«Санкт-Петербургский государственный университет»
Математико-механический факультет
Кафедра информационно-аналитических систем

Чудова Марина Юрьевна

Реализация пакета аналитических функций в PostgreSQL. Алгоритмы прогнозирования.

Выпускная квалификационная работа

Научный руководитель:

д. ф.-м. н., профессор Графеева Н. Г.

Рецензент:

ведущий инженер Отдела автоматизации,

Матвеева И. Е.

Санкт-Петербург

2018

SAINT-PETERSBURG STATE UNIVERSITY
Mathematics and Mechanics Faculty
Department of Analytical Information Systems

Chudova Marina Yurievna

Implementation a package of analytical functions. Forecasting algorithms

Graduation Project

Scientific supervisor:

Prof Grafeeva N.G.

Reviewer:

leading engineer of automatisation,

Matveeva Irina

Saint-Petersburg

2018

Оглавление

1. Введение

В современном мире получаемый нами объем информации слишком велик. Любая деятельность человека предусматривает, как минимум, ее содержание и анализ, ведь информация в чистом виде, поступающая непрерывным потоком трудна для восприятия. Поэтому существует множество способов хранения и обработки данных. Наиболее популярный из них - базы данных.

Сравнивая разные системы управления базами данных, в каждой можно найти свои плюсы и минусы. PostgreSQL [1, 4] стоит наряду со многими крупными СУБД, но по сравнению с другими она бесплатно распространяется. Однако, там не хватает пакетов аналитических функций для полноценного статистического анализа. Нужно сказать, что на сегодняшний день существуют различные способы обхода и решения этой проблемы, но в большинстве случаев это происходит за счёт сторонних программ и приложений, что имеет ряд своих минусов (таких, как возникновение конфликтных ситуаций, неудобство использования, и т.д.). Другие же методы не разрешают или разрешают не в полной мере эти вопросы. Поэтому было решено взяться за реализацию некоторых алгоритмов в этой системе управления базами данных.

2. Задачи

Целью данной работы является создание пакета аналитических методов, включающего в себя наиболее популярные алгоритмы прогнозирования, а также способы оценки ее качества.

Для достижения этой цели были сформулированы следующие задачи:

- Выполнить обзор существующих методов решения данной проблемы;

- Реализация алгоритмов прогнозирования средствами PostgreSQL, их объединение в пакет;
- Тестирование реализованных алгоритмов, проверка их качества с помощью метрик;
- Написание спецификации к пакету функций;
- Написание инструкции по установке и использованию пакета функций в PostgreSQL.

3. Обзор существующих решений

Одним из главных нововведений в области аналитических исследований в PostgreSQL является разработка компании Apache - библиотека MADlib [2]. Она находится в свободном доступе и реализует математические, статистические и машинные методы обучения. MADlib подходит для анализа данных с помощью классификации, кластеризации, описательной статистики, поиска ассоциативных правил и регрессии. Эта библиотека основана на трёх компонентах:

- Python driver functions - по большей части, отвечают за управление потоком алгоритмов.
- C++ implementation functions - являются C++ определением основных составляющих алгоритмов.
- Уровень абстракции базы данных C++ - предоставляют программный интерфейс для поддержания различных backend-платформ.

Ещё одно существующее разрешение проблемы отсутствия аналитических алгоритмов в PostgreSQL является расширение PL/R [3], разработанное Джо Конвэем. Оно позволяет писать функции PostgreSQL на языке статистических вычислений R. Но для его использования необходимо установить среду языка R на тот же сервер, что и PostgreSQL.

Эти способы решения, как уже говорилось, имеют свои недостатки. Хотелось бы иметь такой подход, который имел бы оптимальное и удобное решение.

4. Реализация алгоритмов прогнозирования

4.1 Модели экспоненциального сглаживания

- Одинарное экспоненциальное сглаживание:

  Экспоненциальное сглаживание позволяет отобразить тенденцию данных. Например, в биржевой деятельности часто необходимо видеть направление развития рынка.[5]

  В экспоненциальном сглаживании используется идея постоянного пересчета прогнозных значений по мере поступления фактических. Эта модель присваивает экспоненциально убывающие веса наблюдениям по мере их старения. Таким образом, последние доступные наблюдения имеют большее влияние на прогнозное значение, чем старшие наблюдения.

  В этом методе используется следующая формула для вычисления оценки для момента времени t:

$$S_t = ay_t + (1 - a)S_{t-1},$$

  где t>1, a — коэффициент сглаживания уровня, $0 \leq a \leq 1$.

  Начальные условия определяются как $S_1 = y_1$.

  В данной модели каждое последующее сглаженное значение S_t является

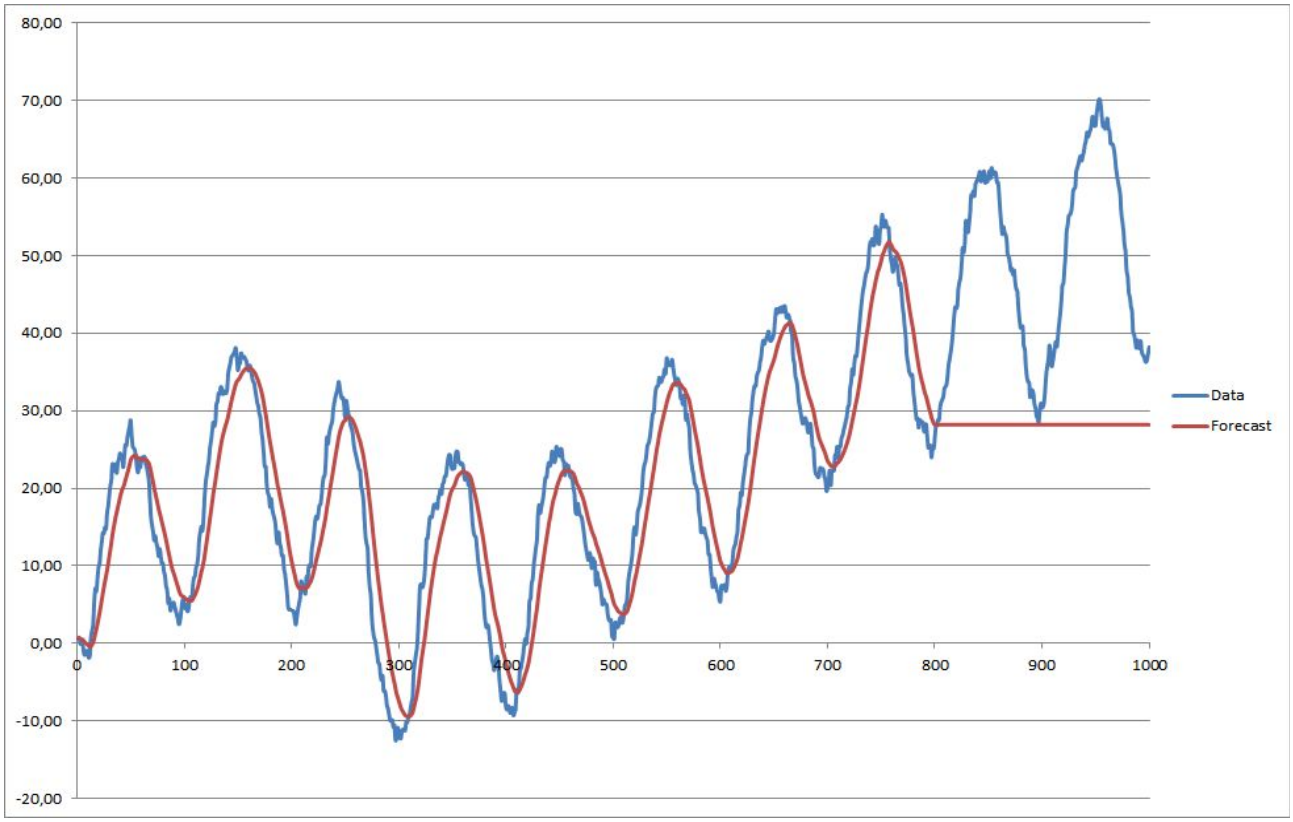
взвешенным средним между предыдущим значением временного ряда y_t и предыдущего сглаженного значения S_{t-1} .

Коэффициент сглаживания уровня мы выбираем сами в диапазоне от 0 до 1. Чем ближе ее значение к 1, тем больше сглаженные значения будут похожи на исходные, и, соответственно, чем коэффициент сглаживания ближе к 0, тем сильнее результат будет отличаться от изначальных данных.

Данный алгоритм прогнозирования в PostgreSQL представляет собой функцию expForecast со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **alpha** - коэффициент сглаживания уровня;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся 20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



- Двойное экспоненциальное сглаживание:

Двойное экспоненциальное сглаживания применяется для моделирования процессов, имеющих тренд. Учитывание трендовых тенденций делает эту модель более точной, чем предыдущая.

В этом методе используется следующие формулы для вычисления оценки для момента времени t:

$$S_t = ay_t + (1-a)(S_{t-1} - B_{t-1}),$$

$$B_t = \gamma(S_t - S_{t-1}) + (1-\gamma)B_{t-1},$$

где t>1, a — коэффициент сглаживания уровня, $0 \le a \le 1$,

γ — коэффициент сглаживания тренда, $0 \le \gamma \le 1$.

Начальные условия определяются как $S_1 = y_1$. Для исходных значений трендовой же компоненты существует несколько вариантов формул. Например:

$$B_1 = y_2 - y_1,$$

$$B_1 = \frac{1}{3}[(y_2 - y_1) + (y_3 - y_2) + (y_4 - y_3)],$$

$$B_1 = \frac{y_n - y_1}{n-1},$$

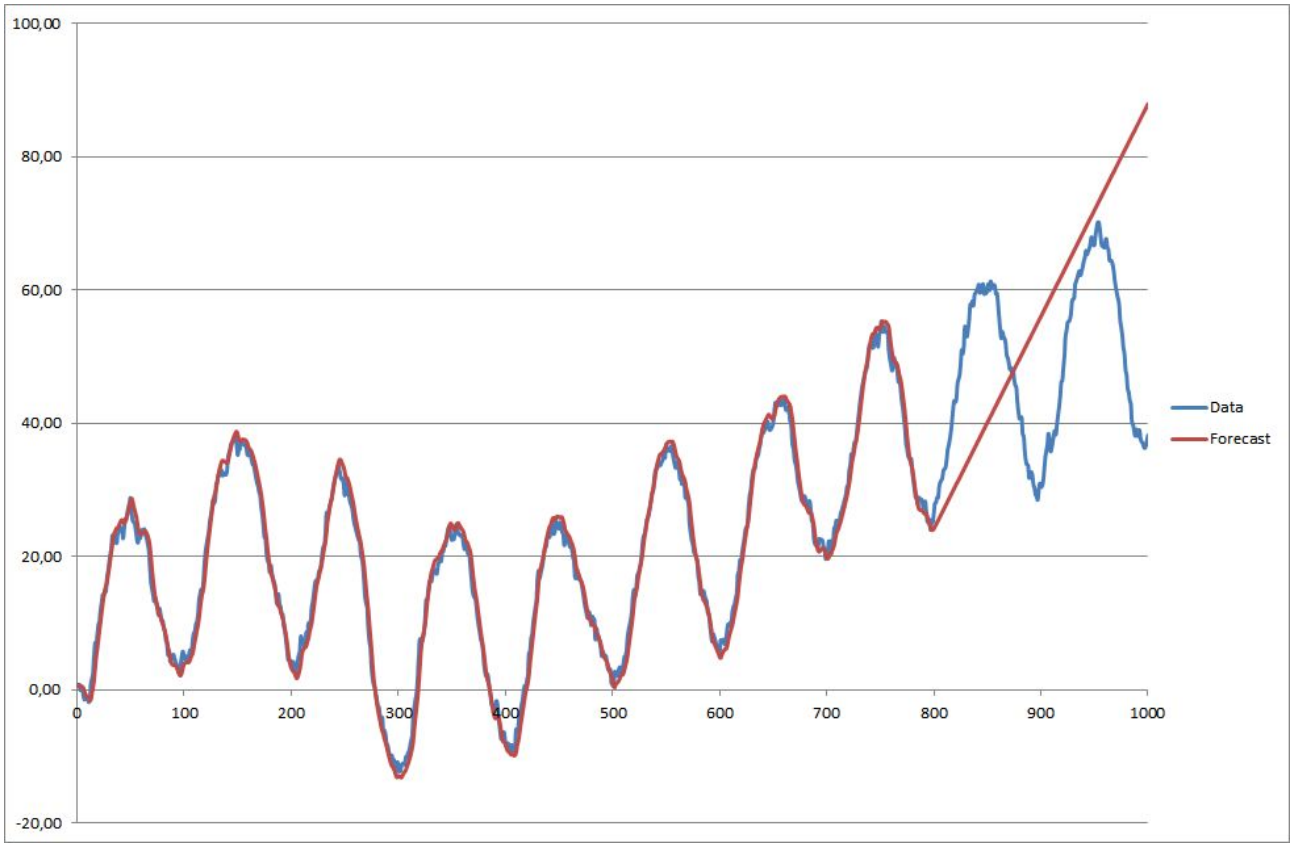
где n - количество значений исходных данных.

Так же, как и в предыдущей модели, коэффициенты сглаживания уровня и тренда мы выбираем сами в том же диапазоне от 0 до 1. Но если с первым значением все понятно (правила те же, что и в предыдущей модели), то со второй сложнее. Как правило, самое подходящее значение устанавливается с помощью перебора некоторых значений в диапазоне от 0 до 1, а в качестве целевой функции можно использовать какую-либо функцию для оценки качества прогнозирования.

Данный алгоритм прогнозирования в PostgreSQL представляет собой функцию expTrendForecast со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **alpha** - коэффициент сглаживания уровня;
- **beta** - коэффициент сглаживания тренда;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся 20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



- Тройное экспоненциальное сглаживание:

Тройное экспоненциальное сглаживание, или модель Хольта-Винтерса [6], применяется для моделирования процессов, которые, помимо тренда, имеют также и сезонную составляющую. Соответственно, эта модель является еще более точной, по сравнению с предыдущими двумя.

В этом методе используется следующие формулы для вычисления оценки для момента времени t:

$$S_t = a\frac{y_t}{I_{t-L}} + (1-a)(S_{t-1} - B_{t-1}),$$

$$B_t = \gamma(S_t - S_{t-1}) + (1-\gamma)B_{t-1},$$

$$I_t = \beta\frac{y_t}{S_t} + (1-\beta)I_{t-L},$$

где t>1, a — коэффициент сглаживания уровня, $0 \le a \le 1$,

γ — коэффициент сглаживания тренда, $0 \le \gamma \le 1$,

β — коэффициент сезонной составляющей, $0 \le \beta \le 1$,

L — длина периода (должна быть определена заранее).

Исходные данные должны содержать как минимум два периода. Начальные условия определяются как $S_1 = y_1$. Исходные значения трендовой же компоненты задаются формулой:

$$B_1 = \frac{1}{L}\left(\frac{y_{L+1} - y_1}{L} + \frac{y_{L+2} - y_2}{L} + \ldots + \frac{y_{L+L} - y_L}{L}\right).$$

Расчет начальных значений для индексов сезонности производится по формуле:

$$I_i = L\frac{y_i}{\sum_1 y_i},$$

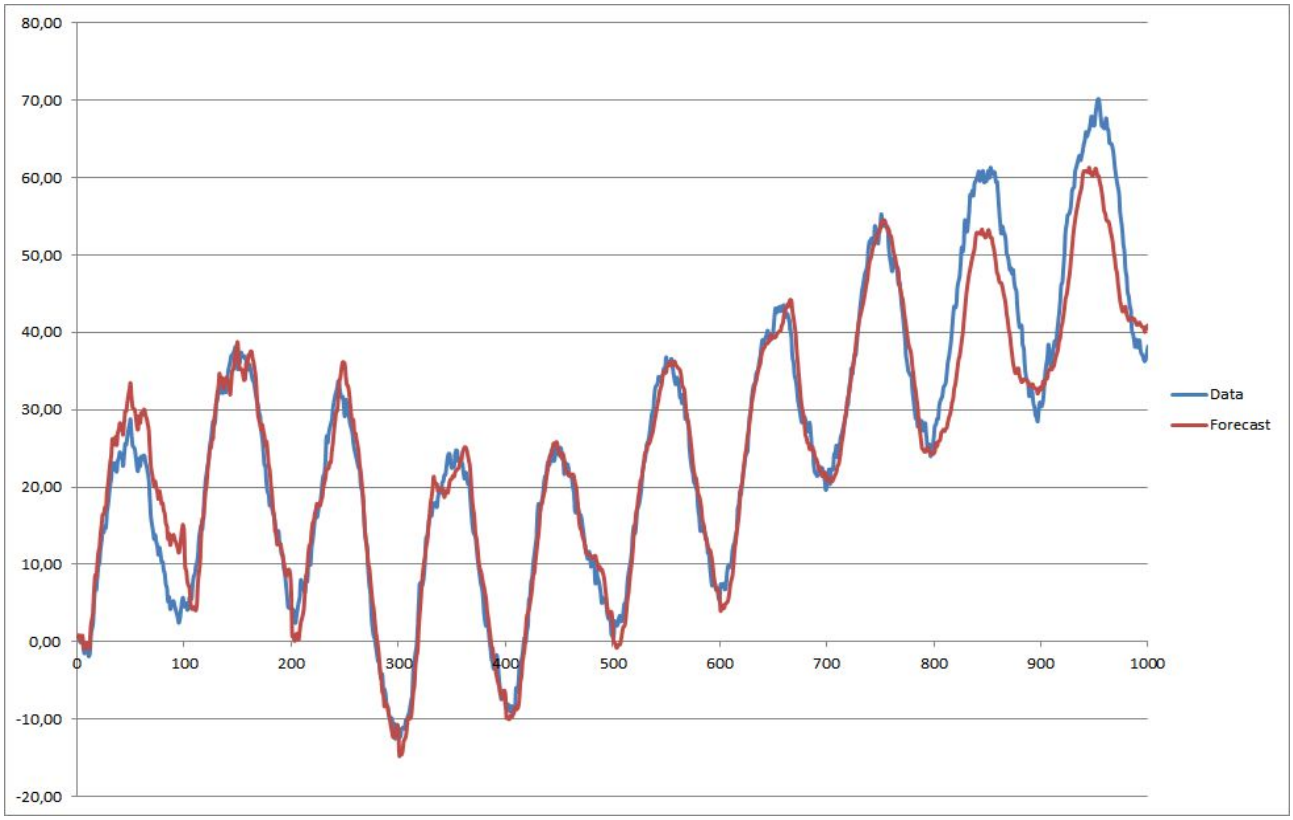
где $1 \le i \le L$.

Так же, как и в предыдущих моделях, все коэффициенты мы выбираем сами в том же диапазоне от 0 до 1. Как подобрать значение для первого коэффициента описано в пункте “Одинарное экспоненциальное сглаживание“. Для других двух обычно самое подходящее значение устанавливается с помощью перебора некоторых значений в диапазоне от 0 до 1, а в качестве целевой функции можно использовать какую-либо функцию для оценки качества прогнозирования.

Данный алгоритм прогнозирования в PostgreSQL представляет собой функцию holtWintersForecast со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **alpha** - коэффициент сглаживания уровня;
- **beta** - коэффициент сглаживания тренда;
- **gamma** - коэффициент сезонной составляющей;
- **P -** количество периодов;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся 20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



4.2 Линейная регрессия

Эта модель является самой простой среди других регрессионных алгоритмов. Она основана на том, что существует лишь один внешний фактор - время. Причем связь между временем и процессом линейна. Модель прогнозирования линейной регрессии описывается уравнением:

$$Z_t = a_0 + a_1 X_t,$$

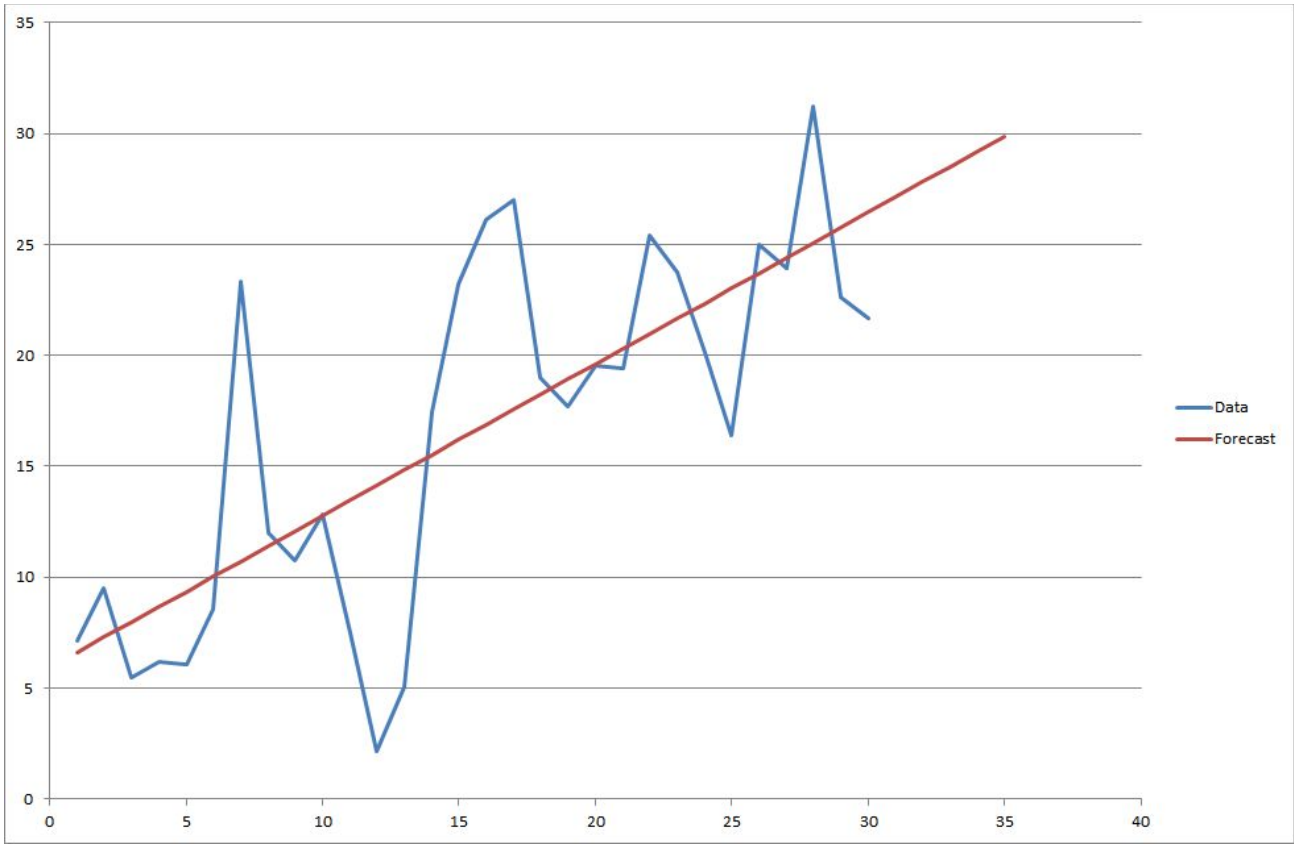
где t>0, a_0, a_1 — коэффициенты регрессии, которые можно получить с помощью метода наименьших квадратов или метода максимального правдоподобия. В данной работе был использован первый.

На практике эта модель прогнозирования показывает общее направление движение процесса.

Данный алгоритм прогнозирования в PostgreSQL представляет собой функцию linearRegression со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся 20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



4.3 Авторегрессионная модель прогнозирования

Авторегрессионные модели основаны на идее, что значение процесса линейно зависит от некоторого количества предыдущих значений того же процесса.

Одна из популярнейших моделей данного класса - ARIMA [7, 9, 10] - авторегрессия проинтегрированного скользящего среднего. Этот алгоритм наиболее обширный, так как включает в себя несколько функций. Поэтому основной упор был сделан именно на него. В этом разделе будет описана каждая функция отдельно, а также работа всего алгоритма.

- Авторегрессия

Модель авторегрессии [8] очень эффективна в прогнозировании некоторых временных рядов. Здесь значение процесса в момент времени t выражается, как линейная комбинация предыдущих значений процесса.

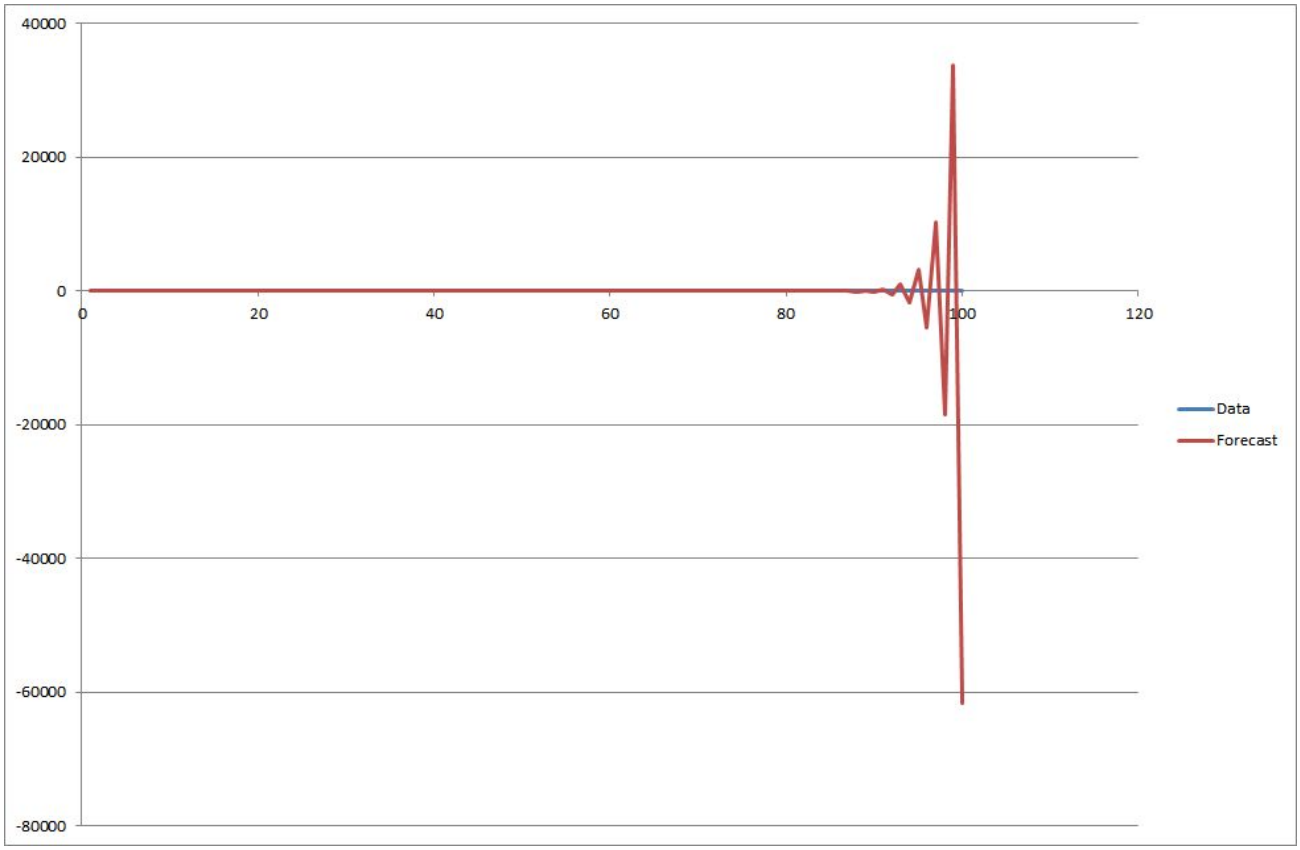
$$Z_t = a_0 + a_1 Z_{t-1} + a_2 Z_{t-2} + \ldots + a_p Z_{t-p},$$

Количество значений процесса в предыдущие моменты времени, на основе которых рассчитывается текущее, обозначается, как правило p, и называется порядком авторегрессии. Для определения коэффициентов авторегрессионной модели используют метод наименьших квадратов (используется в данной работе) или метод максимального правдоподобия.

Авторегрессия в данном пакете аналитических функций PostgreSQL представляет собой функцию AR со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **p -** порядок авторегрессии;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся 20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



Как видно на графике, эта модель не дает хорошего прогноза. Это объясняется тем, что не для каждого временного ряда подходит авторегрессия. Но внутри алгоритма ARIMA, как будет показано позже, эффективность и достоверность данных в разы улучшается.

- Модель скользящего среднего

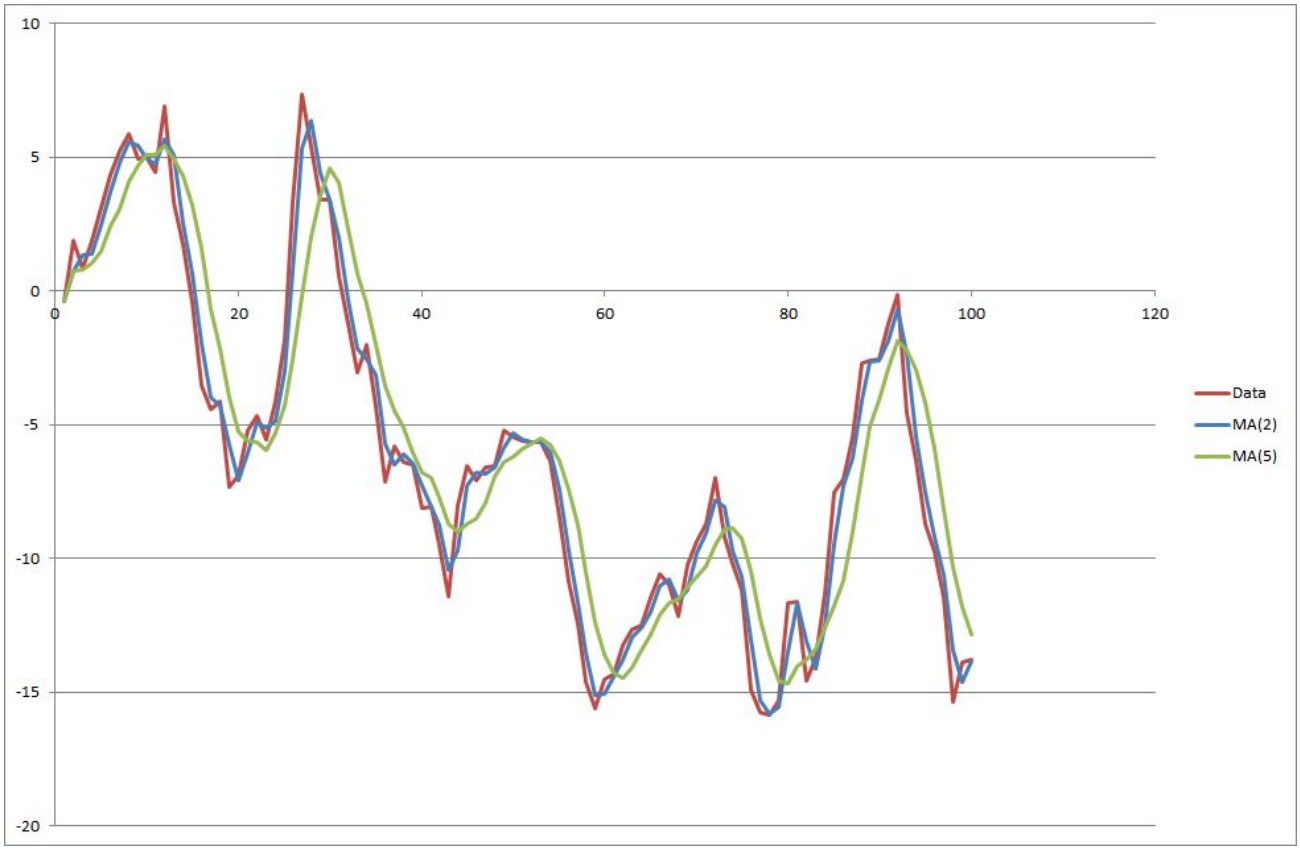
Данная часть алгоритма ARIMA имеет функцию сглаживания. Модель скользящего среднего редко используется самостоятельно. Намного чаще - совместно с авторегрессией. Эта функция также имеет свой порядок q - степень сглаживания.

$$Z_t = \frac{1}{q}(Z_{t-1} + Z_{t-2} + ... + Z_{t-q}),$$

Модель скользящего среднего в PostgreSQL представлена функцией MA со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **q -** степень сглаживания.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а графики MA(2) и MA(5) - сглаженные данные.



- Интегрирование

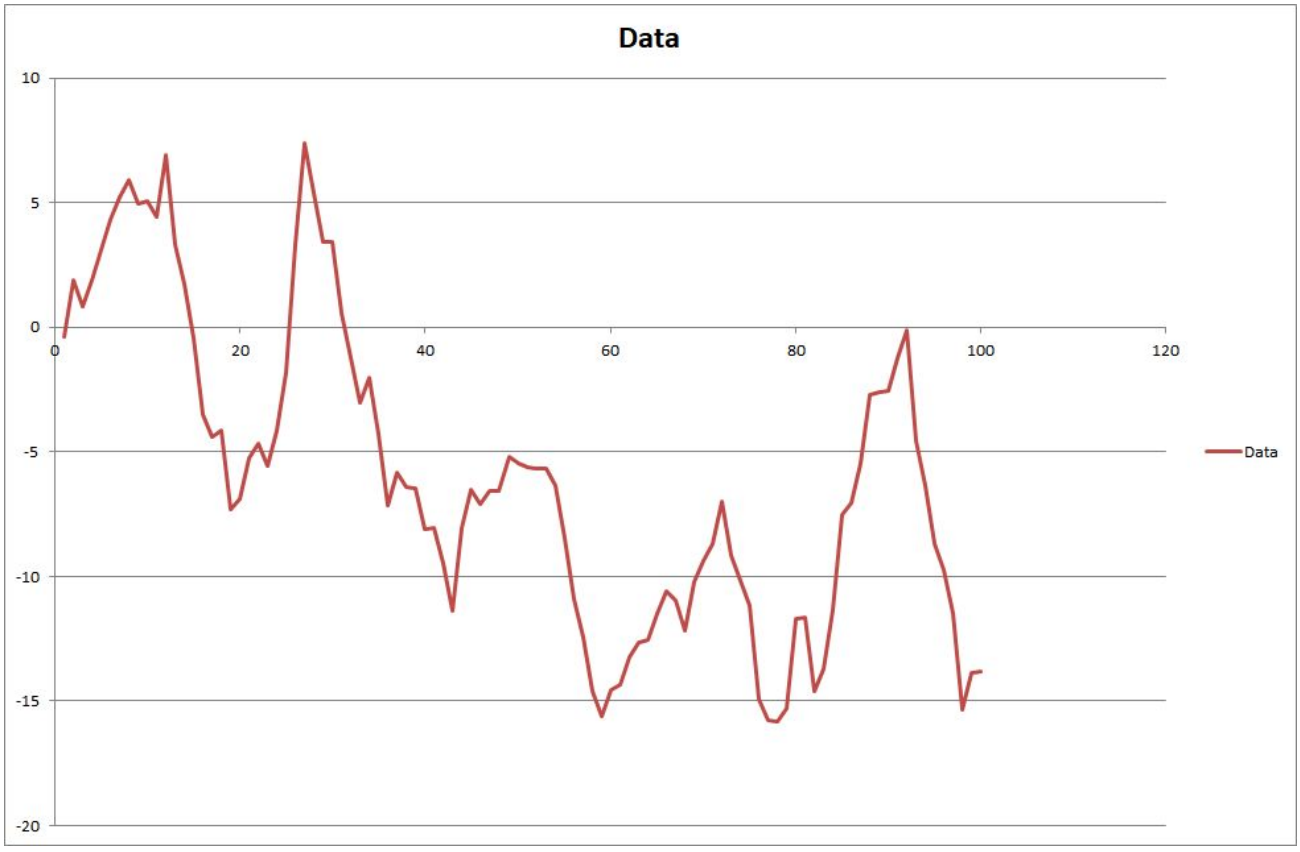
Здесь в качестве входных данных используются не сами значения временного ряда, а их разность d-того порядка. Прогнозировать не исходные данные, а только их изменения обычно бывает намного точнее. Как правило, порядок d задает пользователь, но на практике чаще берут d=2 ввиду лучшего

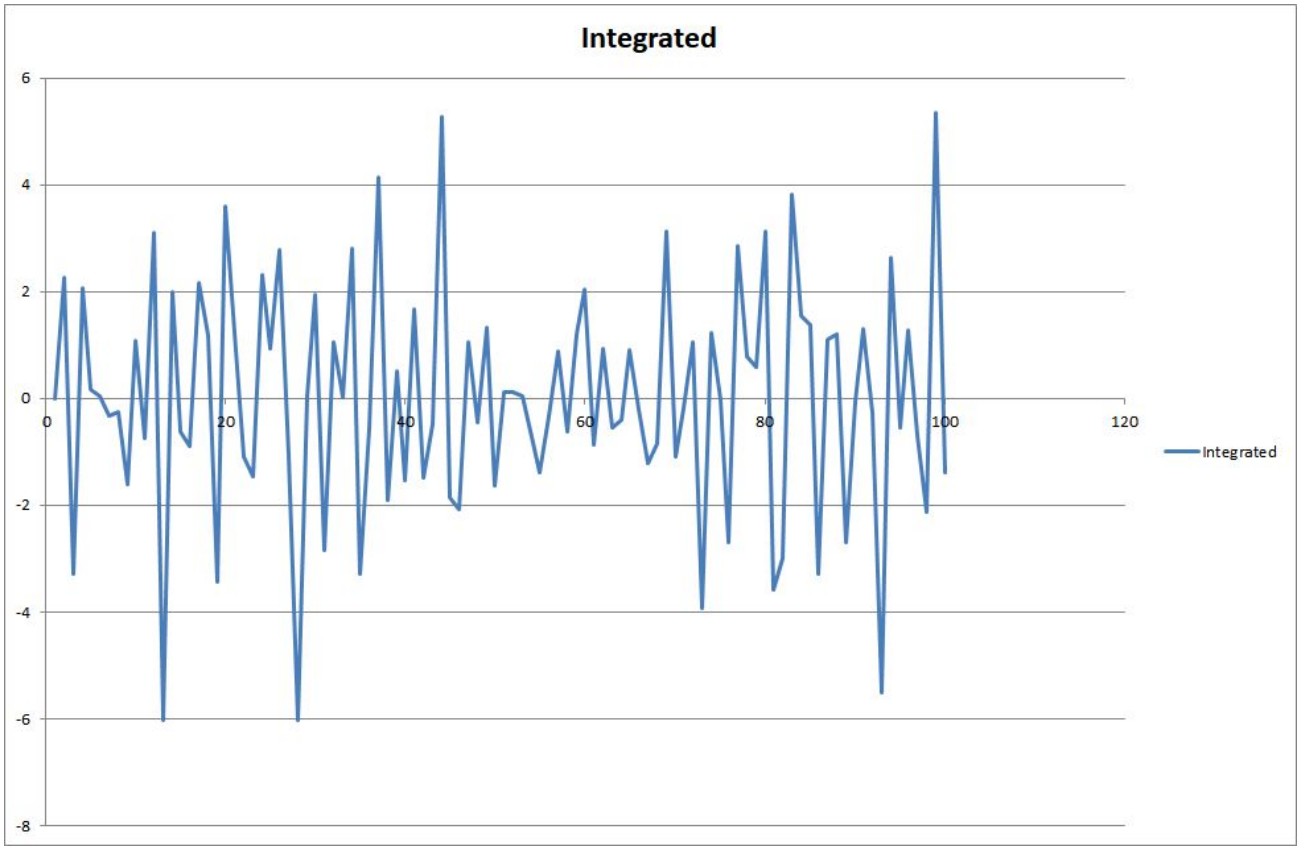
прогнозирования. Но также известна и доказана неэффективность моделей с d>2.

Модель скользящего среднего в PostgreSQL представлена функцией I со следующими параметрами:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;

На графике ниже изображена работа этой функции. Первый график Data представляет собой исходные данные, а второй - Integrated - график изменений начальных значений с порядком 2.



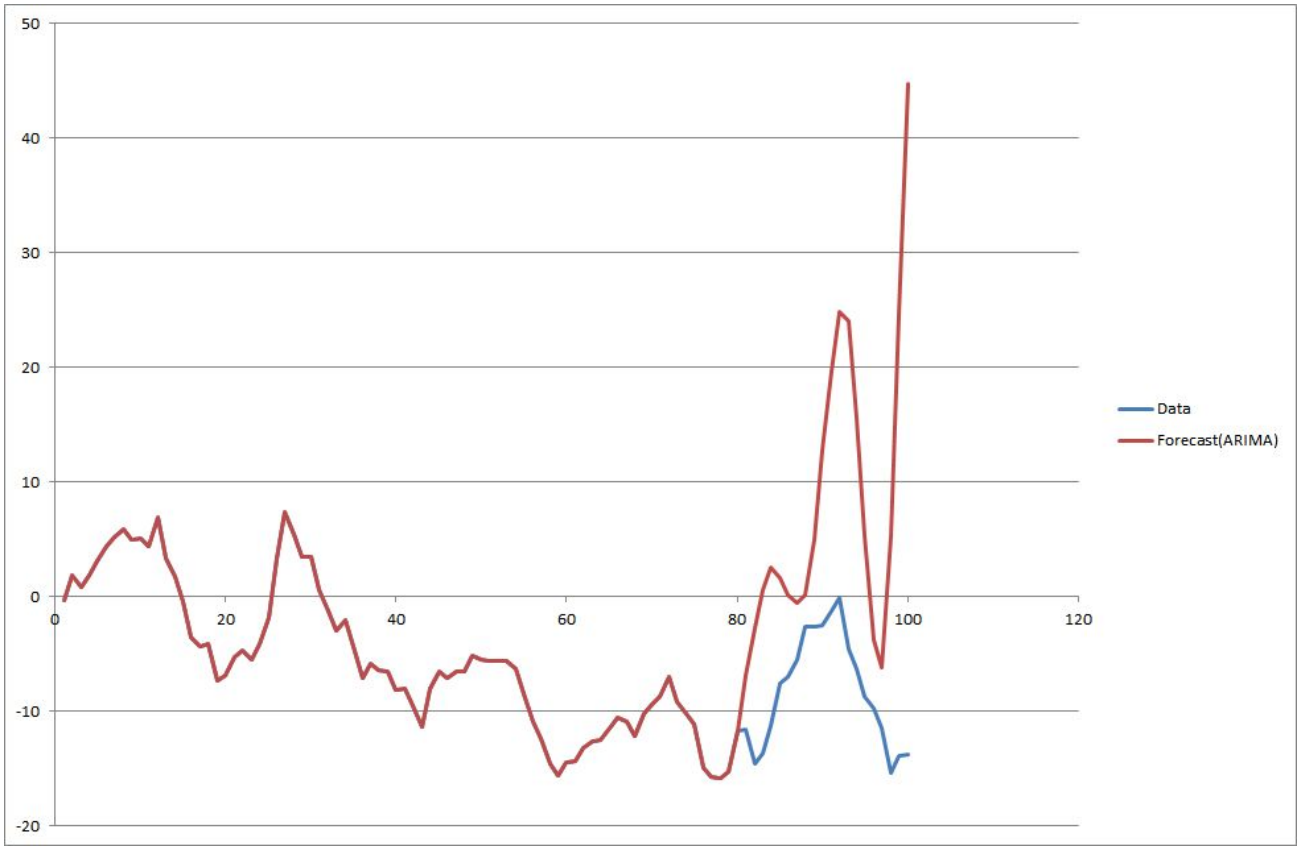


Все эти функции были объединены в одну - ARIMA. Ее параметрами являются:

- **_tbl -** название таблицы с данными - временным рядом и значениями некоторых признаков в данный момент времени;
- **p -** порядок авторегрессии;
- **q -** степень сглаживания;
- **l -** количество строк из таблицы, данные которых будут сглажены и использованы для вычисления прогнозных значений;
- **ll -** количество строк с прогнозными значениями, которое мы хотим получить.

На графике ниже изображена работа этой функции. График Data представляет собой исходные данные, а график Forecast - сглаженные и спрогнозированные данные. Для наглядности 80% исходных данных было сглажено и использовано для вычисления прогнозных значений, оставшиеся

20% показаны на графике для сравнения реальных значений с прогнозными значениями, которые вычислила функция.



Таким образом, авторегрессия работает намного эффективней с моделью скользящего среднего и интеграцией второго порядка.

4.5 Метрики

Чтобы проверить качество того или иного алгоритма, необходимо оценить предсказанный им прогноз. Для этого существуют различные метрики оценки. Среди них есть метрики, основным недостатком которых является то, что само по себе их значение сложно интерпретировать. С их помощью можно лишь сравнивать работу одних моделей с другими моделями на общих данных. То есть значения метрики сильно зависят от данных. Назовем такие метрики субъективными, а все прочие - объективными.

Субъективные метрики:

- Средняя абсолютная ошибка прогноза MAE - представлена функцией MAE;
- Среднеквадратичное отклонение RMSE - представлена функцией RMSE;
- Средний процент ошибки MPE - представлена функцией MPE;
- Средняя относительная ошибка MAPE - представлена функцией MAPE;
- Абсолютное отклонение от среднего AD - представлена функцией AD;
- Среднее абсолютное отклонение MAD - представлена функцией MAD;

Объективные метрики:

- Коэффициент детерминации - представлена функцией R2. Область значений от 0 до 1;
- Коэффициент несоответствия Тейла - представлена функцией THEIL. Область значений от 0 до inf.

4.6 Прочие функции

В том числе в рамках схемы были реализованы вспомогательные функции, перечень которых приведен ниже.

- timeSeriesShow - функция предназначенная для внутреннего обращения; по названию таблицы, содержащей временной ряд и значения процесса, возвращает содержимое этой таблицы;
- randomTimeSeries - функция по заданным параметрам, случайно генерирующая временной ряд и значения процесса;
- Determinant - функция, по заданной в параметрах матрице, возвращающая её определитель;
- Cramer - функция, по заданным матрице и вектору, решающую систему линейных уравнений;
- TransposeMatrix - функция, транспонирующая матрицу, переданную как параметр;

- InvertMatrix - функция, возвращающая матрицу, обратную данной в параметрах;
- MultMatrix - функция, возвращающая произведение двух матриц, переданных как параметры;
- OLS - функция, решающая метод наименьших квадратов.

В тексте подробнее эти функции рассмотрены не будут, так как они отклоняются от темы работы.

5. Заключение

В рамках данной работы был произведен обзор наиболее популярных и полезных алгоритмов прогнозирования, а также метрик оценки их качества. Позднее на языке PL/pgSQL был разработан пакет аналитических функций, разворачиваемый в СУБД PostgreSQL и, включающий в себя наравне с другими функции, реализующие такие алгоритмы как: экспоненциальное сглаживание, линейная регрессия, авторегрессия, ARIMA, а также алгоритмы оценки качества прогнозирования. Результат разработки был опубликован на портале GitHub и доступен по ссылке https://github.com/Twicelab/anfun. К коду приложены файлы спецификации и инструкции по установке. Метод разработки продукта позволяет масштабировать его и добавлять функциональность в будущем, расширяя возможности и оптимизируя его работу.

6. Список литературы

[1] PostgreSQL documentation – URL: https://www.postgresql.org/docs/

[2] Apache MADlib documentation – URL: http://madlib.apache.org/documentation.html

[3] PL/R documentation – URL: http://www.joeconway.com/doc/doc.html

[4] Нейл Мэттью, Ричард Стоунз – “PostgreSQL. Основы”, 2002

[5] Чучуева Ирина Александровна – “Модель Прогнозирования Временных Рядов по Выборке Максимального Подобия”, 2012, МГТУ им. Баумана

[6] Prajakta S. Kalekar - “Time series Forecasting using Holt-Winters Exponential Smoothing”, 2004, Kanwal Rekhi School of Information Technology

[7] Chen C. F., Chang Y. H., Chang Y. W. Seasonal ARIMA forecasting of inbound air travel arrivals to Taiwan //Transportmetrica. – 2009. – Т. 5. – №. 2. – С. 125-140.

[8] Hurvich C. M., Tsai C. L. Regression and time series model selection in small samples //Biometrika. – 1989. – Т. 76. – №. 2. – С. 297-307.

[9] Cortez P., Rocha M., Neves J. Evolving time series forecasting ARMA models //Journal of Heuristics. – 2004. – Т. 10. – №. 4. – С. 415-429.

[10] Zhang G. P. Time series forecasting using a hybrid ARIMA and neural network model //Neurocomputing. – 2003. – Т. 50. – С. 159-175.