****

ОТЧЕТ ПО ЗАДАНИЮ №5

Прогнозирование потока событий

Автоматический поиск аномалий во временных рядах с использованием алгоритма ARIMA

*Работу выполнила Акулова Екатерина для ГК Innostage*

*(задача для стажеров DS, BD)*

27 декабря 2022

**Содержание**

ВВЕДЕНИЕ 3

1. Подготовка окружения 4

2. Алгоритм ARIMA 5

3. Блок - схема 6

4. Реализация 7

4.1 Анализ временного ряда 8

4.2 Подготовка данных 17

4.3 Обучение модели ARIMA и прогнозирование 19

4.4 Поиск аномалий 22

4.5 Результат 24

ЗАКЛЮЧЕНИЕ 29

ВВЕДЕНИЕ

Аномалии - это образцы данных, которые не соответствуют четко определенному понятию нормального поведения. Важность выявления аномалий обусловлена тем фактом, что аномалии в данных приводят к значительной и действенной информации в самых разных областях применения. Например, аномальная схема трафика в компьютерной сети может означать, что компьютер взломан и отправляет конфиденциальные данные неавторизованному адресату. Aномалии в данных транзакций по кредитной карте, которые могут указывать на кредитную карту или кражу личных данных. Обнаружение аномалий было изучено несколькими исследовательскими сообществами для решения проблем в разных областях применения. Во многих областях, таких как безопасность, обнаружение мошенничества, медицинское обслуживание и т.д.

Временной ряд - это собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Другими словами, упорядоченная во времени последовательность значений какого-либо датчика. Такие данные легко собрать, и они имеют высокую значимость в определении состояния системы, так как могут быть получены в реальном времени. Таким образом остро встает задача анализа данного типа данных.

Задача направлена на прогнозирование потока событий и обнаружение аномалий c использованием языка python и любых библиотек. В результате должна получиться таблица интервалов и значений, на которых произошли отклонения от прогнозных данных. Количество значений не должно превышать 1% от всего объема данных.

В данной работе рассматривается алгоритм ARIMA (семейство алгоритмов ARMA/ARIMA/ARMA/SARMA). ARIMA - авторегрессионная интегрированная модель скользящего среднего, представляет собой алгоритм прогнозирования, основанный на идее, что информация о прошлых значениях временного ряда может использоваться для прогнозирования будущих значений.

1. Подготовка окружения

Вся работа выполнялась на macOS Monterey 12*.*6

Для выполнения задания необходимо установить:

1. Jupyter Notebook. *Использовался для анализа данных и визуализации*
2. Pycharm *Использовался для написания программного кода*

Библиотеки для работы:

* statsmodels
* pmdarima
* sklearn
* pandas
* numpy
* plotly
* itertools

Данные были приложены к заданию.

2. Алгоритм ARIMA

ARIMA - (A)авто- (R)егрессионная- (I)интегрированная модель (M)скользящего(A)среднего. Вот разбивка того, что означает каждый из этих терминов:

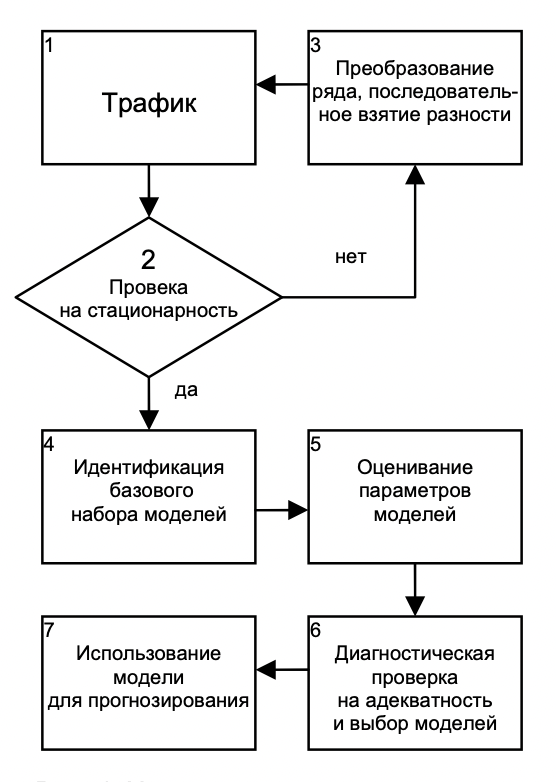
Авторегрессионная: это, по сути, означает, что ARIMA просматривает исторические данные, чтобы помочь предсказать следующую точку данных (насколько далеко назад мы ищем, чтобы помочь предсказать следующую точку данных). В ARIMA термин «AR» также обозначается как «p» и может быть установлен вручную, просмотрев графики PACF.

Интегрированная: этот термин в ARIMA относится к разности значений в наборе данных временного ряда. Это помогает преобразовать ряд в стационарный ряд. Когда временной ряд является стационарным, среднее значение остается примерно одинаковым. Нет тренда вверх или вниз с течением времени. В ARIMA «i» также упоминается как «d». Тест ADF может помочь определить, является ли ряд стационарным.

Скользящее среднее: компонент скользящего среднего в ARIMA по сути является окном. Этот компонент просматривает предыдущие значения, чтобы помочь оценить тенденцию ряда. В ARIMA этот термин обозначается как «q», и его можно установить вручную, посмотрев на графики ACF.

Было бы удобно, если бы была функция, которая автоматически пыталась оценить эти параметры для нас, чтобы мы могли просто, быстро и легко получить прогноз? Такая функция есть! Аналогичная auto.arima для R функция есть и в Python благодаря библиотеке pmdarima. Мы также будем использовать эту библиотеку и функцию auto\_arima, чтобы упростить задачу.

3. Блок - схема

Блок-схема программы Блок-схема алгоритма

4. Реализация

Программа разделена следующим образом:

1. Директория dataсодержитфайлы - исходные данные, evaluate\_arima с оценкой разных параметров для модели arima, сохраненные модели, таблица-результат выявленных аномалий для auto arima и arima
2. Директория notebooks содержит ноутбук с первичным анализом данных и графиками.
3. Директория detect\_anomalies:

detect.py для обнаружения аномалий.

1. Inference

evaluate\_arima.py для оценки параметров модели.

validate\_model.py для валидации модели.

visualize.py для построения итоговых графиков.

5. Директория training

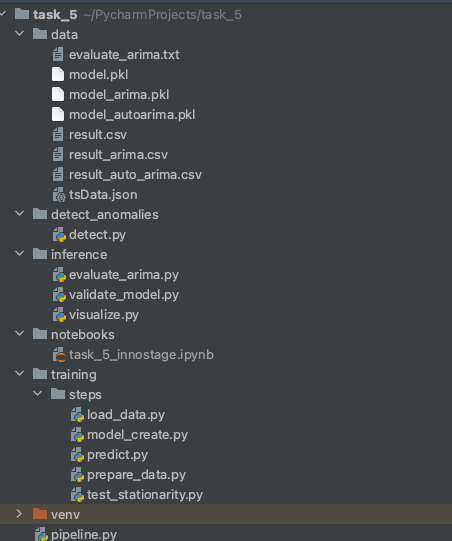
load\_data.py для загрузки исходных данных.

prepare\_data.py для подготовки данных.

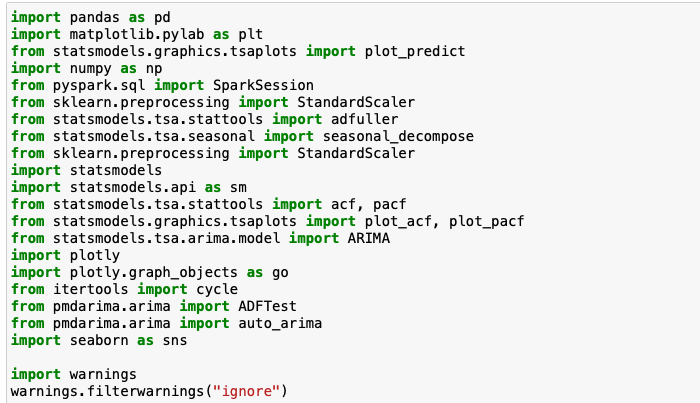
test\_stationarity.py содержит тест на стационарность данных.

model\_create.py для создания моделей.

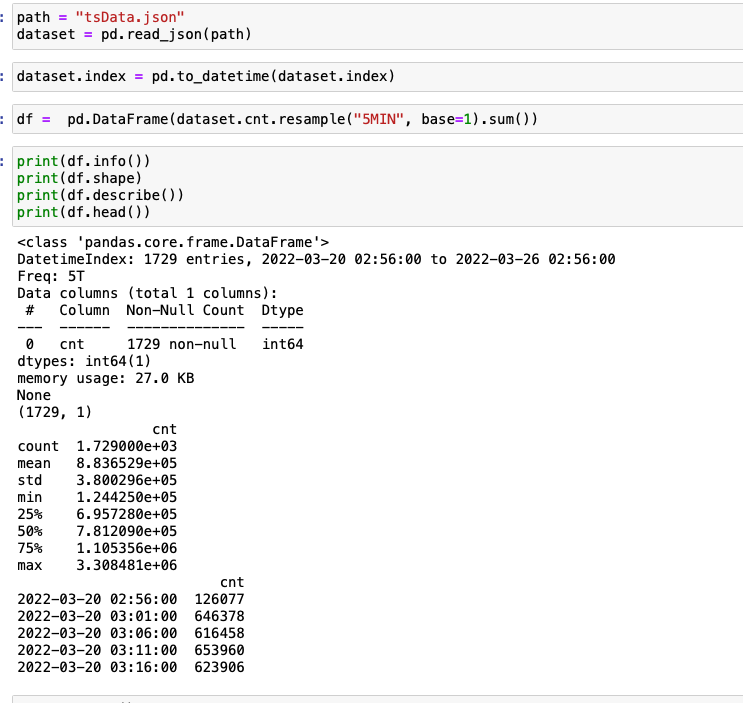
predict.py для получения предсказаний.

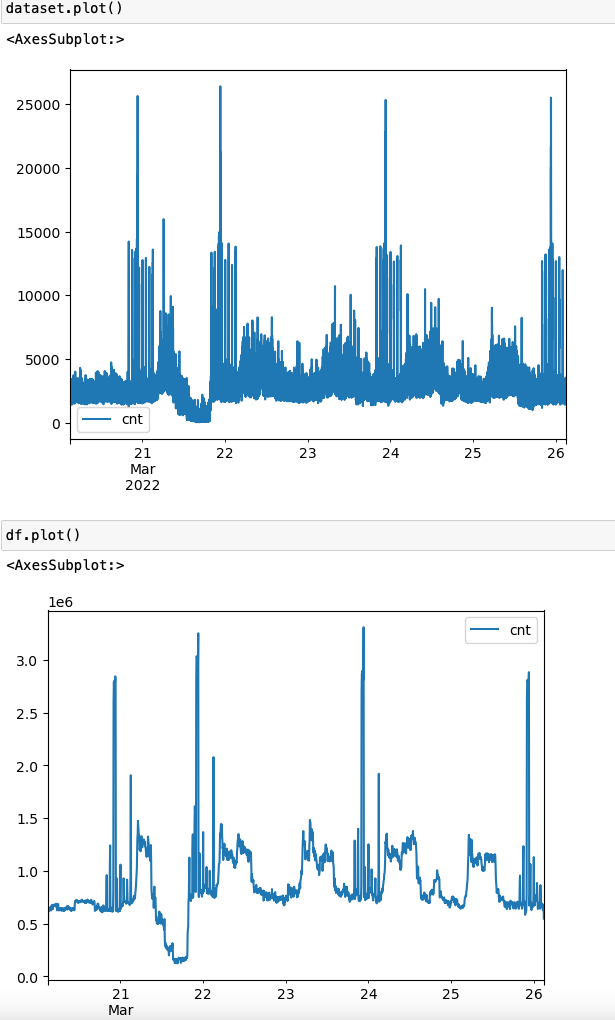
6. pipeline.py - cодержит три функции пайплайна, которые используют данные с разными преобразованиями и параметрами модели ARIMA.

4.1 Анализ временного ряда

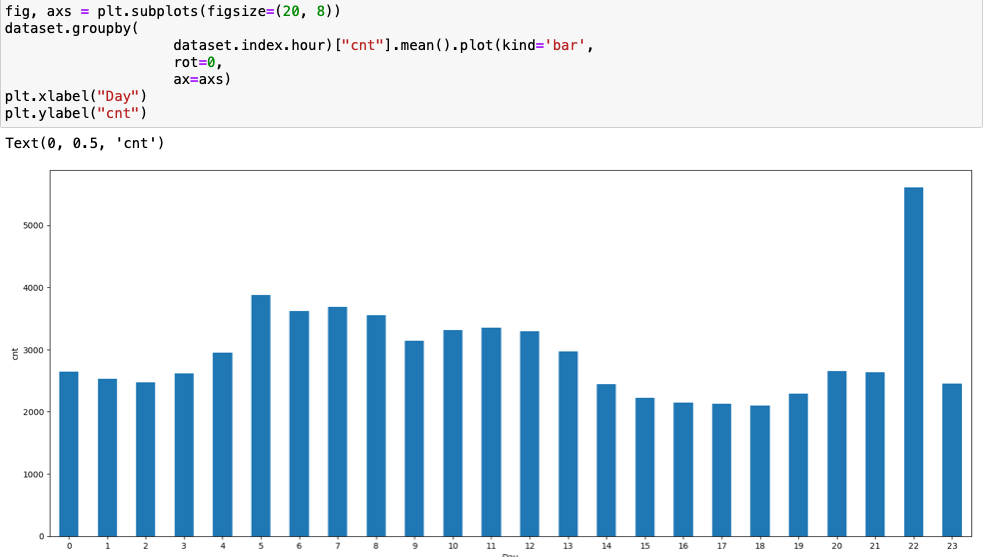
Для начала загрузим нужные библиотеки.

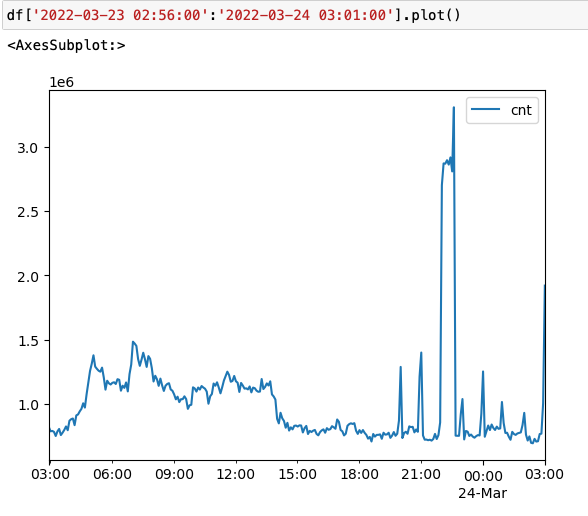
Загрузим данные, переведем индекс датасета в формат datetime, посмотрим на данные и установим интервал временного ряда в 5 минут, так как данных много и это повлияет на скорость обучения модели.Изображение



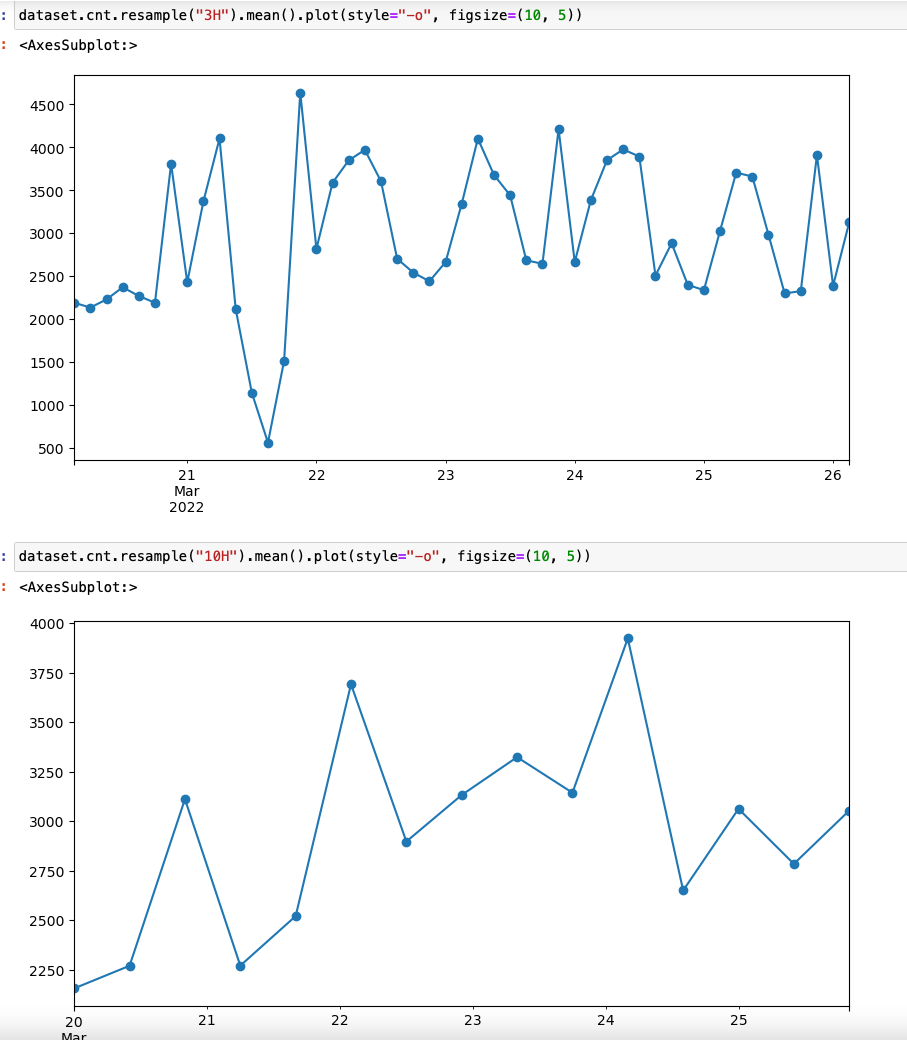
Посмотрим на графики с интервалами в 1 секунду и в 5 минут.

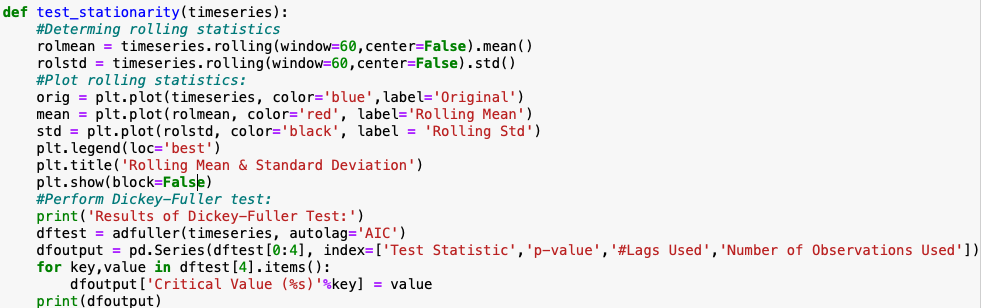
Тренд и сезонность в данных с интервалом в 5 минут сохранились. Значит такой интервал подходит для повторной выборки.

Построим график среднего значения cnt для каждого часа в сутках.

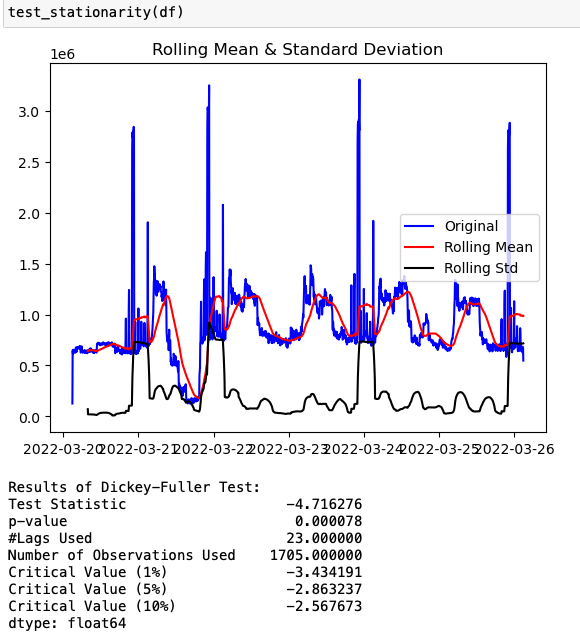
Выберем любой день и построим график.

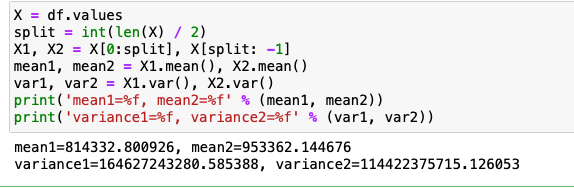
Действительно, можно отметить что самые высокие значения наблюдаются в 22 часа и значения выше среднего в ~ 5 - 12 часов.

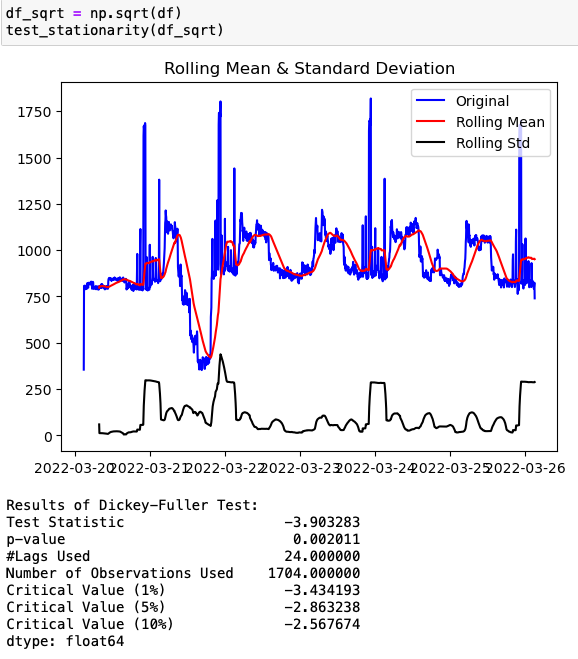
Построим еще пару графиков с другими временными интервалами и пойдем дальше.

Перейдем к тесту на стационарность временного ряда. Данные для обучения модели ARIMA должны быть стационарны, то есть статистические свойства процесса, генерирующего временные ряды, не должны изменяться со временем. Если временной ряд имеет определенное поведение на временном интервале, то существует высокая вероятность того, что на другом интервале оно будет вести себя так же, при условии, что временной ряд является стационарным. Это помогает точно прогнозировать.

В данной функции мы определяем окно для скользящего среднего и скользящего стандартного отклонения и рисуем график. Далее используем тест Дики-Фуллера для проверки на стационарность.

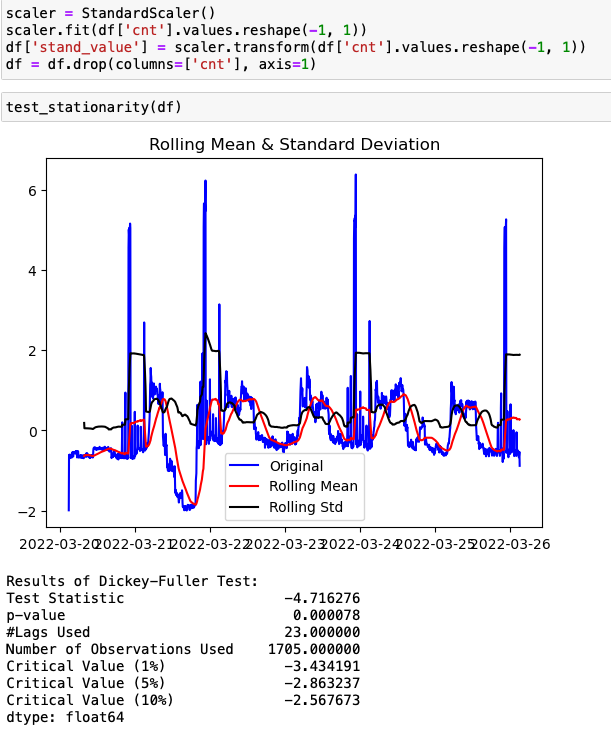
Посмотрим на результат выполнения функции для данных с интервалом в 5 минут.

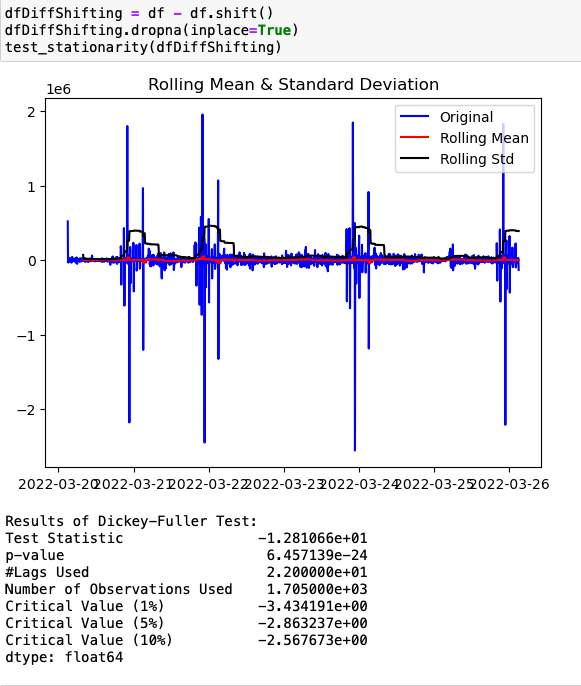
Теперь разделим данные наполовину и посмотрим, насколько различается медиана и дисперсия двух массивов.

Нулевая гипотеза о нестационарности временного ряда отвергается. Но тем не менее полученный график выглядит не идеально, можно попробовать преобразовать данные.

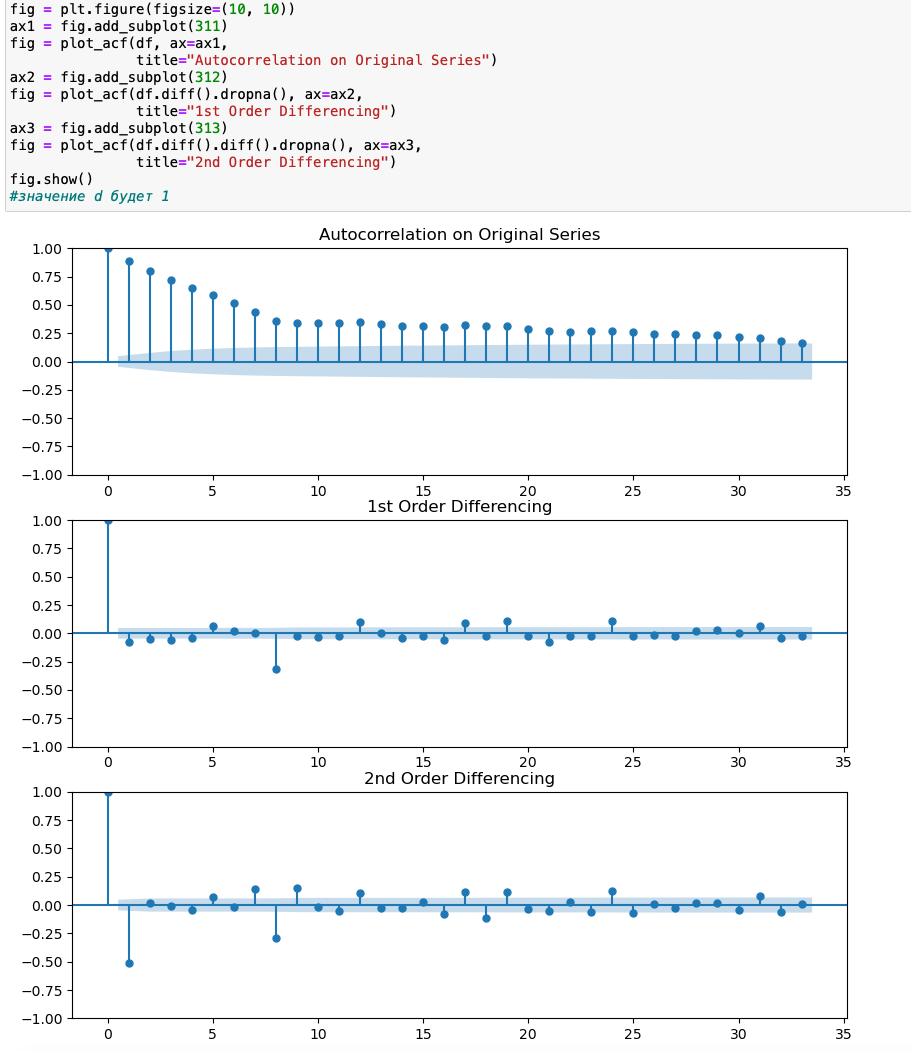
Набор данных мы преобразовали в новый набор данных, где вместо предыдущих значений - их положительный квадратный корень. Результат теста на стационарность положителен.

Проведем стандартизацию данных и посмотрим на результат.



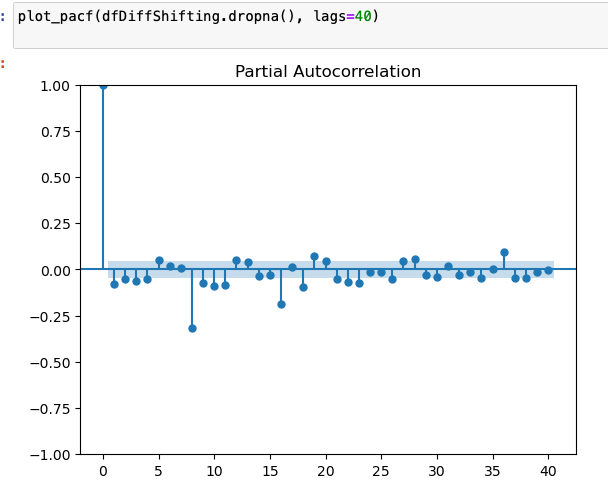
Данные стационарны и их можно использовать для обучения модели. А что если вычислить дискретную разность и сдвинуть данные?

Тут мы получили ровную линию скользящего среднего и крайне низкое значение p-value путем сдвига данных. А еще мы получили значение d = 1. Но можно проверить это другим способом.

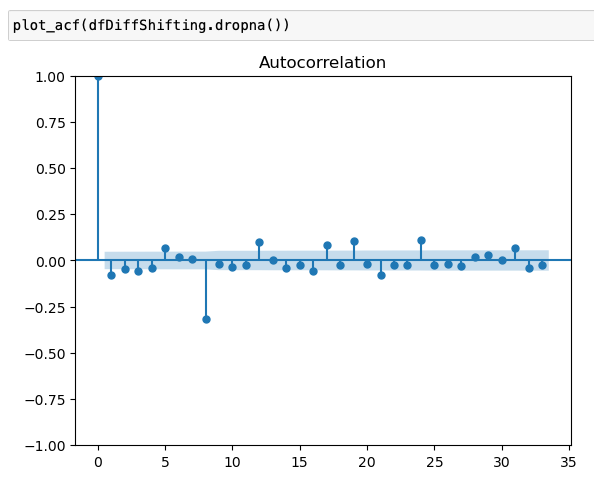
Итак, нам нужно определить значение параметров для ARIMA - p, d и q (p - порядок компоненты [AR](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C), d - порядок интегрированного ряда, q - порядок компоненты [MA](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D1%81%D0%BA%D0%BE%D0%BB%D1%8C%D0%B7%D1%8F%D1%89%D0%B5%D0%B3%D0%BE_%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B5%D0%B3%D0%BE)). Это можно сделать через построения графиков автокорреляции и частичной автокорреляции. Нам нужно проверить, при каком значении по оси x линия графика падает до 0 по оси y в первый раз.

Как видим по графикам, d = 1, так как только первый лаг гораздо выше уровня значимости.

Зная, что мы должны вычислить 1-ю дискретную разность, мы продолжаем выяснять порядок AR - p. AR относится к прошлым значениям, используемым для прогнозирования следующего значения. Мы получаем p путем подсчета количества лагов выше уровня значимости в функции частичной автокорреляции:

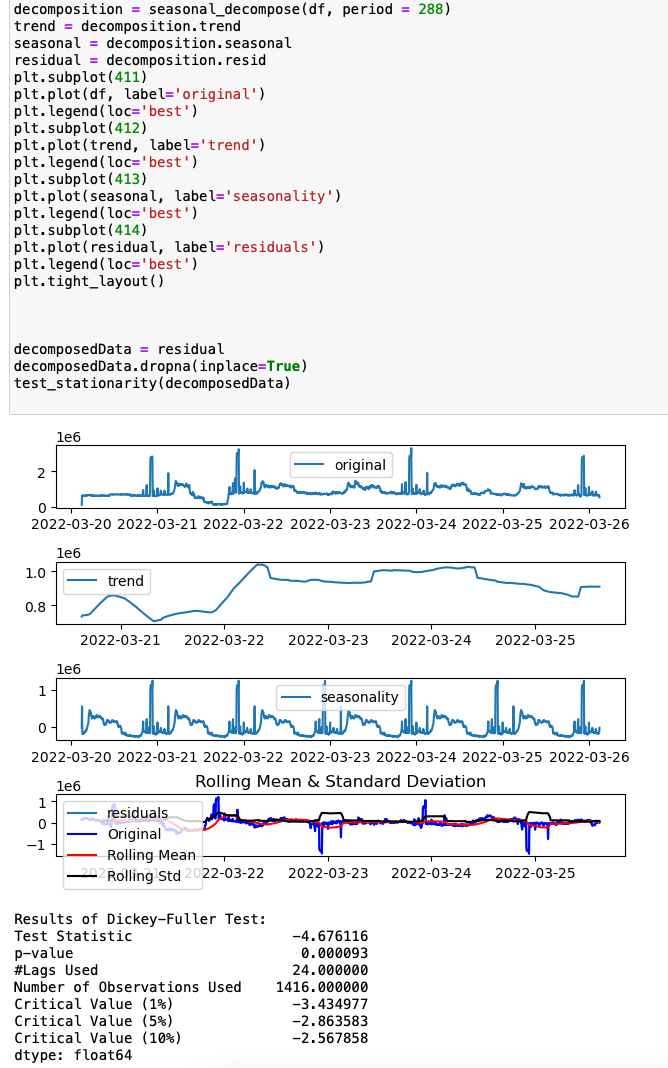


Выше только один лаг, таким образом, р = 1.

MA (скользящее среднее) — используется для определения количества прошлых ошибок прогноза, используемых для прогнозирования будущих значений. Функция автокорреляции поможет нам определять порядок членов МА - q, т. к. по ее коррелограмме можно определить количество автокорреляционных коэффициентов сильно отличных от 0 в модели MA.

Можно заметить больше всего отличен один лаг. Но возможно, p, d, q могут принимать другие значения. Об этом будет написано далее.

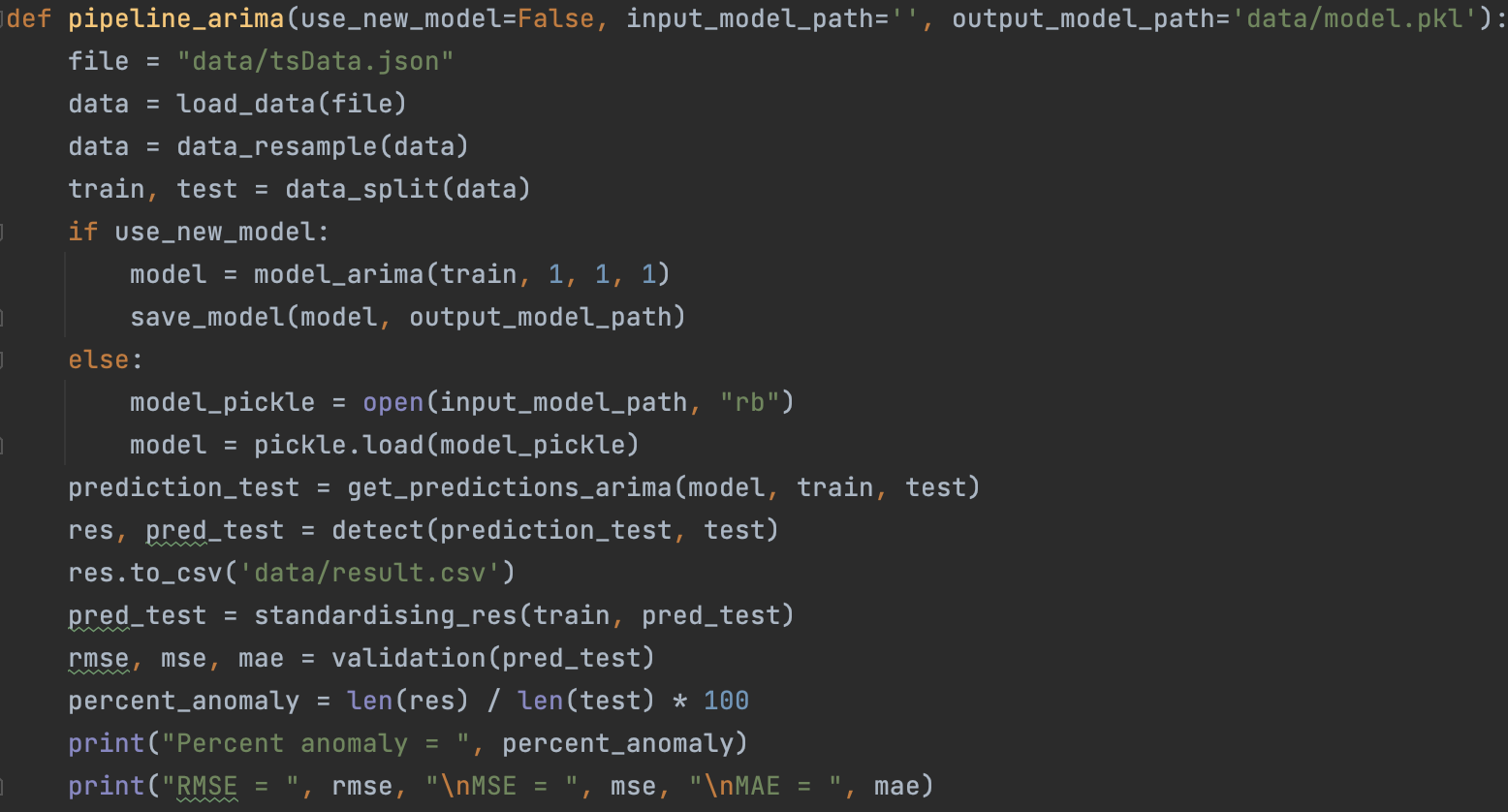
Напоследок посмотрим на компоненты временного ряда.

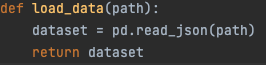


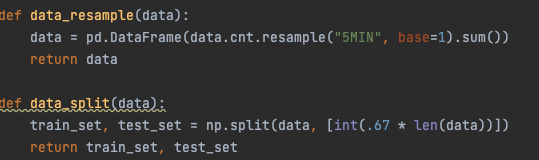
Последний график это неравномерности которые есть во временном ряду, они не имеют порядок, размер и не могут помочь в прогнозировании. Судя по тесту, этот шум стационарен.

Теперь мы закончили с анализом временного ряда. Можно переходить к следующему пункту.

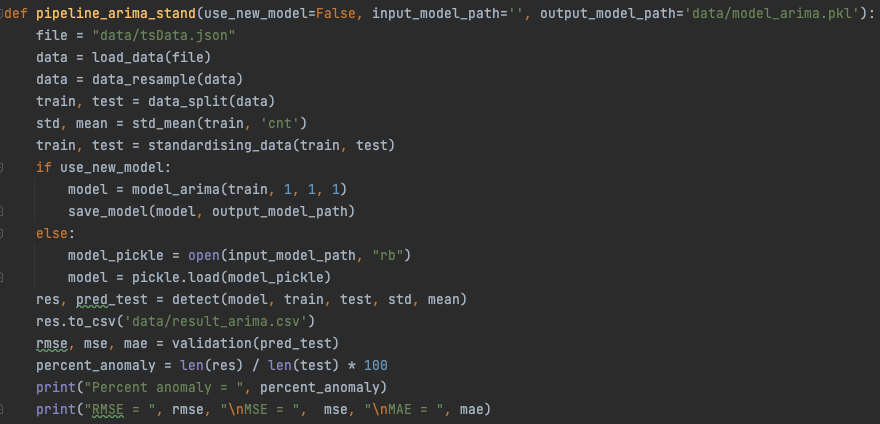
4.2 Подготовка данных

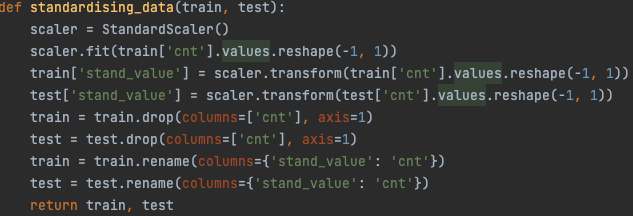
Сначала нам нужно ввести данные, создать повторную выборку с интервалом в 5 минут, разделить данные на тренировочную и вестовую выборки, и преобразовать данные. В одной из трех функций данные не преобразовываются, начнем с нее.

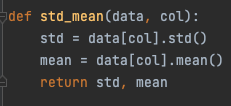
В данном участке кода подготовка данных происходит в первых четырех строчках. Заглянем в каждую из функций.

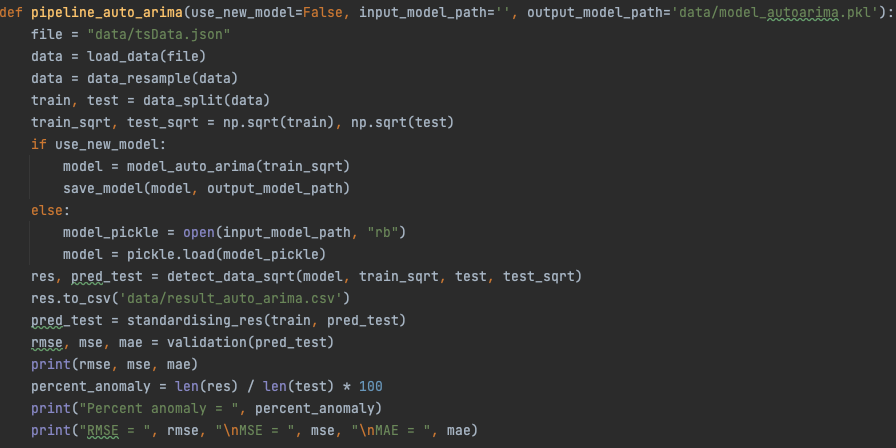


Во втором случае добавлена функция для стандартизации данных и функция для вычисления медианы и стандартного отклонения. Стандартизация изменяет форму распределения данных (приводится к нормальному распределению). Медиану и стандартное отклонение нужно вычислить, чтобы затем привести данные к их изначальному виду.

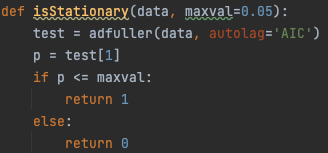




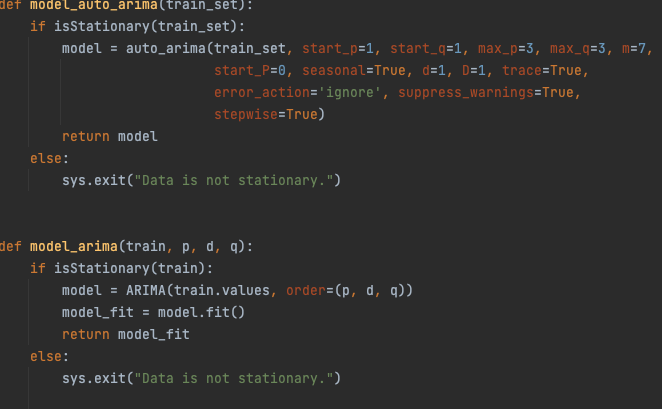


В третьем случае была добавлена строчка с преобразованием тренировочных и тестовых данных через нахождение квадратного корня их значений.

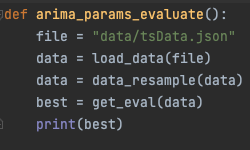
4.3 Обучение модели ARIMA и прогнозирование

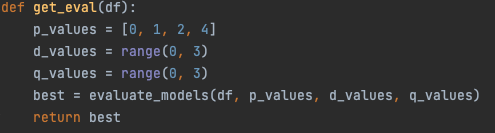
Перед тем как создать и обучить модель, нужно проверить данные на стационарность.

Если выбран параметр «AIC» или «BIC», то количество лагов выбирается таким образом, чтобы минимизировать соответствующий информационный критерий.

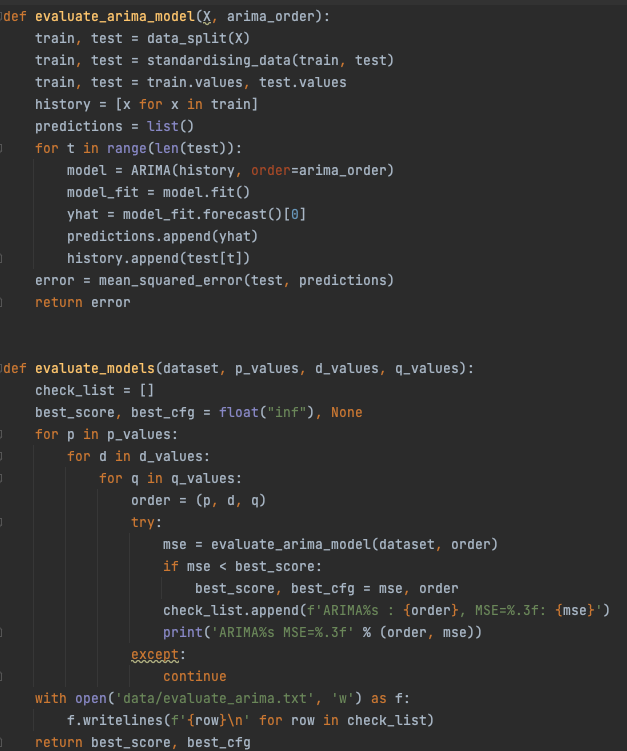
Теперь перейдем к обучению модели.

В данном участке кода создаются две модели ARIMA с автоматическим подбором параметров и без него.

Рассмотрим функцию, которая поможет оценить, насколько подобранные параметры подходят для модели без автоматического подбора через вычисление средней квадратичной ошибки моделей с различными параметрами путем их перебора. Ниже описан способ ее реализации.



Здесь мы определяем диапазон значений для каждого из трех параметров. Таким образом, функция будет перебирать каждую возможную комбинацию.

Посмотрим, как работает подбор параметров. Все параметры модели и полученные ошибки записываются в файл, чтобы можно было запустить данный код отдельно и затем оценить параметры перед обучением, прогнозом и выявлением аномалий.

Здесь создаются различные комбинации из переданных значений и каждая из них оценивается. Раскроем алгоритм:

1. Разделить набор данных на обучающий и тестовый наборы

2. Пройти временные шаги в тестовом наборе данных

а. Обучить модель ARIMA

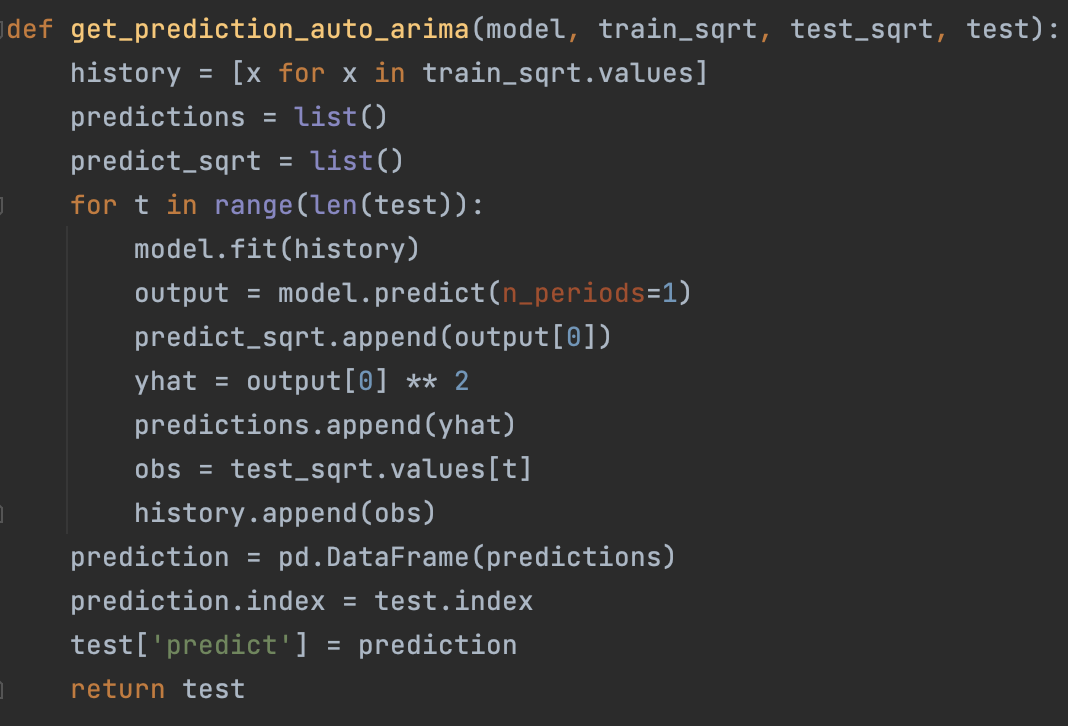
б. Сделать одношаговый прогноз

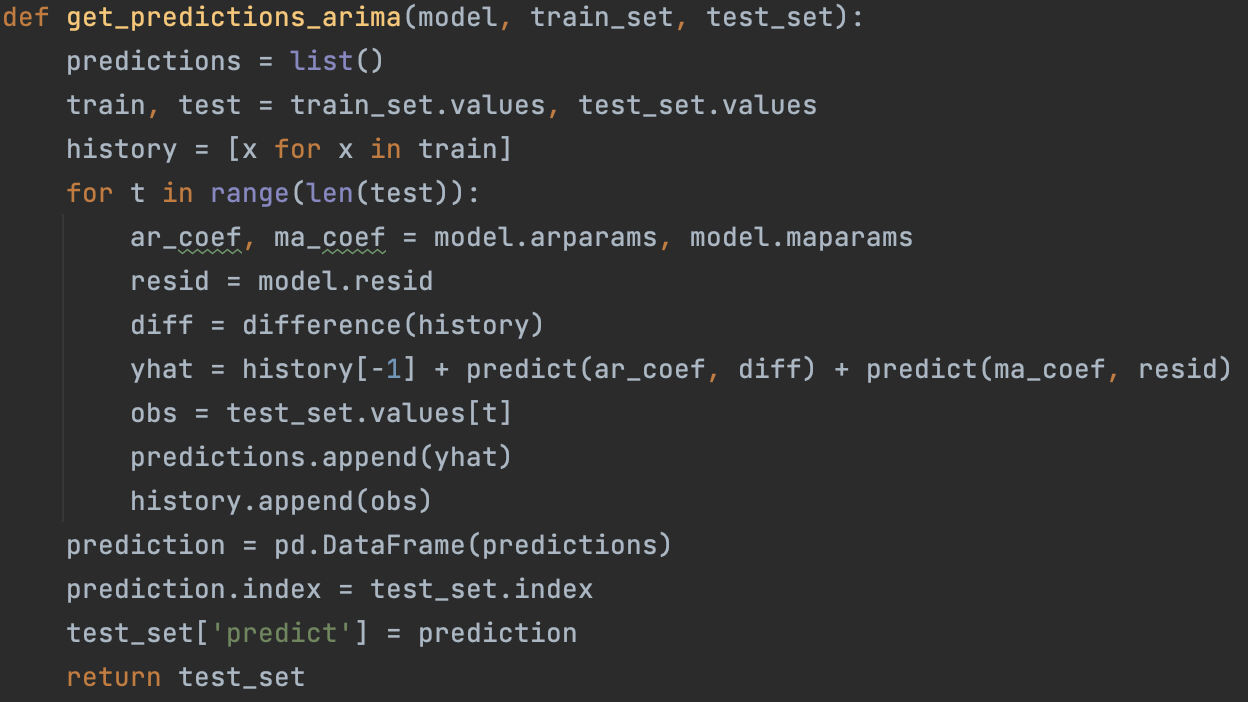
в. Cохранить прогноз; получить и сохранить фактическое наблюдение.

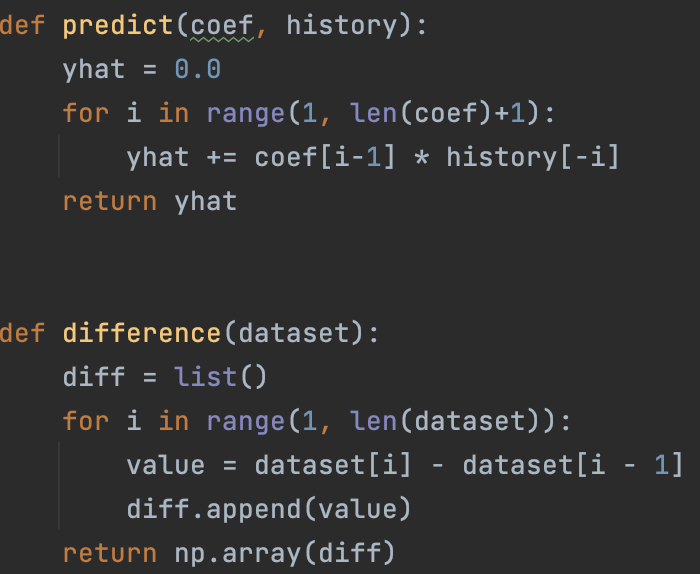
3. Вычислить ошибку для прогнозов по сравнению с ожидаемыми значениями.

Наименьшее полученное значение MSE — лучший набор параметров.

Вывод будет показан в разделе с результатами.

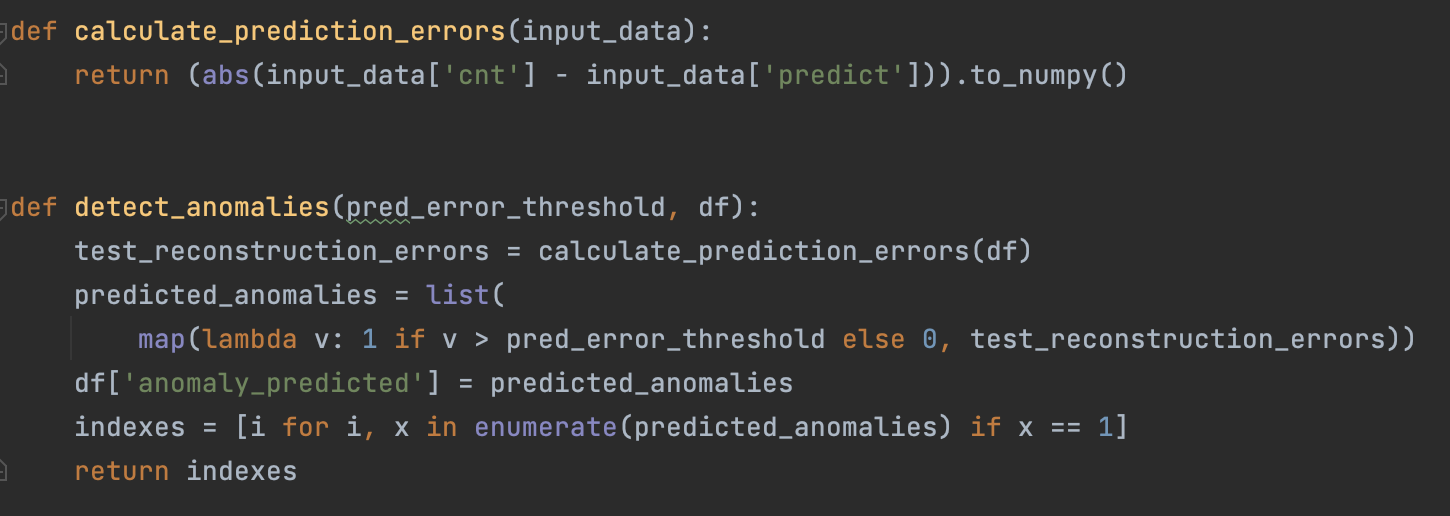
Теперь перейдем к получению предсказаний. Напишем две функции для двух моделей, одна из которых (1) будет использовать model.predict(), а другая (2) работать по похожему принципу, но используя написанную от руки функцию предсказания. В нее мы будем передавать коэффициенты AR и MA и таблицу разностей каждого элемента в тестовой выборке с его предыдущим элементом. В результате обучения мы должны получить новый столбец в датафрейме тестовых данных, который будет содержать предсказанное значение для каждой строки.



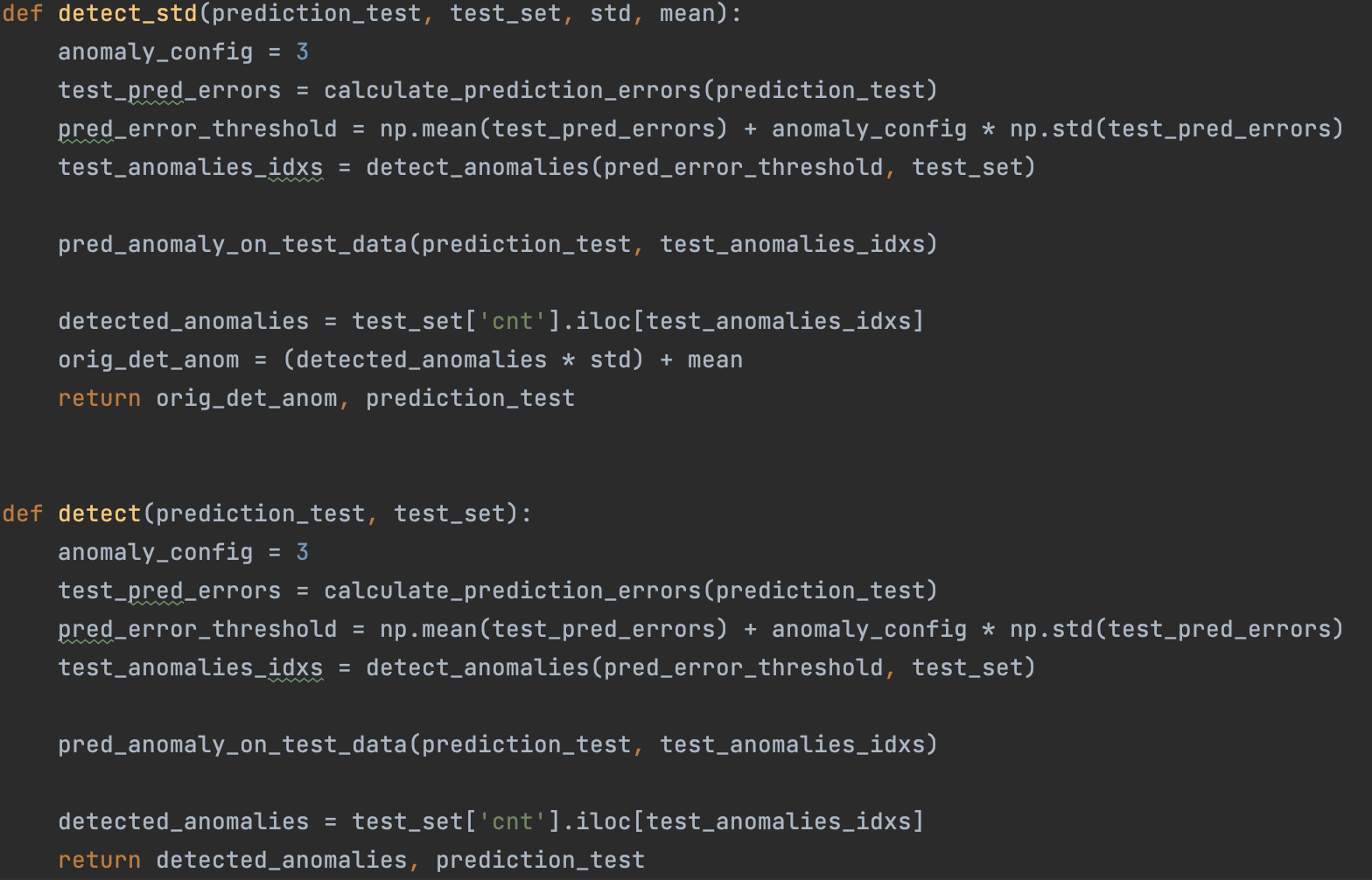


В первом случае мы имели преобразованные значения в виде квадратного корня значений из оригинальной выборки. Мы возводим их во вторую степень. Во втором случае данные не меняются.

4.4 Поиск аномалий

Рассмотрим функцию для обнаружения аномалий на основе ошибок прогнозирования и определенного порогового значения.

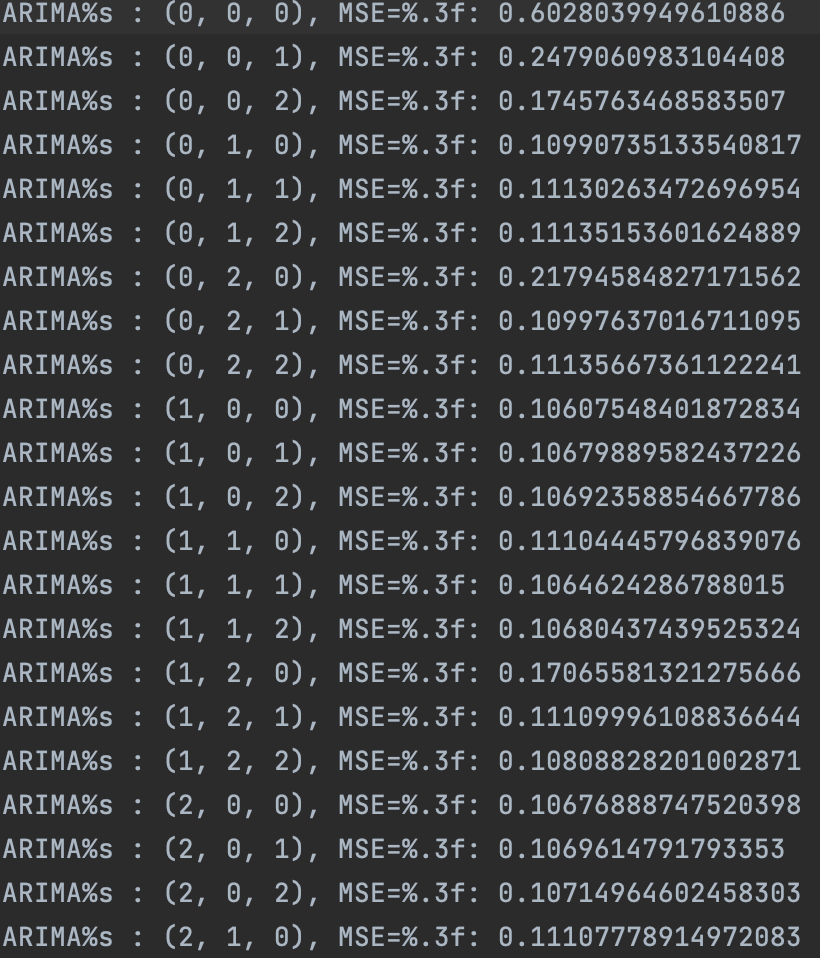
Сначала мы вычисляем ошибку. Делаем мы это путем вычисления абсолютного значения числа, которое представляет из себя разность значений cnt с предсказанными значениями. Затем мы обозначаем предсказанные значения как аномалии либо не аномалии в соответствии с пороговым значением. Пороговое значение представляет из себя число стандартных отклонений ошибки прогнозирования, которое отличается от средней ошибки прогнозирования, чтобы классифицировать ее как аномалию. Чем больше значение параметра конфигурации тем меньше чувствительность обнаружения аномалий. Функция возвращает индексы выявленных аномалий.

Теперь переходим к моменту, где мы должны получить итоговые графики и таблицу с выявленными аномалиями.

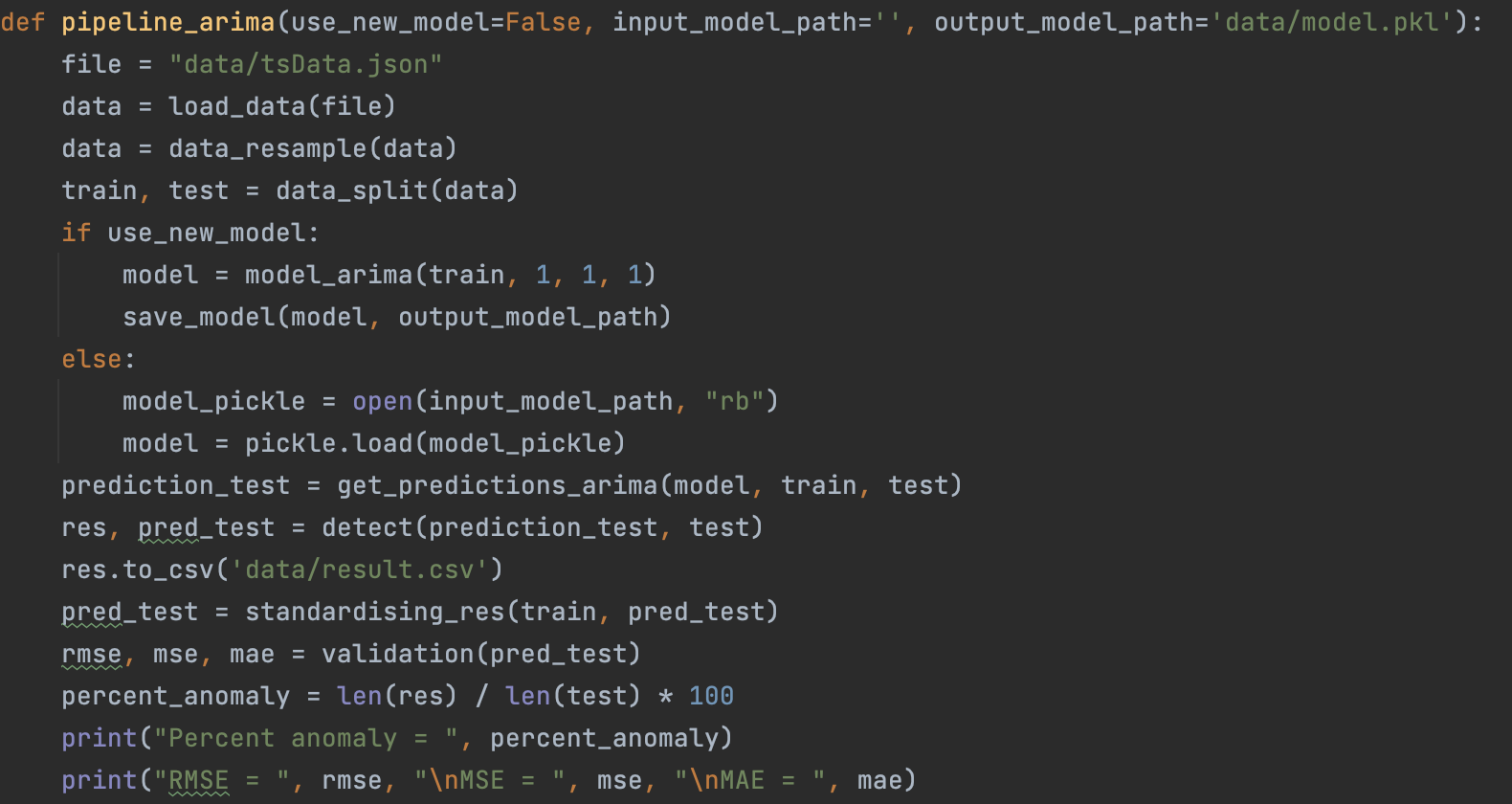
Здесь две разных функции так как на вход они принимают таблицы со стандартизированными данными, которые нужно обратно преобразовать, чтобы функция вернула таблицу с выявленными аномалиями, как положено, либо с данными, которые не нужно обратно преобразовывать. Все шаги были описаны выше, поэтому посмотрим только на часть с визуализацией.

В итоге должен получиться scatter plot, c точками изначальных данных синего цвета, предсказанных - желтого, а красными будут обозначены аномалии. Давайте посмотрим на полученные результаты.

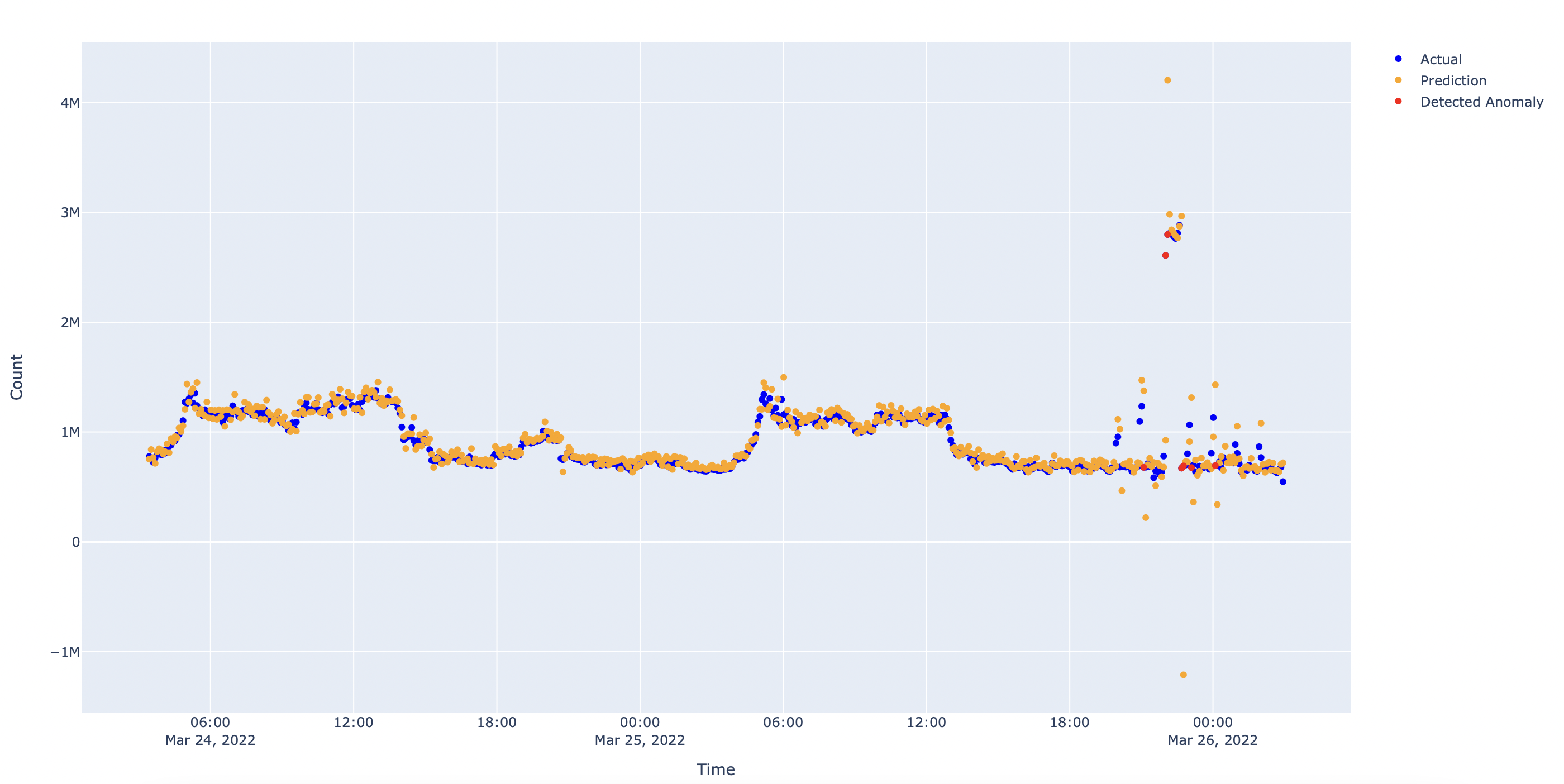
4.5 Результат

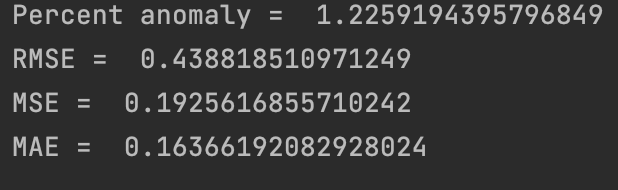
Первое, на что стоит обратить внимание - параметры модели ARIMA и ее ошибки.

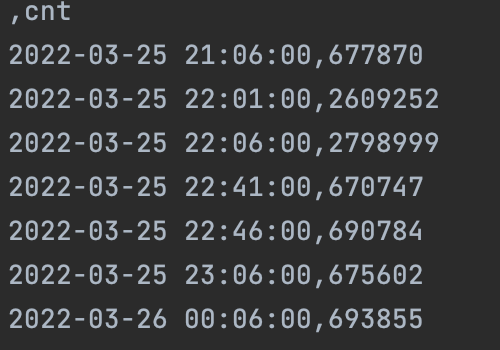
Запуская код, получаем файл примерно такого содержания. Весь файл можно посмотреть в приложении. Посмотрим, модель с какими параметрами имеет наименьшую MSE.Изображение

Теперь мы сможем сориентироваться, какие параметры можно использовать для получения хорошего результата. Начнем с обозначенных в начале 1, 1, 1.

Изображение

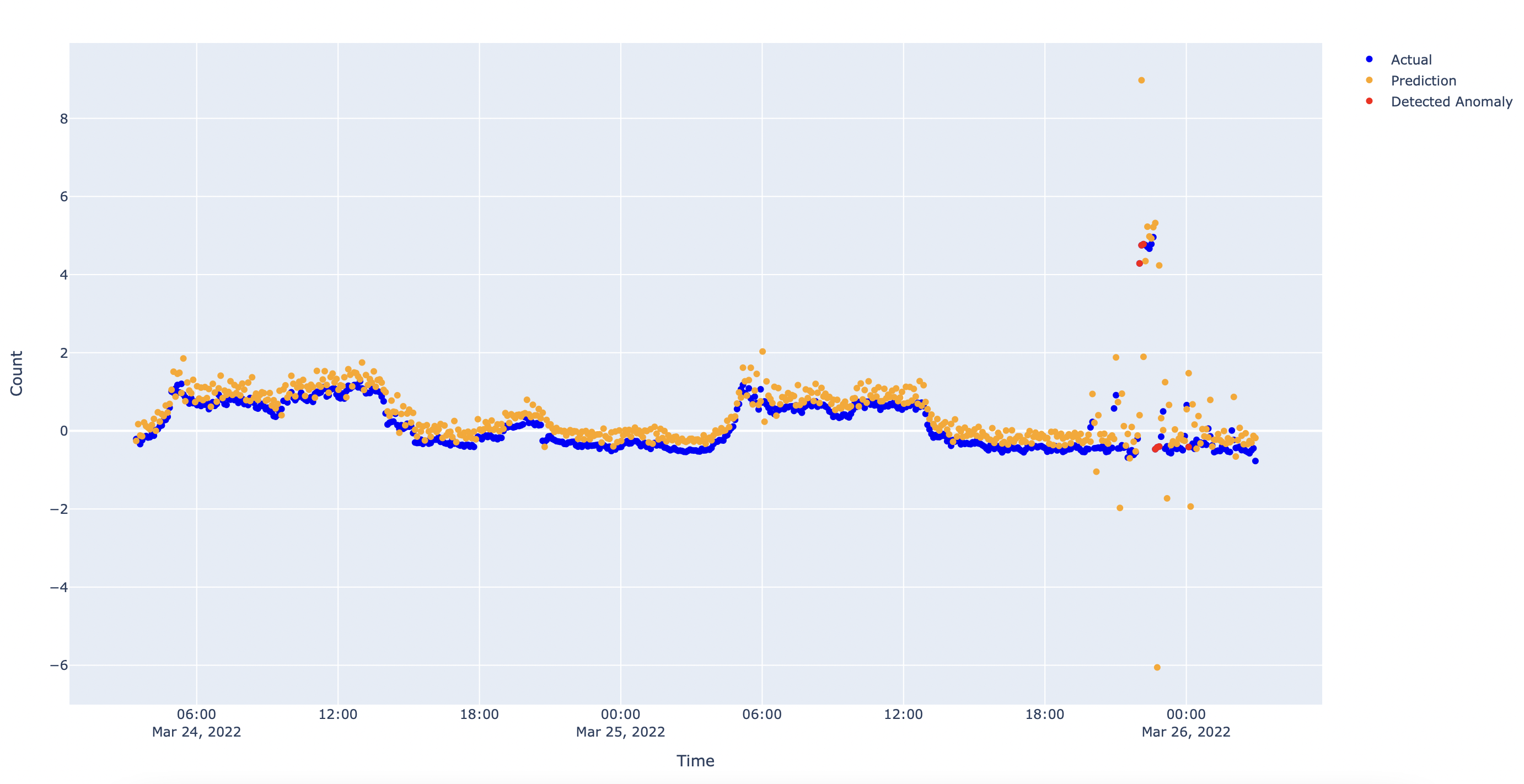
Полученный график:

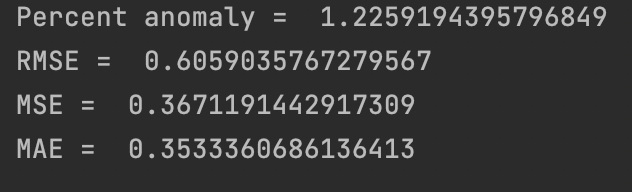
Вывод программы:

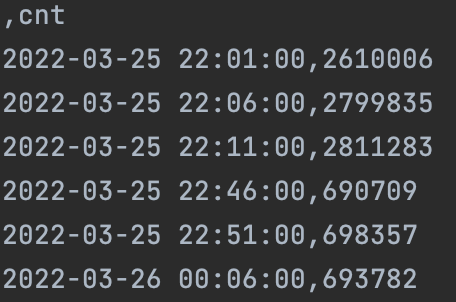
Файл csv с таблицей аномальных значений:

В тестовых данных было найдено 7 аномальных значений, что равняется ~1.2% общего объема тестовой выборки.

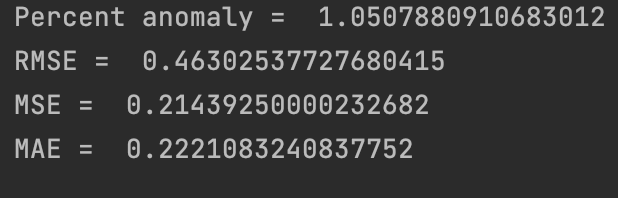
Теперь укажем лучшие параметры по результату перебора. Так же используем нормализацию данных.

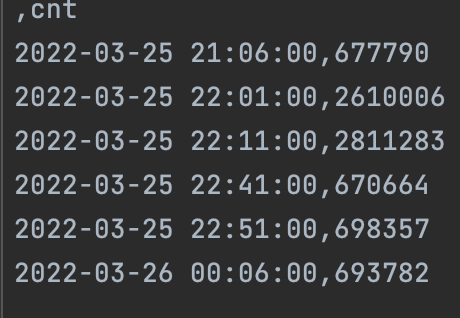
Полученный график:

Вывод программы:

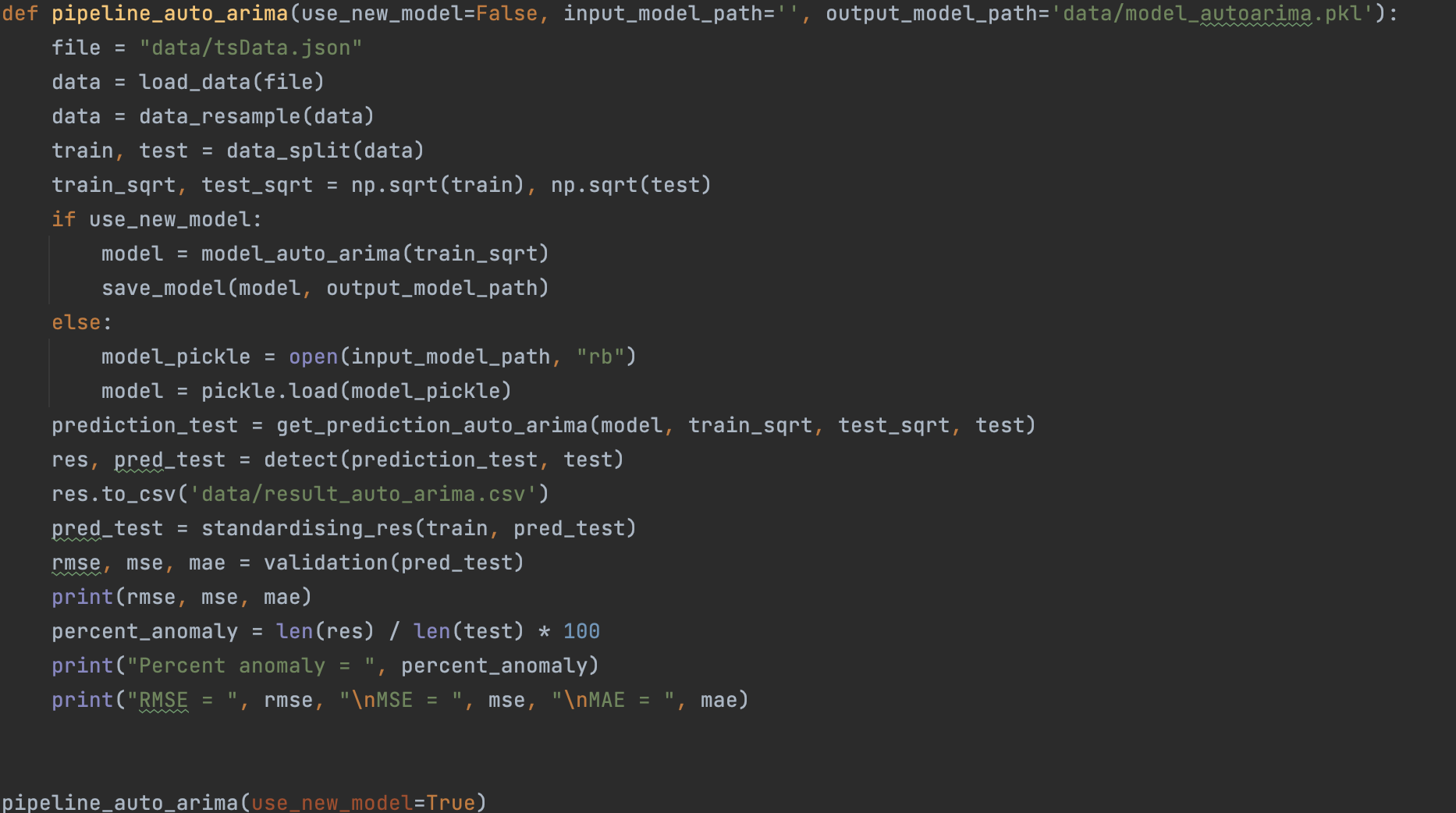
Файл csv:

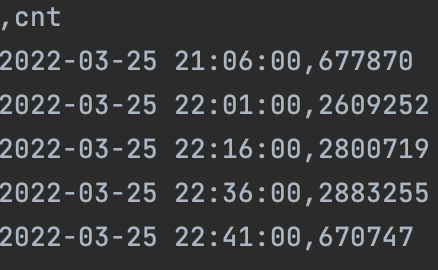
Отметим, процент аномалий не поменялся, но некоторые из строк оказались другими, а так же возросли ошибки.

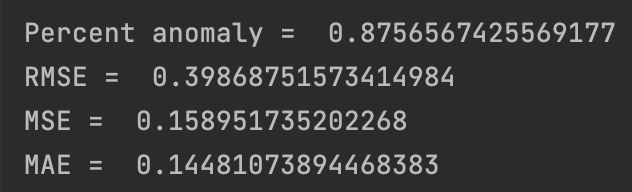
Заглянув в файл с параметрами моделей и их ошибками, видим что модель с параметрами 2, 1, 2 имеет так же относительно маленькую ошибку. Давайте используем эти параметры.

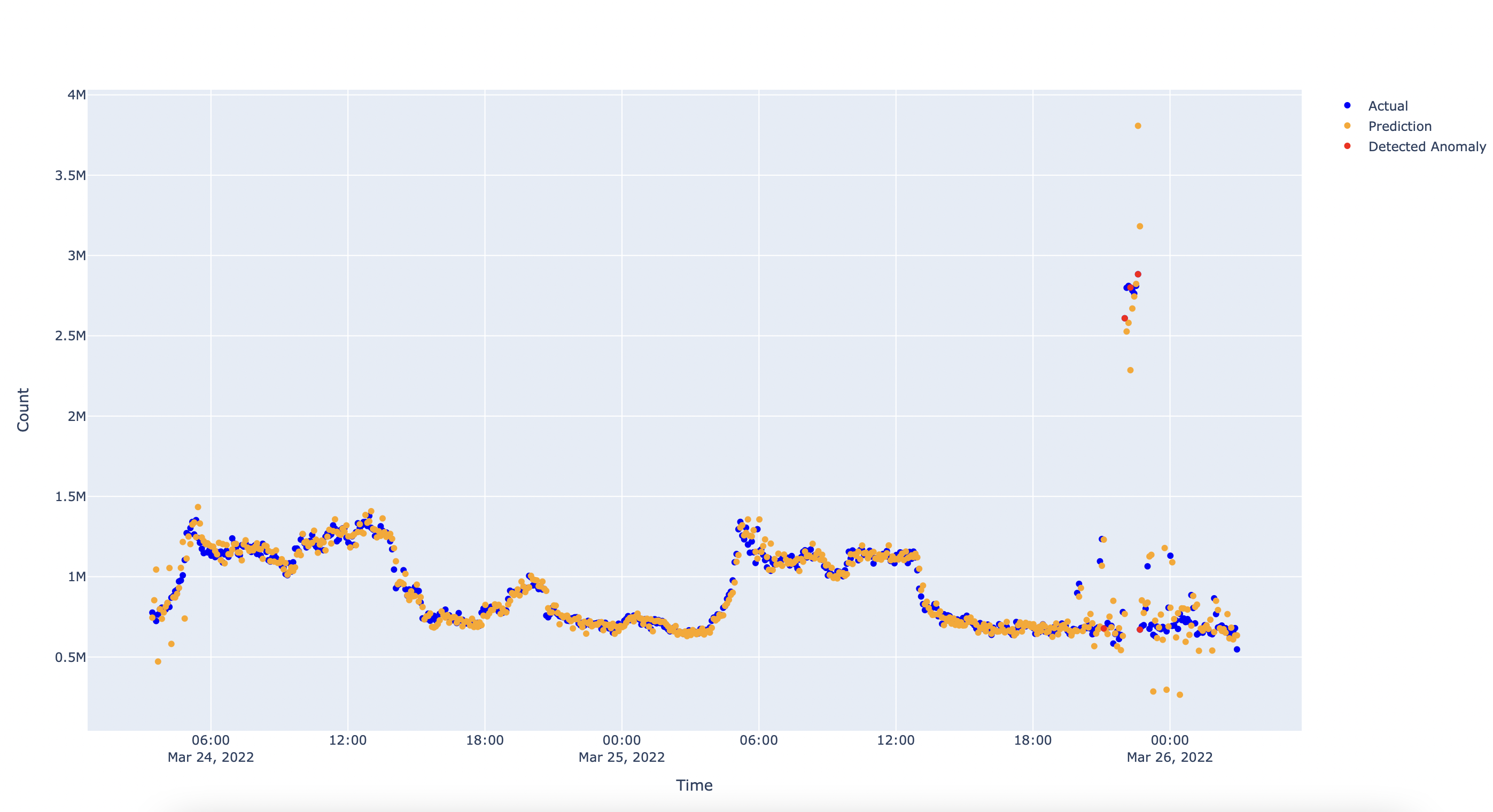
Процент уменьшился как и ошибки.

В результате получаем 6 аномальных значений. Уже лучше.

Что выдаст модель с автоматическим подбором параметров? Запустим код.







Мы нашли лучший результат.

ЗАКЛЮЧЕНИЕ

В работе был использован классический метод подгонки модели ARIMA - метод Бокса - Дженкинса.

Это процесс, который использует анализ временных рядов и диагностику для определения подходящих параметров модели ARIMA.

Таким образом, этапы следующие:

1. Идентификация. Использовались графики и сводная статистика для определения трендовых, сезонных и авторегрессионных элементов, чтобы понять величину отклонения и размер требуемой задержки.
2. Оценка параметров. Использовалась программа подбора, чтобы найти коэффициенты регрессионной модели.
3. Проверка модели.

Этот процесс описан в классическом учебнике 1970 года с темой анализ временных рядов Джорджа Бокс и Гвилима Дженкинса.

В результате был достигнут целевой показатель менее 1%. Это означает, что количество аномальных значений не превышает 1%. Значит, можно сделать вывод об эффективности алгоритма.

Удалось реализовать различные подходы к решению задачи, что заняло больше времени, но помогло разобраться в алгоритме прогнозирования и выявления аномалий. При выполнении задачи был получен уникальный для меня опыт.