* Для прогноза временного ряда нужно запустить файл get\_prediction.py с параметром id у строки в базе данных, отвечающей за запрос на прогноз временного ряда.

Для создания окружения с библиотеками, с помощью которых мы запускали пайплайн, вы можете найти в файле docs\libraries.txt.

Для того, чтобы использовать данный пайплайн для другого проекта, необходимо:

* Переопределить функцию get\_data в файл «data.py», которая должна возвращать все доступные временные ряды в этой задаче.
* Переопределить функцию get\_test в файл «data.py», которая получает необходимый временной ряд для прогнозирования.

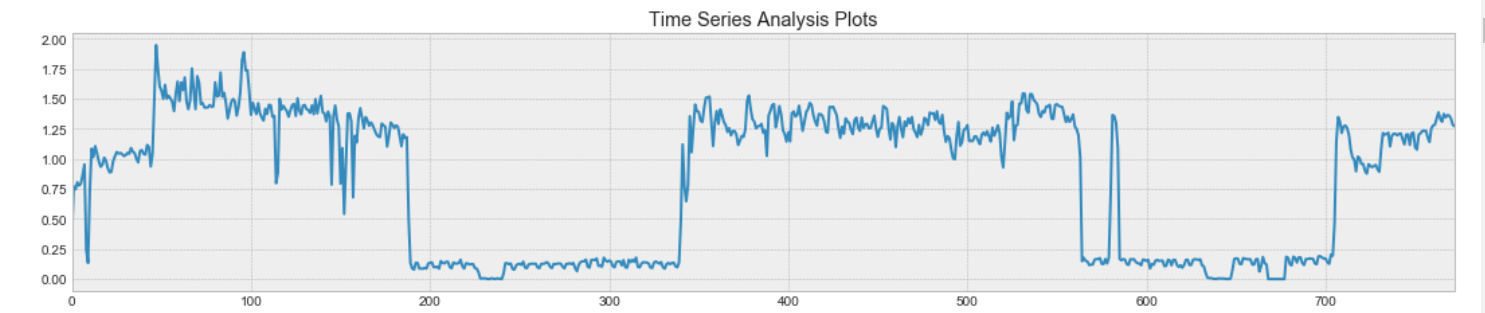
Остальные шаги аналогичны вышеописанным.

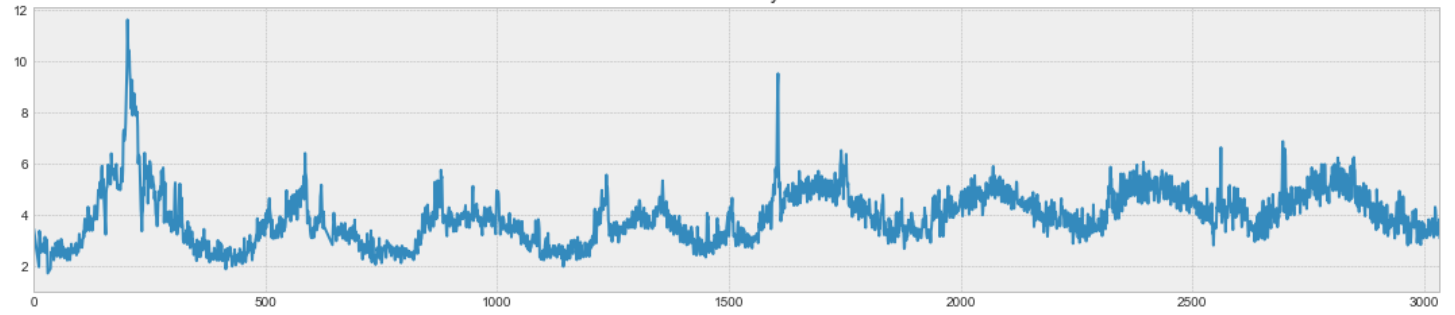
# **Глава 2 – Работа с данными**

## **Изучение данных**

Имеются данные по 3 ресурсам- тепловая энергия, горячая вода, электрическая энергия за различные промежутки времени. Большинство рядов умеют посуточные показания, 2 здания имеют почасовые показания. В рядах есть пропуски.

Во многих рядах визуально наблюдается 7-дневная и годовая сезонность (рис. 1 – тепловая и электрическая энергия), в некоторых виден тренд (рис. 1 – электрическая энергия).





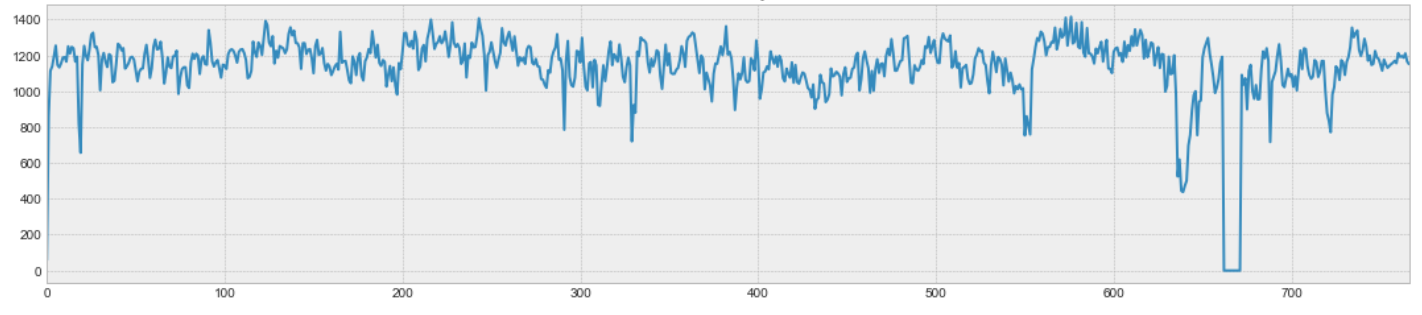


Рисунок 1. Сверху-вниз: Потребление тепловой, электрической энергии, потребление горячей воды.

Наши временные ряды не являются стационарными, имеют тренд и имеют сезонный компонент. Некоторые из них имеют еще и меняющуюся со временем дисперсию.

## **Предобработка данных**

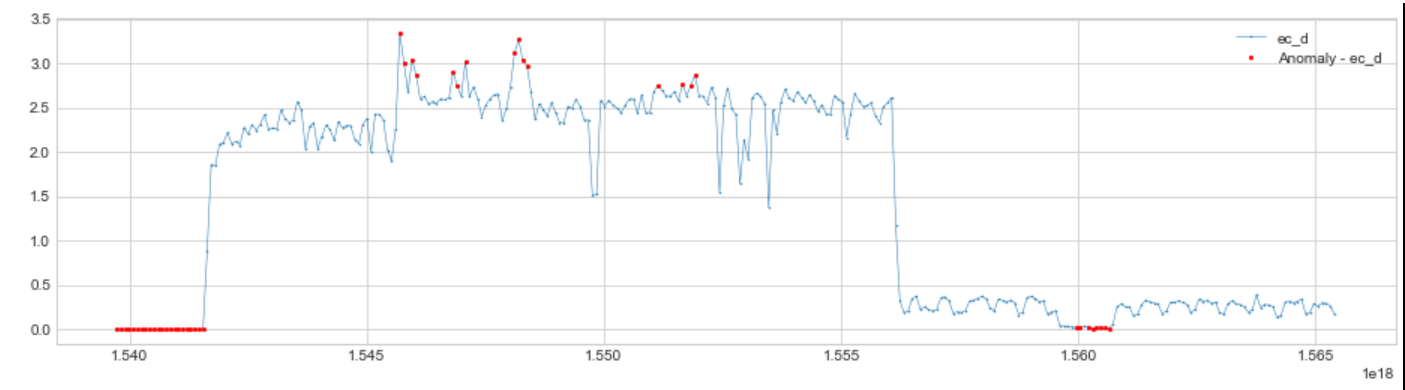
В модуле обработки данных мы проводили несколько последовательных процедур: 1) выявление аномальных значений и пропущенных данных 2) замена этих данных 3) кластеризация

Некоторые дома имеют почасовые показания. Для удобства мы у всех зданий приводим данные к посуточным.

## **Устранение аномалий**

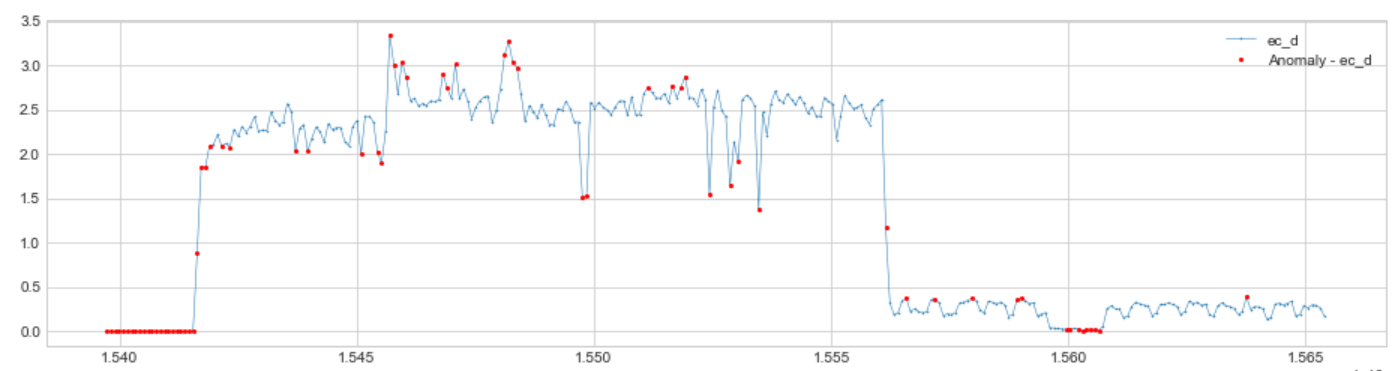
Для прогнозирования полезно исключить данные, которые неоправданно сильно выбиваются из общих данных. Исключаемые данные заменяются по тому же принципу, что и пропущенные. Аномальные значения могут иметь различную природу. Некоторые аномальные значения могут быть объяснены дополнительными данными, например аномально низкая температура может повлечь за собой аномально высокое потребление тепловой энергии. Цель определения аномальных значений в этой работе – это моделирование потребления в «нормальных условиях», исключение ложных показаний, которые возникли по различным причинам (ошибочный ручной ввод в систему), либо показаний, которые не могут быть объяснены в рамках имеющихся данных. Если оставить их в данных, эти выбросы будут мешать модели определить правильные зависимости в данных.

Сначала мы определяли аномалии, как значения, которые выходят за 5-95 квантили. Однако, на графике ниже, видно, что если определять аномалии таким способом, то упускаются аномалии с минимальными значениями в зимнем периоде и максимальными в летний. Поэтому будем делить ряд на 2 сезонные составляющие, и в каждой из них находить свои аномалии.



Для разделения ряда на летний и зимний отопительные периоды необходимо было определить значение для ряда, выше которого начинается зимнее потребление, а ниже – летнее. Для определения этого значения использовалась следующая схема. Строится гистограмма распределения с 10 численными интервалами (Строится не непрерывное распределение для того, чтобы сгладить функцию и привести ее в нужный вид). Видно, что самые частотные значения будут попадать примерно в начало и конец гистограммы (так как летнее потребление принимает минимальные значения, а зимние – максимальные, а переходный период – средние значения - самый краткосрочный), а сама гистограмма принимает вид “седла”. По этому графику несложно определить минимальное значение седловой функции, которое и будет точкой смены отопительного сезона.

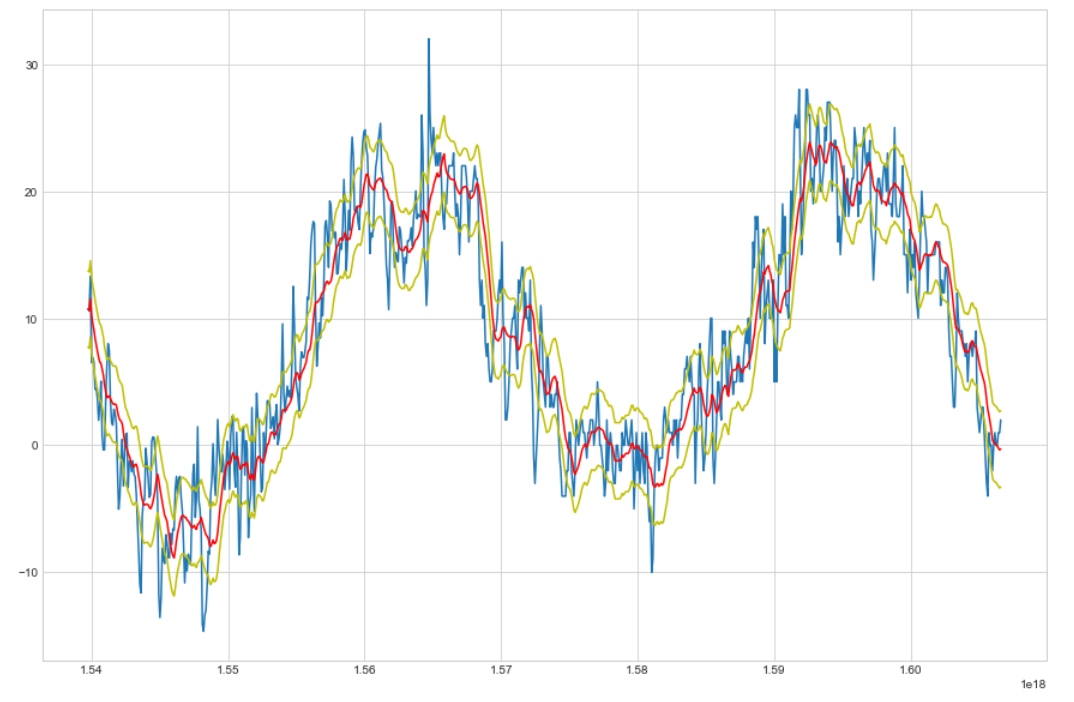
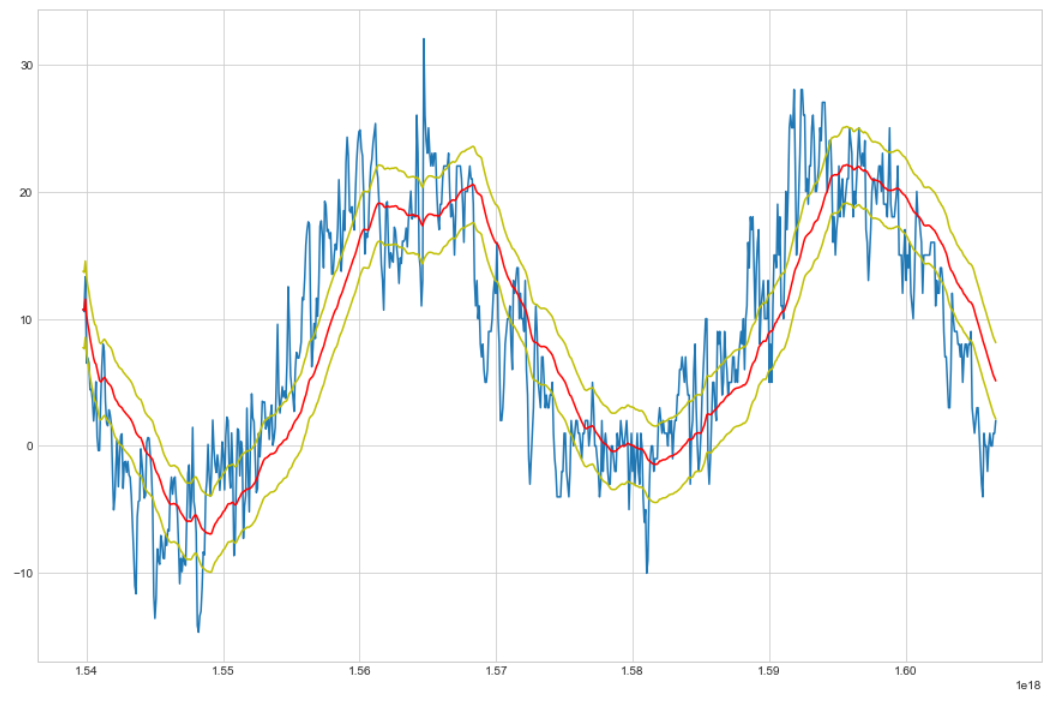
Иной способ отделения летнего и зимнего отопительного периода – это определение сдвига уровня значения, отслеживая разницу между медианными значениями в двух скользящих временных окнах рядом друг с другом. Работоспособны и реализованы оба метода.



После применения разделения отопительный периодов аномалии стали выявляться значительно лучше. Следующий шаг — это убрать те аномалии, которые хорошо описываются дополнительными данными и могут быть объяснены моделью.

Для того, чтобы убрать аномалии, которые могут быть описаны дополнительными данными, мы провели следующую процедуру. Для дополнительных данных находятся аномалии и аномальные даты у двух временных рядов сравниваются. Если даты аномалий совпадают, или аномалии потребления происходят с задержкой 1-2 дня, мы считаем, что аномальные значения объяснимы.

Выбросы у температурных данных считаются иначе, чем у потребления. Это связанно с тем, что температурные данные нельзя разделить на 2 составляющие, и выявлять значения, отклоняющиеся за 5-95 квантиль будет не правильным. Сначала мы определяли сезонную составляющую ряда, в идеальном варианте это должен был быть график, похожий на синусоиду. А затем, все показания, которые откланяются от этой кривой больше чем на 3 градуса, считаются аномальными. Сезонную составляющую мы определяли 2 способами – среднее скользящее окно и сглаживание ряда с помощью фильтра Савицкого–Голея. При получении сезонной составляющей простым сглаживанием, получается довольно рваный ряд (с большими изменениями за короткий промежуток времени) со смещением вниз или вверх. При попытке увеличить окно, получается сильное смещение. При сглаживании ряда с помощью фильтра Савицкого–Голея, сглаживание происходит с помощью свертки, путем подгонки последовательных подмножеств смежных точек данных полиномом низкой степени методом линейных наименьших квадратов. Результаты получаются более понятными, без резких скачков.

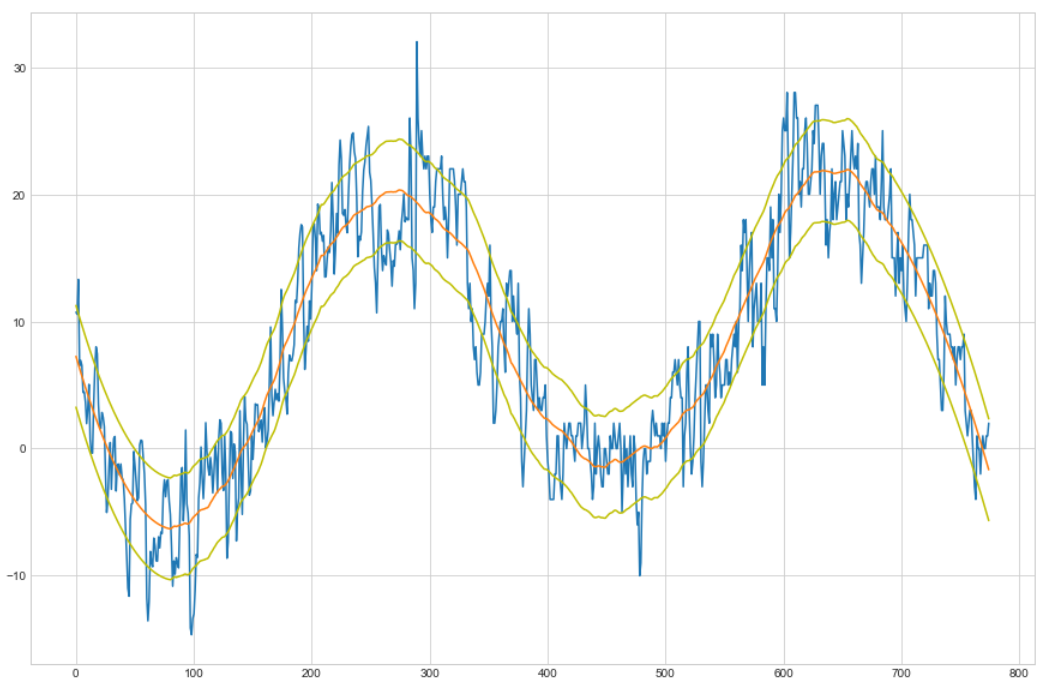
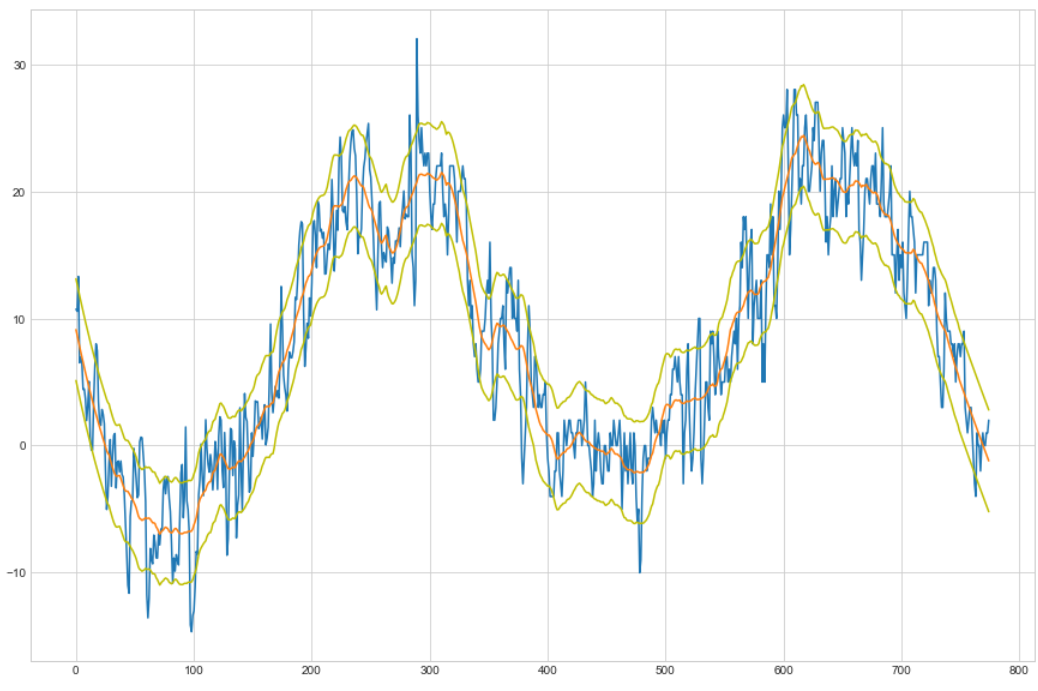


Рисунок 3. Сверху - среднее скользящее окно, снизу - сглаживание ряда с помощью фильтра Савицкого–Голея.

## **Интерполяция**

Для заполнения пропущенных данных у временных рядов из первой группы можно использовать несколько методов:

* Линейная интерполяция по двум соседним точкам.
* Скользящее среднее.
* Результаты предыдущего периода.
* Данные за аналогичный период в прошлые годы.

В нашем Pipeline реализованы все методы, по умолчанию используется линейная интерполяция.

## **Кластеризация временных рядов**

Необходимость кластеризации рядов возникает из-за 2 причин. Первая причина — это большое количества рядов, для каждого из которых необходимо подбирать прогнозную модель. Мы хотим проверить гипотезу, что можно выделить несколько кластеров, и для каждого кластера применять наиболее подходящую модель. Второй причиной является наличие временных рядов за слишком малый промежуток времени. Необходимо проверить предположение, что имея ряд за малый промежуток времени, можно найти максимально похожий ряд и строить прогноз, отталкиваясь и от его данных тоже.

Кластеризовать ряды можно 2 путями. Первый вариант - представлять ряд в виде вектора его различных характеристик (его тренд, сезонности, число значимых лагов в частичной корреляции) и кластеризовать уже новые объекты с меньшим количеством измерений. Этот вариант подходит для решения нашей первой задачи, для выделения групп временных рядов, для которых можно использовать одну модель. Второй вариант – сравнивать сами временные ряды, используя специальные метрики. Этот вариант больше подходит для решения второй задачи.

Остановимся подробнее на кластеризации временных рядов с помощью преобразования ряда в объект с меньшей размерностью. Показатели, которые мы выделили, для описания рядов:

- Тренд (летний / зимний). Определяется как линейный, с помощью линейной регрессии.

- Среднее летнее потребление. Этот показатель добавлен, так как наблюдение показало, что есть здания, у которых летнее потребление равно нулю. Для таких зданий летний прогноз будет константным.

- Частичные автокорреляции 7 лагов

- Совместная информация ряда с экзогенными факторами

После подсчета выбранных факторов у каждого ряда, мы кластеризовали ряды с помощью KMeans. Выделенные кластеры на тестовом запуске пайплайна можно увидеть в Приложении А.

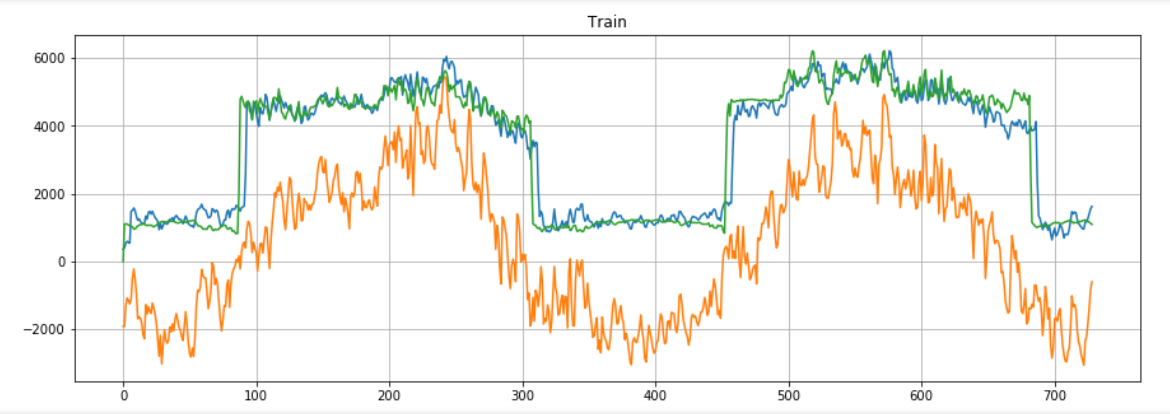
# **Глава 3 – Признаковое пространство**

## **Создание признакового пространства**

В качестве признаков, подающихся на вход модели, могут быть авто регрессионные факторы ряда, экзогенные факторы, и производные от экзогенных факторов. При создании признакового пространства мы старались не упустить любой фактор, который может положительно влиять на прогноз.

1. Авто регрессионные факторы ряда. Для предсказания потребления ресурса в определенную дату полезно узнать потребление за предыдущие даты. В начальном приближении мы взяли показания за предыдущий год до периода предсказания.
2. Экзогенные факторы. В этом разделе мы изучаем, какие внешние данные могут помочь для построения прогноза. Проверялось несколько вариантов: данные о погоде, производственном календаре, длине светового дня, день недели. Также была написана функция, которая добавляла бинарный признак “Отопление включено”. Признак “Отопление включено” зависит от признака «Температура». (По закону «Отопительный период должен начинаться не позднее и заканчиваться не ранее дня, следующего за днем окончания 5-дневного периода, в течение которого соответственно среднесуточная температура наружного воздуха ниже 8 градусов Цельсия или среднесуточная температура наружного воздуха выше 8 градусов Цельсия.»).

Для того, чтобы определить, какие данные нужно добавить для улучшения прогноза, а какие окажутся бесполезными, посмотрим на корреляцию между новыми временными рядами и целевым рядом.



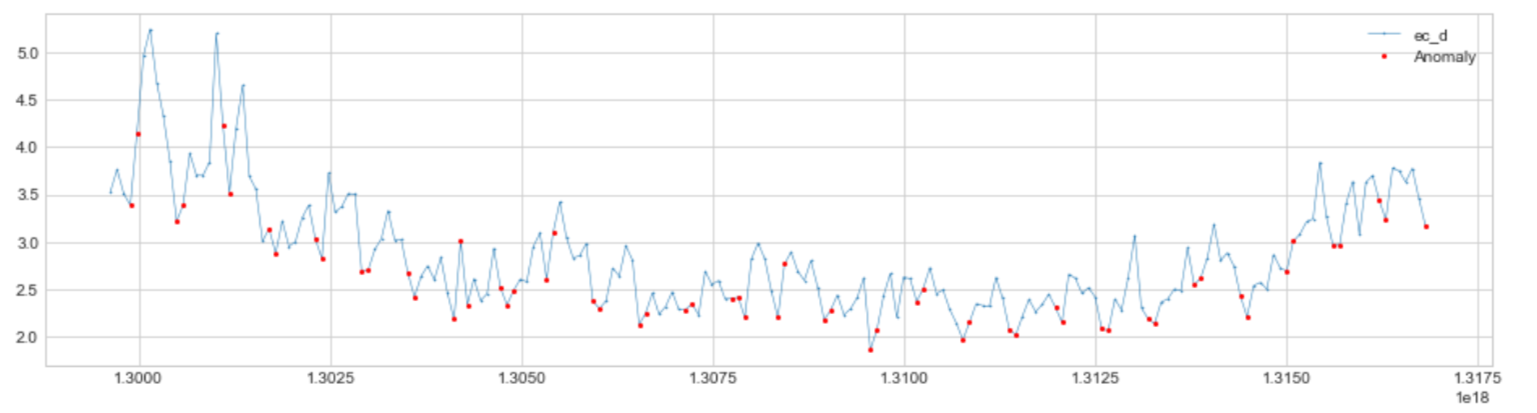
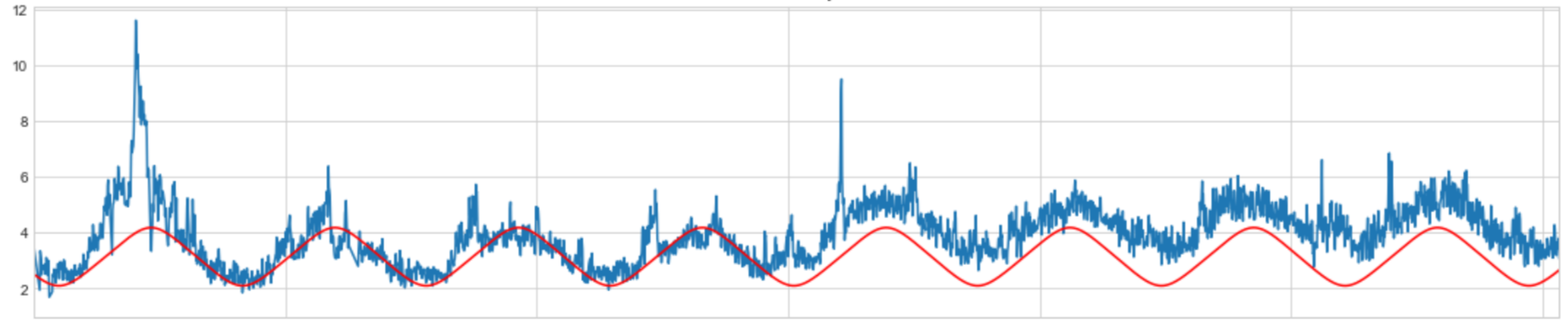


Рисунок 4. На верхнем графике изображено потребление горячей воды и температура. На графике температура умножена на -200 и сдвинута на 1000, чтобы были видны их относительные колебания. На втором графике показана зависимость между потреблением электричества и длиной светового дня. Длина светового дня умножена на -0.2 и сдвинута на 14. На нижнем графике показано потребление электричества, а красным выделены праздничные даты.

На верхнем графике можно увидеть, что температурные данные коррелируют с данными о потреблении горячей воды, пики потребления соответствуют температурным локальным экстремумам. На среднем графике видно, что потребление электроэнергии хорошо описывается преобразованной длиной светового дня (умножено на число и сдвинуто). На нижнем графике можно заметить, что праздничные даты часто совпадают с нижними экстремумами в потреблении. Все эти данные могут помочь в построении прогноза.

Некоторые дополнительные данные сами требуют некого прогноза на будущее. В отличие от длины светового дня, которая аналитически просчитана на каждый будущий день, температура в следующем месяце не так очевидна. Будущую температуру мы предсказываем отдельно. Так как это отдельная большая задача, учитывающая много факторов, для бейзлайна прогноза температуры мы взяли для каждого будущего дня усредненную температуру в этот день за прошлые года. В будущем можно загружать прогнозы с платных сервисов, либо написать для этого отдельный модуль.

1. Производные от экзогенных факторов. Внешние данные могут влиять на результат с некоторой задержкой. Например, резкое пятничное похолодание, может влиять на потребление тепловой энергии только в понедельник. Поэтому, мы добавили в признаковое пространство 7 лагов каждого из экзогенных факторов.

## **Отбор признаков**

Отбор значимых признаков необходим по нескольким причинам. Это скорость вычисления, как обучения, так и прогноза модели, также это точность прогноза – так как некоторые модели машинного обучения чувствительны к размеру входного вектора и отброс коллинеарных признаков, слабо влияющих на результат, поможет улучшить точность предсказания.

Способы отбора признаков: оберточный, встроенный, фильтрационный. Фильтрационные методы ранжируют признаки по степени влияния на целевую переменную. Плюсы этих методов в том, что время вычислений линейно зависит от количества признаков. Проблема этих методов в том, что не учитывается взаимное влияние друг на друга, каждый признак рассматривается изолированно.

В качестве фильтрационных методов мы использовали оценку корреляции и совместной информации признака и целевой переменной. Корреляция оценивает линейные взаимосвязи, и в отличие от совместной информации не работает на сложных, нелинейных зависимостях.

Изображение выглядит как квадрат

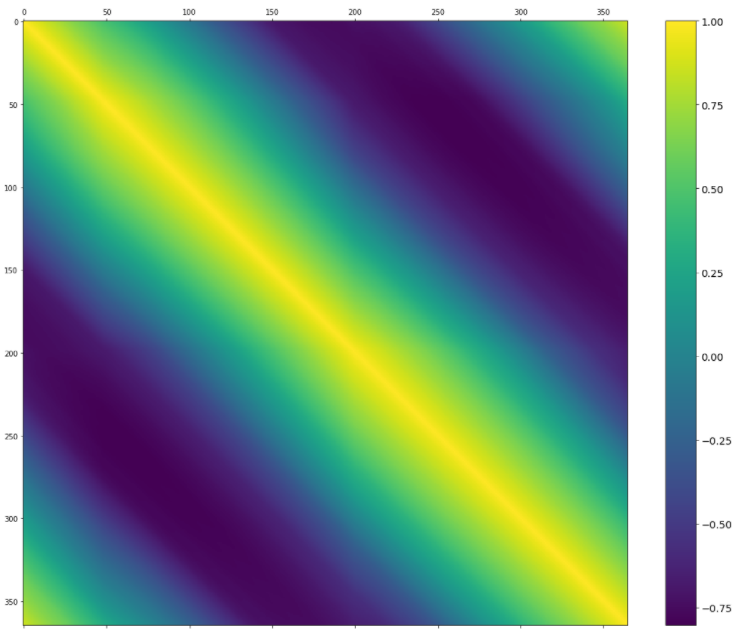
Автоматически созданное описание

Рисунок 5. Матрицы корреляций признакового пространства потребления тепловой энергии. Слева показана матрица корреляций последовательных 365 лагов временного ряда. Справа изображена матрица корреляций потребления с экзогенными факторами.

Оберточные методы рассматривают признаки как вход, а их качество прогноза – выход. Метод запускает прогнозную модель на различных включениях признаков и подмножество, на котором достигается лучшее качество, считается оптимальным. По перебору подмножеств методы разделяют на включающие и исключающие. Включающие методы начинают с пустого множества, добавляя в него различные признаки. Исключающие методы, наоборот, начинают с полного множества признаков. Оберточные методы самые затратные по времени обработки среди всех рассматриваемых. Также, при большом объеме тренировочного набора и большом количестве признаков, возможно переобучение. Взаимосвязи между признаками отслеживаются.

Встроенные методы не отделяют отбор признаков в отдельный шаг, отбор признаков происходит при обучении самой модели, например, с помощью L1 или L2 регуляризации. Возможно отбирать признаки, опираясь на древесные модели. Отбирая те признаки, которые в модели были наиболее важными для прогноза. Такой способ подходит для определения взаимовлияний признаков и требуют меньше времени, чем оберточные методы.

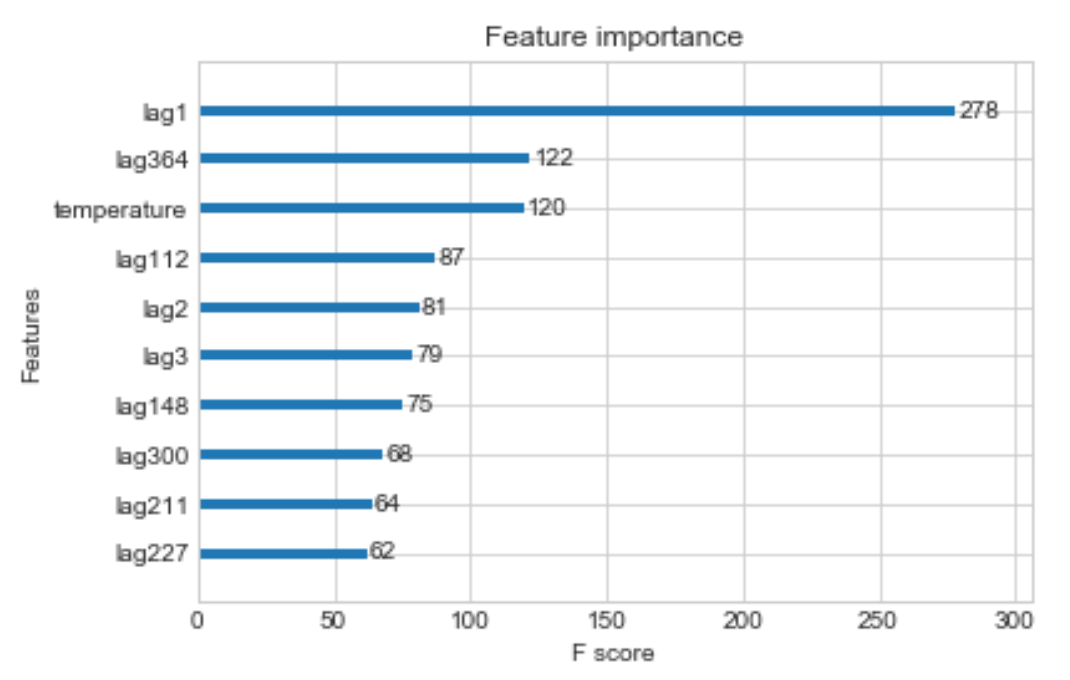


Рисунок 6. Значимость факторов во встроенном метода xgboost

Алгоритм отбора необходимо проверить признаков на «стабильность». Алгоритм считается «нестабильным», если небольшое изменение данных приводит к большим изменениям в выбранном подмножестве признаков. В нашем случае, важно, чтобы подмножество признаков не сильно менялось, либо вообще не менялось при изменении периода, за которое известно потребление ресурса у здания либо при изменении данных в рамках одного кластера временных рядов. Такой мерой «стабильности» алгоритма мы использовали меру из статьи S. Nogueira, K. Sechidis, G. Brown [8], которая обладает следующими свойствами: монотонность, везде определенность, поправка на случайность, существование единственного минимума, существование и достижение единственного максимума только при выборе одних и тех же признаков на различных выборках. Такие меры, как мера Kuncheva или индекс Джаккарда удовлетворяют не всем свойствам (Kuncheva не полностью определена, а индекс Джаккарда не имеет поправки на случайность).

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Где

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Группа | | Фильтрационные | | Встроенные | | Оберточные |  |
| Метод | Корреляция | | Совместная информация | Случайный лес | Линейная регрессия с L1/L2 | Включающий, метод наименьших квадратов | Ансамбль методов |
| Мера Nogueira | 0.45 | |  | 0.49 | 0.27 | 0.19 | 0.43 |
| Согласованность методов | 0.38 | | | | | |  |

Таблица 1. Сравнение методов отбора признаков.

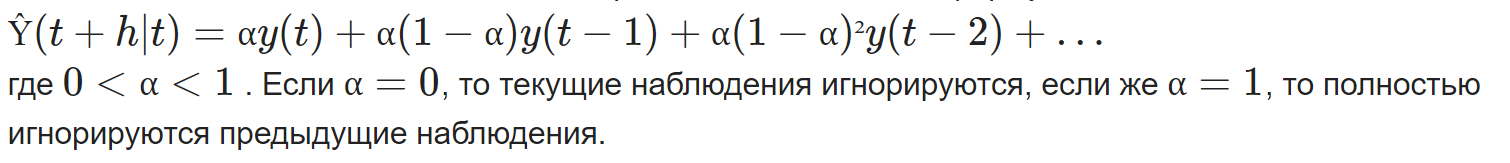
Также мы считали согласованность методов. Для этого также использовали меру Nogueira. Чтобы сравнивать методы, мы ограничили мощность набора итогового признакового пространства 10. Значение выбрано с запасом, исходя из корреляционных значений признаков с целевой переменной. Согласованность методов равна 0.38. Это говорит о том, что все методы выбирали примерно одинаковые признаки.

Также мы создали еще один “Ансамблевый” метод, который использует данные всех остальных алгоритмов. Для того, чтобы ошибка одной модели не влияла на конечный результат, метод сравнивает наборы, выданные разными методами. Признаки, выбранные наибольшее количество раз, входят в окончательный набор. Этот метод показал себя как достаточно стабильный и именно он принимает окончательный результат в выборе признаков.

# **Глава 4 – Подбор модели**

## **Сравнение методов прогнозирования временных рядов**

В данном разделе мы сравниваем различные методы и модели для прогнозирования временных рядов. Основные методы, использующиеся для моделирования временных рядов:

* Авто регрессионные модели
  + Экспоненциальное сглаживание. Прогноз описывается формулой:Подходит только для монотонных периодов, без смен характеристик ряда. Для рядов со сменами периодов отопления данный метод подходит плохо.
  + ARIMA, SARIMA, SARIMAX. Зависимость целевой переменной в авто регрессионных моделях описывается следующей формулой:Изображение выглядит как текст

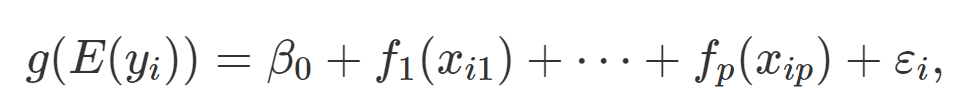
    Автоматически созданное описание

Минус авторегрессионных моделей в том, что они подвержены суммарному воздействию предыдущих ошибок.

* Динамические линейные модели

Отличие динамических моделей от авто регрессионных в том, что коэффициенты регрессии меняется по определенному закону, а не является константой.

* + TBATS. Это абревиатура от тригонометрической сезонности, преобразования Бокса-Кокса, ошибки ARMA, трендовых и сезонных компонент. Основано на экспоненциальном сглаживании, однако может взаимодействовать с несколькими сезонностями. На практике TBATS показал плохие результаты на наших рядах.
  + Prophet. Prophet «подгоняет» кривую тренировочного ряда, раскладывая его на сумму функций, зависящих от времени: тренд, сезонность, выходные дни (аномальные данные).
  + GAM. Цель GAM состоит в том, чтобы смоделировать переменную отклика независимыми переменными, которые представлены в виде некоторых гладких функций



* Древовидные модели.
  + Среди древовидных моделей лучше всего зарекомендовали бустеры. В этой работе мы взяли XGBoost

Мы отобрали несколько моделей, которые хорошо себя показали на разных рядах (см Приложение Б). Это оказались:

* SARIMAX. На рядах с потреблением тепловой энергией модель показывала неплохие результаты.
* GAM. Исследование показало, что эта модель подошла лучше всего на ряде с электрической энергией.
* Prophet. Точнее всего предсказала потребление горячей воды.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тепловая энергия | Электрическая энергия | Горячая вода |
| SARIMAX | 0.027\* | 0.29 | 126 |
| XGBoost с лаговыми признаками | 0.048 | 0.28 | 91.9 |
| XGBoost без лаговых признаков | 0.033 | 0.26 | 91.9 |
| Prophet | 0.112 | 0.32 | 84.6\* |
| TBATS | 0.030 | 0.43 | 104.1 |
| GAM | 0.029 | 0.24\* | 92.3 |

Таблица 2. Сравнение моделей на рядах с различными ресурсами.

## **Автоматизация выбора алгоритма**

Для сравнения моделей обычно используются информационные критерии (AIC, BIC, AICc и т. д.). Однако, эти критерии обычно применяются для оценивания моделей только из одного семейства. Для автоматического сравнения алгоритмов из одного семейства мы использовали информационный критерий Акаике (AIC), который штрафует модель за количество параметров и поощряет логарифмическое правдоподобие построенной модели.

Изображение выглядит как текст

Автоматически созданное описание

Для сравнения моделей из разных семейств мы использовали ошибку MAE. При этом мы предусмотрели возможность для пользователей программы использование кастомной ошибки, так как зачастую для разных задач в бизнесе потери могут описываться более сложными функциями. В частности, потери при меньшем предсказании целевой переменной, бизнес может терять намного больше, чем при большем предсказании, и наоборот.

## **Перебор гиперпараметров**

Настройка гиперпараметров для выбранной модели происходит поиском по сетке. Для SARIMAX мы подбираем p, q, d. Для Prophet – сезонность, для GAM – коэффициенты для регуляризации.

Обучение модели происходит с помощью кросс валидации.

# **Глава 5 – Калибровка модели**

Узким моментом в нашей модели является определение кластера для нового ряда. В случае, если ряд имеют новую структуру, модель для него не получится подобрать корректно. В таком случае мы предлагаем перезапускать пайплайн и новый ряд будет определен в новый кластер, для которого будет подобрана своя модель.

Определение разладки

## **Географическое масштабирование**

Было необходимо реализовать пайплайн, который способен прогнозировать потребление для зданий из разных регионов России. В связи с этим понадобились некоторые модификации. Например, в базу данных были залиты данные о погодных условиях в 10 различных точках России, взятых равномерно по широте. При прогнозировании данные о погоде для региона брались из максимально близкой широты. Также, для определения длины светового дня была написана функция, которая находит эту длину по широте географического местоположения объекта и дню в году, используя приближенную математическую формулу [6] (точное определение - сложная задача, из-за неправильной формы Земли и географических особенностей нахождения того или иного здания).

# **Тестирование**

Для оценки качества пайплайна мы сравнили качество прогноза с существующими библиотеками для автоматизации прогнозирования временных рядов или автоматического машинного обучения на рядах с потреблением различных ресурсов.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тепловая энергия | Электроэнергия | Горячая вода |
| Pipeline | 1078 | 0.26 | 84.6 |
| Light automl |  |  |  |
| Prophet |  |  | 84.6 |
| Auto arima |  |  |  |
| TBATS |  |  |  |

Таблица 3. Сравнение пайплайна с автоматизированными библиотеками машинного обучения.

Результаты сравнения.

# **Заключение**

Получена автоматизированная библиотека для прогнозирования различных временных рядов, показавшая себя работоспособной на большинстве имеющихся данных. Исходный код, описание проекта и эксперименты выложены в открытый доступ <https://github.com/sonya123454/Automated-Pipeline-for-Predicting-Readings-for-Objects-with-Different-Characteristics>.

Важно отметить, что данная работа открывает много возможных направлений для изучения и улучшения текущего пайплайна. Возможные задачи, которые могут быть поставлены в рамках развития данного решения:

1. Более тщательная работа с созданием признакового пространства. Упор на прогнозирование погоды, и определение смены сезона отопления (сейчас определяется по законам РФ).
2. Проверка иных моделей прогнозирования рядов в том числе, реализованных на языке R. Например, NNetar и LSTM.
3. Работа над удобством интерфейса пайплайна для более легкого внедрения решения на другие проекты.

# **Список источников**

1. <https://pypi.org/project/pyramid-arima/>
2. https://facebook.github.io/prophet/docs/quick\_start.html
3. <https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html> - Библиотека для авто регрессионных моделей
4. Saeed Aghabozorgi, Ali Seyed Shirkhorshidi n , Teh Ying Wah. Time-series clustering – A decade review // [Information Systems](https://www.sciencedirect.com/science/journal/03064379), 2015

<https://wiki.smu.edu.sg/18191isss608g1/img_auth.php/f/fd/Time_Series_Clustering_A_Decade_Review.pd>

1. Joan Serria, Josep Lluis ArcosA. Competitive Measure to Assess the Similarity Between Two Time Series// 2012
2. [Yacine Ben Baccar](https://www.researchgate.net/profile/Yacine-Ben-Baccar). Comparative Study on Time Series Forecasting Models // 2019
3. Yue Teng, Pengjian Shang Transfer entropy coefficient: Quantifying level of information flow between financial time series // Physica A, 2016.
4. Forsythe W. C. et al. A model comparison for daylength as a function of latitude and day of year //Ecological Modelling. – 1995. – Т. 80. – №. 1. – С. 87-95. (моделирование длины светового дня)
5. Ахтямов П. И., Нейчев Р. Г., Стрижов В. В. Отбор мультикоррелирующих признаков в задаче векторной авторегрессии.
6. Nogueira S., Sechidis K., Brown G. On the stability of feature selection algorithms //J. Mach. Learn. Res. – 2017. – Т. 18. – №. 1. – С. 6345-6398.
7. <https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D1%80%D1%8F%D0%B4%D0%BE%D0%B2>
8. <https://pypi.org/project/pyramid-arima/>
9. Шерстобитова А. О. и др. Последовательное обнаружение разладки во временных рядах. – 2019.

# **Приложения**

## **Приложение A**

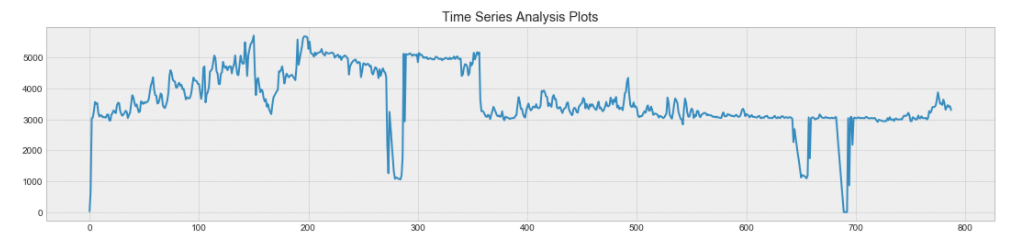
**Кластер 1**

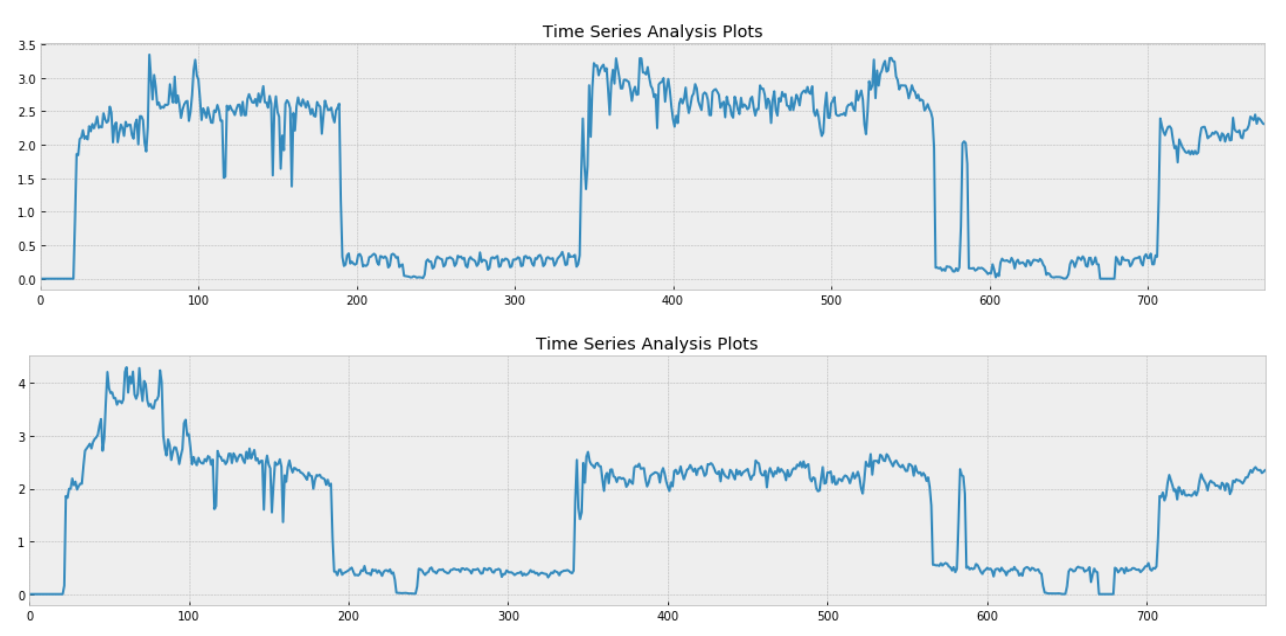


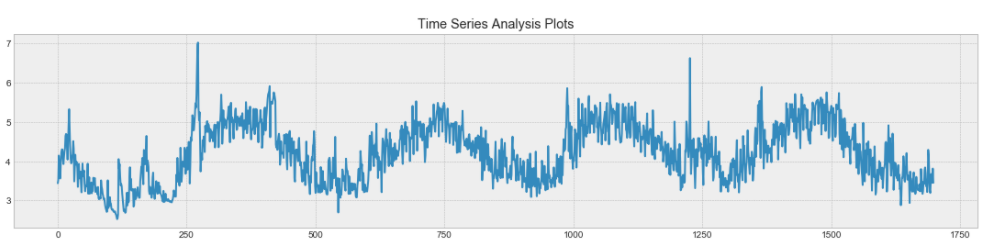
**Кластер 2**

Изображение выглядит как текст, доска

Автоматически созданное описание

Кластер3

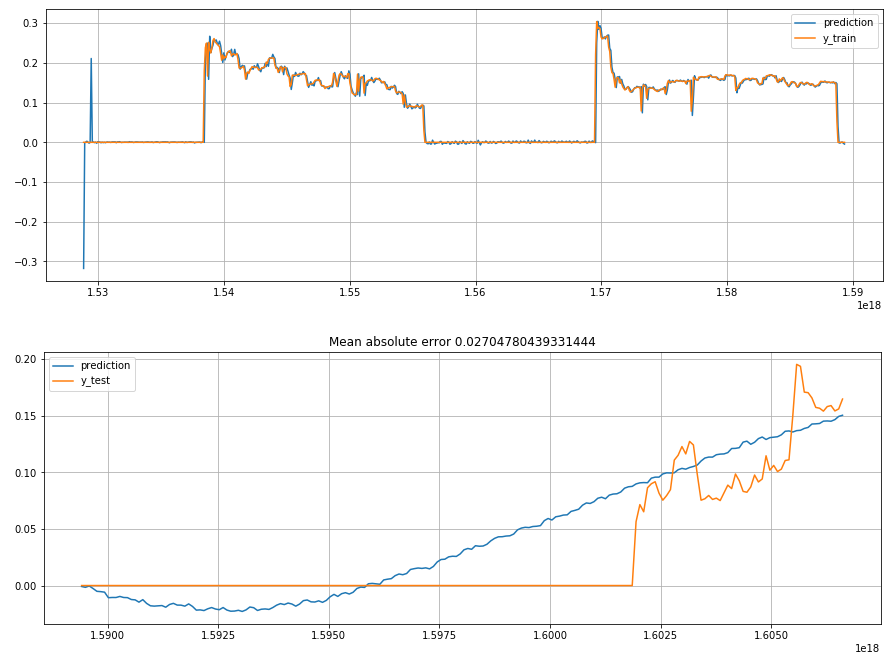
Кластер 4

Кластер 5

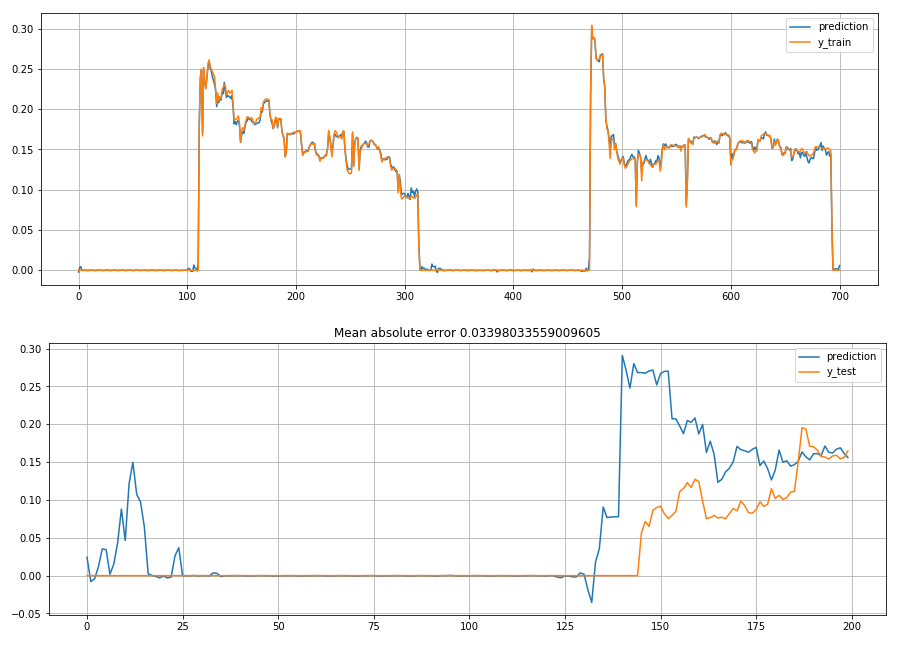
## **Приложение Б**

Тепловая энергия

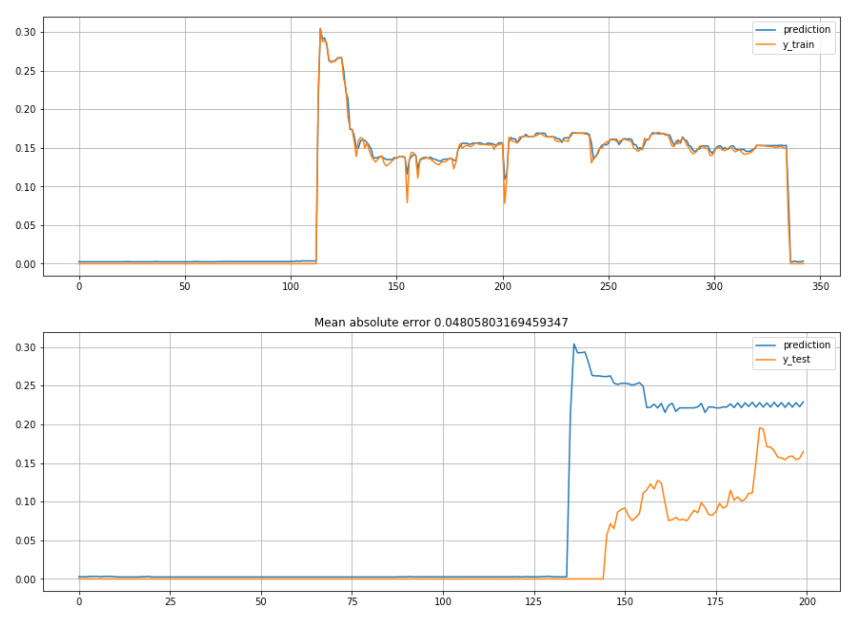
SARIMAX



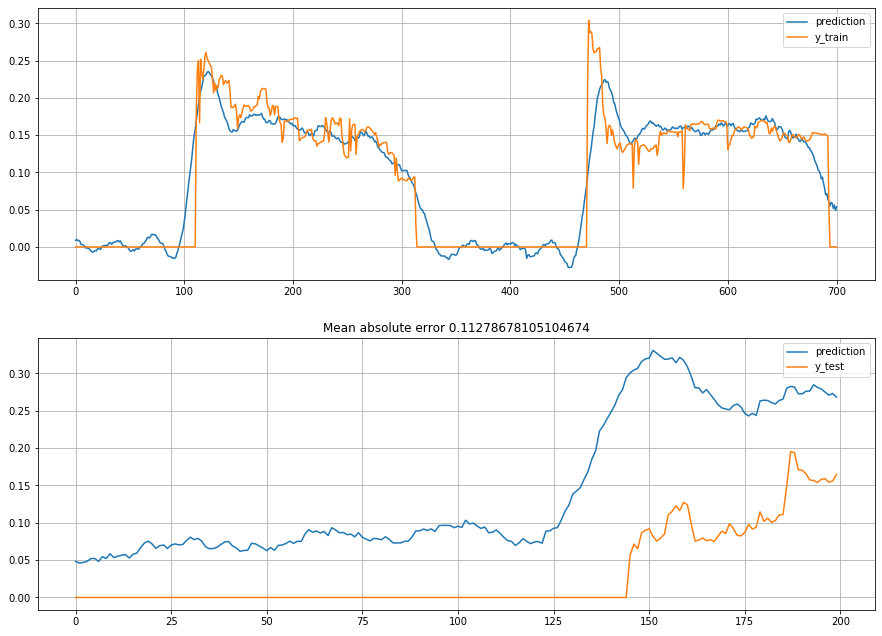
XGBoost без лагов



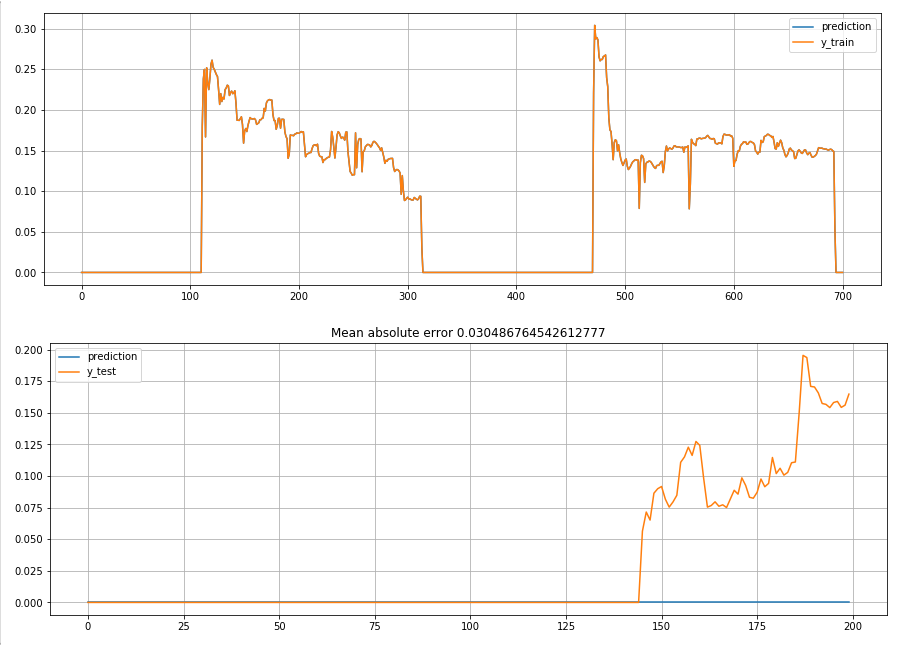
XGBoost с добавлением лагов

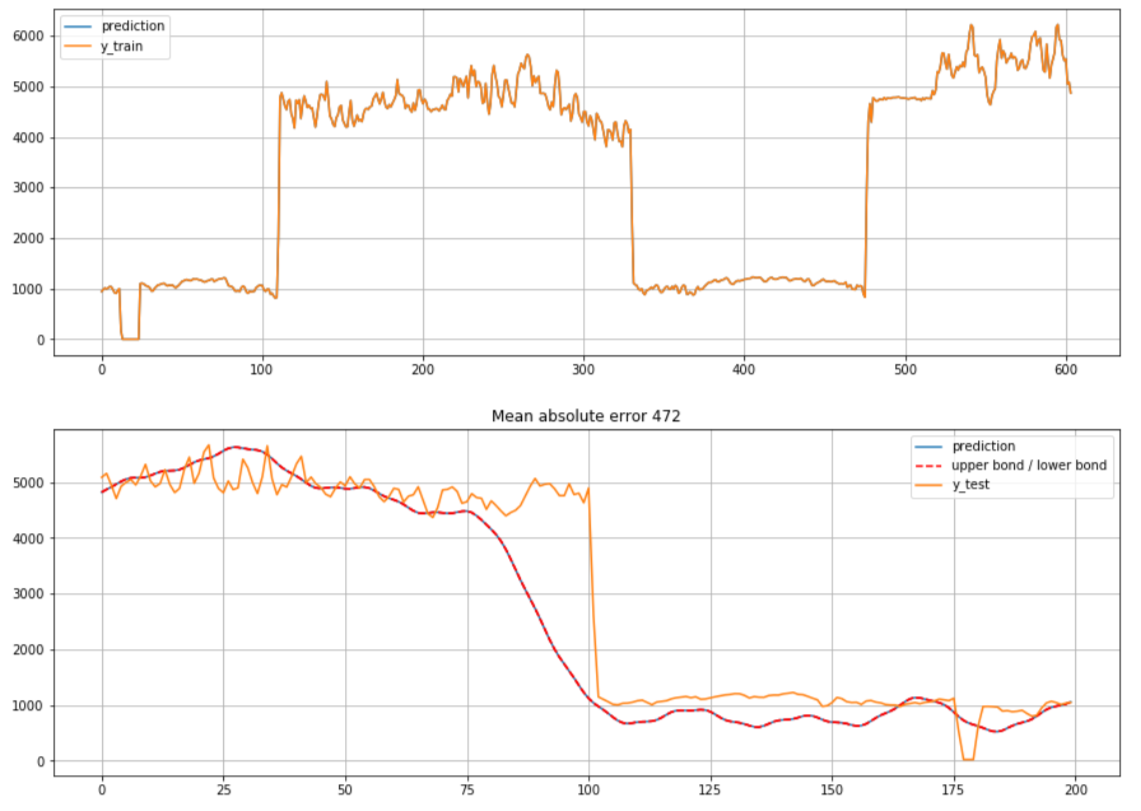


Prophet

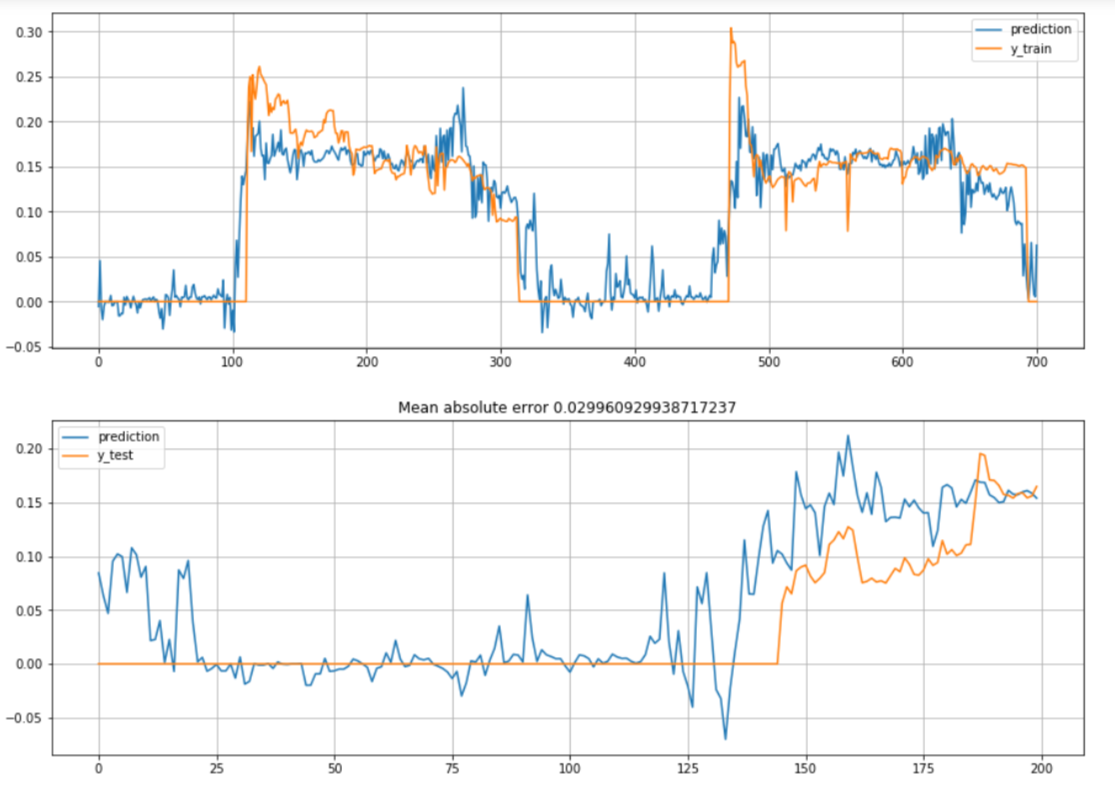


TBATS





GAM



Электрическая энергия

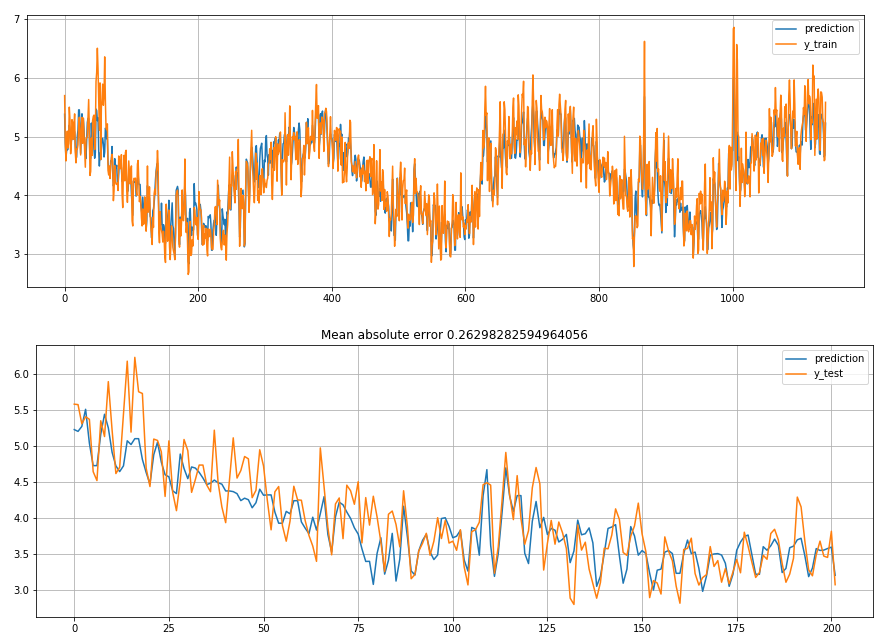
SARIMAX



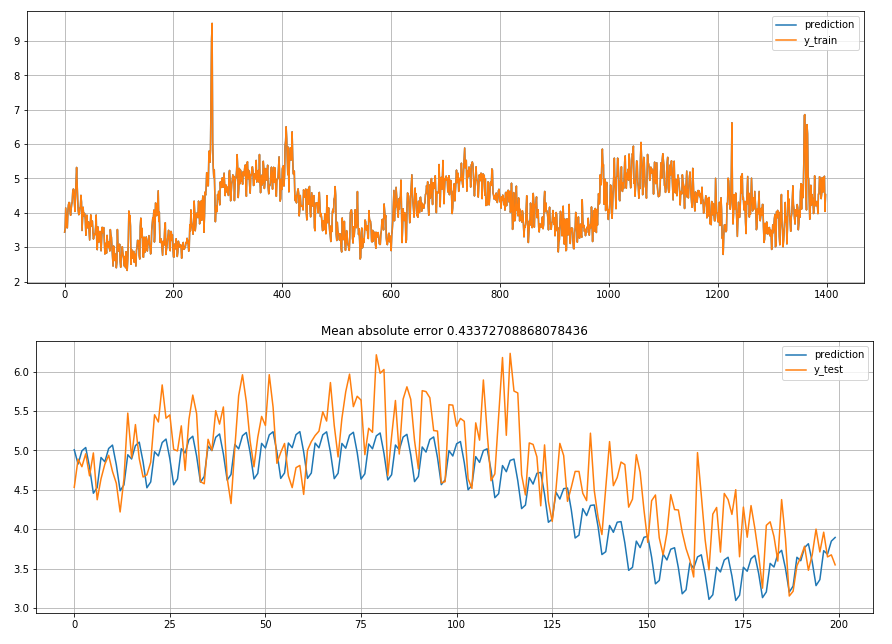
Prophet



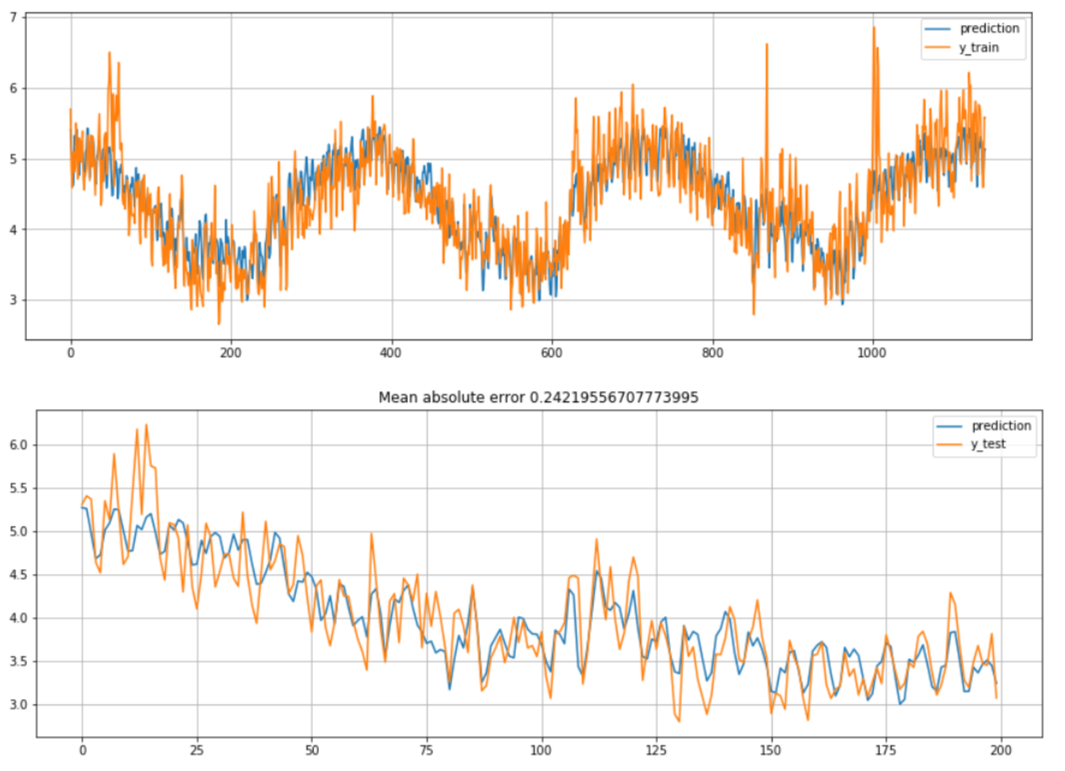
XGBoost



TBATS



GAM

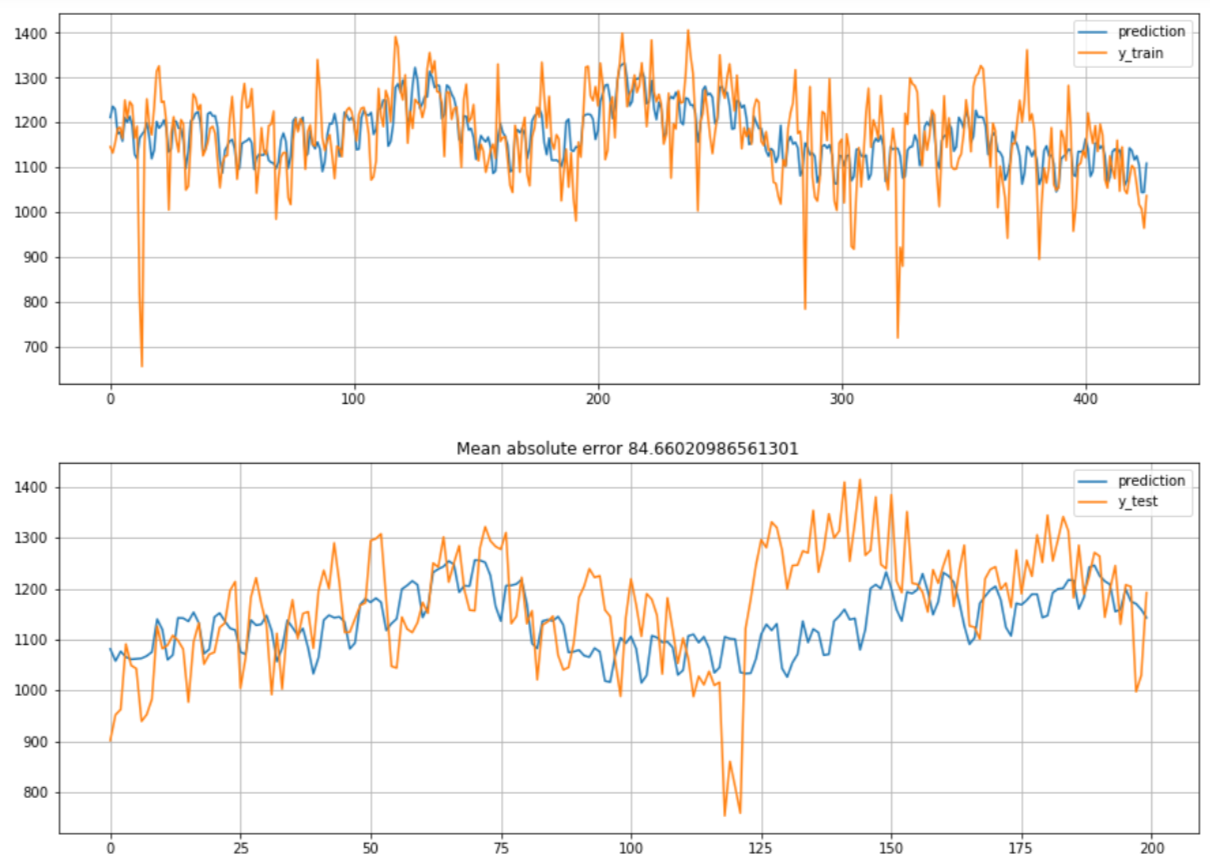


Горячая вода

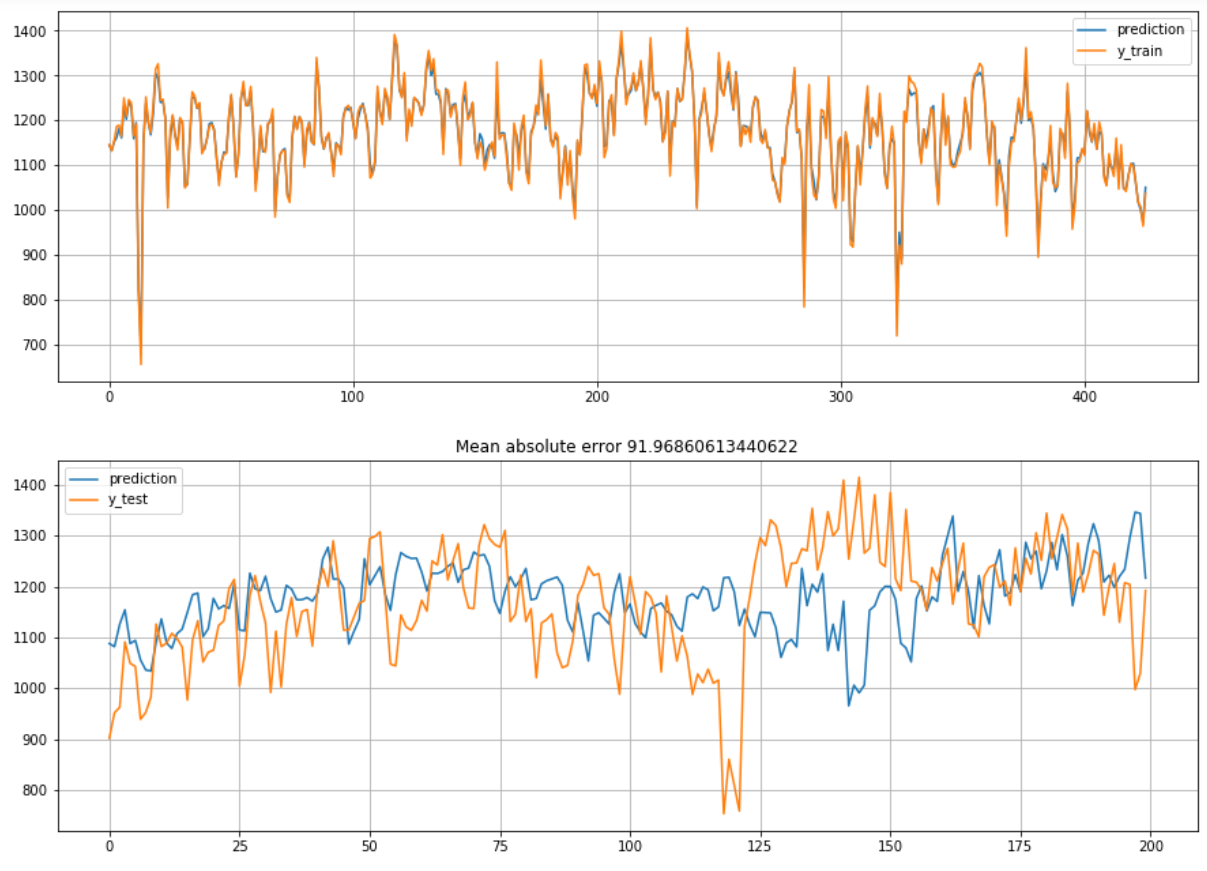
SARIMAX



Prophet



XGBoost



TBATS



GAM

