**БАКАЛАВРСКАЯ РАБОТА**

**НА ТЕМУ**

**Исследование и разработка прозрачного метода прогнозирования**

**временных рядов**

**АННОТАЦИЯ**

Объем выпускной квалификационной работы составляет 63 страниц,   
3 главы, введение, заключение, 29 рисунков, 1 приложение, использовано   
12 источников.

Выпускная квалификационная работа посвящена исследованию и разработки прозрачного метода прогнозирования временных рядов. Содержит в себя введение, три главы и заключение. Модели которые были разработаны в данная работе позволяет анализировать и прогнозировать различные входные данные например: количество ежедневных рождений в Квебеке, температуру, производство электроэнергии в Австралии. В работе рассмотрены модели AR, MA, ARIMA, SARIMA, быстрое преобразование Фурье, закрытая периодическая установка, долгосрочная кратковременная память, нейронная сеть ,рекуррентная нейронная сеть, SAPTF, одновекторная регрессия. На основе рассмотренных моделей было спроектировано и реализовано модели AR, MA, ARIMA, SARIMA. Были представлены алгоритмы работы каждого из перечисленных моделей.

Содержание

[Введение 7](#_Toc106396752)

[Список сокращений 10](#_Toc106396753)

[1. Анализ предметной области 11](#_Toc106396754)

[1.1 Что такое временной ряд 11](#_Toc106396755)

[1.2 Типы временных рядов 12](#_Toc106396756)

[1.3 Почему мы проводим анализ временных рядов 13](#_Toc106396757)

[1.4 Компоненты временного ряда 14](#_Toc106396758)

[2. Методы исследования временных рядов 15](#_Toc106396759)

[2.1 Как прогнозировать временные ряды 15](#_Toc106396760)

[2.1.1 Прогнозирование временных рядов 15](#_Toc106396761)

[2.1.2 На шаг впереди 16](#_Toc106396762)

[2.1.3 Многоступенчатый 16](#_Toc106396763)

[2.1.4 Перекрестная проверка временных рядов 17](#_Toc106396764)

[2.2 Прогнозирование временных рядов SAP (SAPTF) 20](#_Toc106396765)

[2.3 ARMA и его варианты 21](#_Toc106396766)

[2.3.1 АР(р) 21](#_Toc106396767)

[2.3.2 MA(q) 22](#_Toc106396768)

[2.3.3 ARMA(p,q) 22](#_Toc106396769)

[2.3.4 ARIMA(p,d,q) 23](#_Toc106396770)

[2.3.5 SARIMA(p, d, q)(P, D, Q)S 23](#_Toc106396771)

[2.3.6 Функция автокорреляции и частичной автокорреляции 24](#_Toc106396772)

[2.4 Прогнозист на основе преобразования Фурье 27](#_Toc106396773)

[2.4.1 Оценщик БПФ, как он работает 27](#_Toc106396774)

[2.4.2 Прогнозирование числа ежедневных рождений в Квебеке набор данных с БПФ 28](#_Toc106396775)

[2.5 Facebook Prophet 29](#_Toc106396776)

[2.5.1 Полезные дополнения к традиционной аддитивной модели регрессии 30](#_Toc106396777)

[2.6 Регрессия опорных векторов (SVR) 31](#_Toc106396778)

[2.6.1 Теоретическая формулировка СВР 31](#_Toc106396779)

[2.7 Подход к глубокому обучению 32](#_Toc106396780)

[2.7.1 Нейронная сеть 32](#_Toc106396781)

[2.7.2 Рекуррентная нейронная сеть (RNN) 32](#_Toc106396782)

[2.7.3 Архитектура моделей RNN 34](#_Toc106396783)

[2.7.4 Вейвлеты + нейронные сети 35](#_Toc106396784)

[2.7.5 Обзор архитектуры 35](#_Toc106396785)

[2.8 Анализ и выбор модели 36](#_Toc106396786)

[2.8.1 Эксперименты и результаты 37](#_Toc106396787)

[2.9 Обоснование выбора среды разработки и языка программирования 39](#_Toc106396788)

[3. Программная реализация 41](#_Toc106396789)

[3.1 Описание работы моделей 41](#_Toc106396790)

[3.2 Реализация моделей прогнозирования 43](#_Toc106396791)

[3.2.1 Авторегрессионные модели 43](#_Toc106396792)

[Заключение 56](#_Toc106396793)

[Список использованных источников 58](#_Toc106396794)

[Приложение А. Программный код приложения 59](#_Toc106396795)

# Введение

Моделирование временных рядов — это динамично развивающаяся область исследований, которая в последние десятилетия привлекала внимание исследовательского сообщества. Основная цель моделирования временных рядов состоит в том, чтобы собрать, проанализировать и тщательно изучить прошлое и разработать соответствующую модель, описывающую внутреннюю структуру ряда. Затем эта модель используется для объяснения временного ряда и прогнозирования его будущих значений, т.е. прогнозирования.

Прогнозирование временных рядов является одной из наиболее распространенных и используемых учебных задач. Каждый день предприятия используют прогнозирование временных рядов для самых разных целей, таких как прогнозирование ежедневных цен на акции, прогнозирование обменных курсов иностранной валюты, прогнозирование уровня безработицы; метеорологи используют его для оценки скорости ветра, дневных максимальных и минимальных температур и приблизительного количества осадков. Все эти и многие другие задачи показывают важность временных рядов и важность хорошей оценки будущего, поскольку для бизнеса может быть первостепенной задачей подготовиться к потенциальному всплеску/падению своих продаж и подготовиться к нему или избежать катастрофы при наблюдении. метеорологические данные.

Прогнозирование временных рядов — это действие по предсказанию будущего после тщательного рассмотрения и анализа прошлого в связи с незаменимой важностью этой задачи во многих областях, таких как финансы, метеорология, бизнес, наука и техника. Подготовка адекватной модели для подбора и последующего прогнозирования ряда не является очевидной задачей, поскольку каждый сигнал/ряд имеет свои собственные свойства и зависимости от экзогенных параметров, которые не могут быть легко представлены в модели. За прошедшие годы ученые, статистики и экономисты предложили множество исследований и моделей для повышения точности прогнозов. В результате были запущены/улучшены различные модели временных рядов, но такое обилие моделей не обязательно означает, что эти модели применимы повсеместно. Наиболее популярными и используемыми подходами по-прежнему являются статистические модели, такие как ARIMA и ES, и модели регрессии машинного обучения (соответствующим образом применяемые к временным рядам), такие как SVR. Но в последние несколько лет RNN становятся все более и более конкурентоспособными по сравнению с традиционными подходами и обеспечивают гораздо большую расширяемость ранее упомянутых моделей. Их можно использовать с широким спектром данных (экзогенных), которые могут обогатить понимание, обеспечиваемое временными рядами.

Главной целью дипломной работы является разработка ПО для прогнозирования временных рядов с помощью модели временных рядов.

Для достижения этой цели необходимо решить следующие задачи:

Изучить методы анализа временных рядов.

Разработать модели для прогнозирования временных рядов, использующую на этапе формирования модели временных рядов.

Если мы хотим писать на python, то Jupiter notebook отличным преимуществом для нас что бы видеть результат выполнения кода и его отдельных фрагментов. В такой среде разработки можно, например, написать функцию и сразу проверить её работу, без запуска программы целиком. А ещё можно поменять порядок выполнения кода. Можно отдельно загрузить файл в память, отдельно проверить его содержимое, отдельно обработать содержимое.

Данная работа состоит из трех глав:

«Анализ предметной области», «Модели прогнозирования» и «Программная реализация».

В первой главе дипломной работы представлено основные понятия временных рядов, описывается типы временных рядов, почему мы проводим анализ временных рядов. Во второй главе приводятся модели прогнозирования временных рядов, анализ и выбор модели, обоснование выбора архитектуры и языка программирования. Третья глава посвящена созданию программы для прогнозирования временных рядов с применением более подходящего метода и анализа временных рядов.

# Список сокращений

**ARIMA** Авторегрессионное интегрированное скользящее среднее

**ARMA** Авторегрессивная скользящая средняя

**AR** Авторегрессивный

**БПФ** Быстрое преобразование Фурье

**GRU** Закрытая периодическая установка

**LSTM** Долгосрочная кратковременная память

**MAPE** Средняя абсолютная ошибка в процентах

**МА** Скользящая средняя

**NN** Нейронная сеть

**RNN** Рекуррентная нейронная сеть

**SAPTF SAP** Прогноз временных рядов

**SARIMA** Сезонная авторегрессивная интегрированная скользящая средняя

**SMAPE** Процентная ошибка симметричного среднего значения

**SVR** Одновекторная регрессия

**P2** Корреляция в квадрате Пирсона

**RMSE** Среднеквадратическая ошибка

# Анализ предметной области

## Что такое временной ряд

Каждый последовательный набор точек данных составляет временной ряд. Каждый временной ряд имеет два основных компонента: время и значение, присвоенное соответствующему временному шагу. Одним из наиболее распространенных примеров временных рядов является дневная цена закрытия акций, на этом графике ниже мы представляем цену закрытия акций SAP в период с 2007 по 2019 годы (рисунок 1.1).

****

Рисунок 1.1 - Цена закрытия акций SAP в период с 2007 по 2019 год

Временной ряд, который регистрирует измерения одного явления/переменной, называется одномерным, тогда как измерения нескольких явлений/переменных называются многомерными. В этом исследовании мы сосредоточимся на моделях прогнозирования, которые имеют дело с одномерными временными рядами. Другой способ различать временные ряды — это способ их записи, поскольку у нас есть дискретные временные ряды и непрерывные временные ряды [6].

Дискретные временные ряды измеряются (выбираются) через равные промежутки времени (ежемесячно, ежеквартально, ежегодно), эти временные ряды очень распространены (отслеживание уровня безработицы, который обычно измеряется каждый квартал). Что касается непрерывных временных рядов, то они имеют тенденцию быть более плотными (поскольку они записываются непрерывно) и изменяются намного быстрее, чем дискретные временные ряды, примером таких временных рядов является ЭЭГ (электроэнцефалограмма) или измерения датчиков. На практике все временные ряды дискретны, но отличаются степенью детализации и зависимостью от прошлых значений.

## Типы временных рядов

Существует множество критериев классификации временных рядов, в этой части мы сосредоточимся на зависимости от времени и стационарности.

Зависимость от времени относится к влиянию прошлых значений на вновь наблюдаемые значения записанной переменной/явления, мы можем разделить временные ряды на две категории: временные ряды с долгой памятью и временные ряды с короткой памятью. Временные ряды с длинной памятью — это те, которые имеют автокорреляционную функцию, которая медленно убывает, эти временные ряды описывают процесс, который имеет медленно изменяющееся поведение. Временные ряды с длинной памятью обычно можно найти в метеорологических, геологических данных; Примером временных рядов с долгой памятью является эволюция средней температуры на планетах. Краткосрочные временные ряды описывают процесс с быстрым оборотом и имеют функцию автокорреляции, которая быстро уменьшается по мере того, как чем дальше мы находимся от настоящего, тем менее полезной для будущего является мера. Типичным примером таких временных рядов являются финансовые данные, такие как цены на акции.

Для второго критерия классификации: стационарность, мы имеем стационарные и нестационарные временные ряды. Стационарные временные ряды — это процессы, обладающие статистическими свойствами (среднее значение, дисперсия), не зависящими от времени. Временные ряды, которые не подходят под вышеуказанное описание, называются нестационарными, эти временные ряды очень распространены в финансах и розничном секторе. Эти типы временных рядов трудно предсказать без какой-либо предварительной обработки, поэтому мы используем несколько методов для стационаризации этих процессов, чтобы иметь возможность получать более точные прогнозы.

## Почему мы проводим анализ временных рядов

Анализ временных рядов проводится в основном по двум причинам:

Понимание поведения процесса путем изучения его прошлых записей, чтобы иметь возможность моделировать и определять основные параметры, влияющие на временной ряд, и определять его компоненты [8].

Прогнозирование будущих значений ряда с использованием адекватной модели, котораябыли обучены на прошлых ценностях (рисунок 1.2).

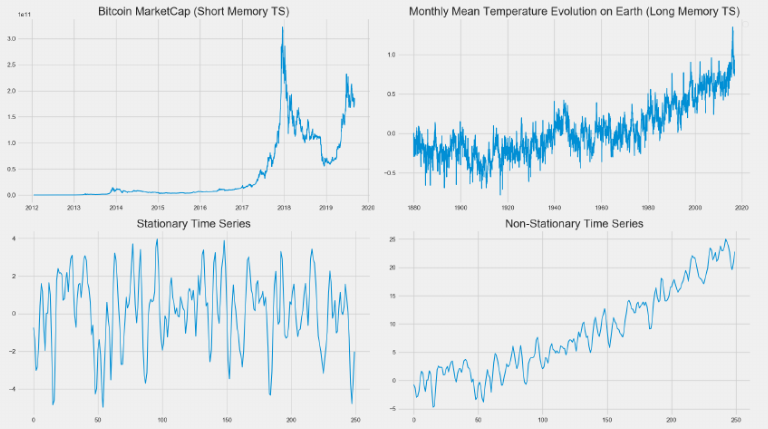
****

Рисунок 1.2 - Пример длинной/короткой памяти, стационарных/нестационарных временных рядов

## **Компоненты временного ряда**

y(t) = T(t) + S(t) + C(t) + ε(t) (1.1)

y(t) = T(t) × S(t) × C(t) × ε(t) (1.2)

y(t) представляет меру, полученную на временном шаге t, T(t) – общий тренд ряда, S(t) описывает сезонный аспект временного ряда, C(t) – циклический компонент наблюдения и ε(t) представляет собой нерегулярные шаблоны в ряду, остатки.

На (рисунке 1.3) представлены продажи розничного магазина в период с 2013 по 2015 год, в этом временном ряду мы можем наблюдать вышеупомянутые компоненты (рисунок 1.4). Тенденция в этом случае немного меняется, но сезонную часть можно вывести из поведения ряда и некоторого здравого смысла [7]. Мы можем наблюдать всплеск продаж в конце каждого года, чего и следовало ожидать, поскольку эти даты соответствуют праздникам, когда большинство розничных магазинов совершают наибольшие продажи.

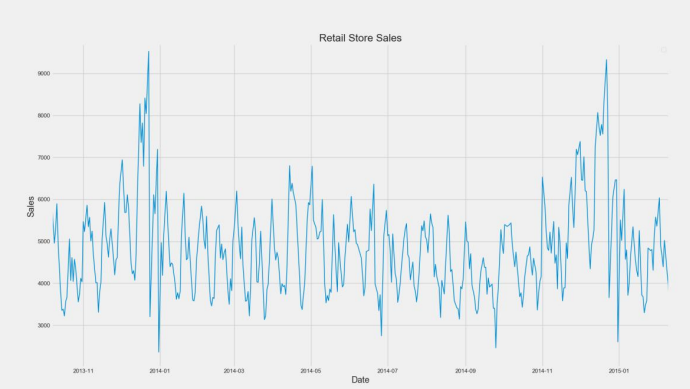
****

Рисунок 1.3 - Продажи розничного магазина

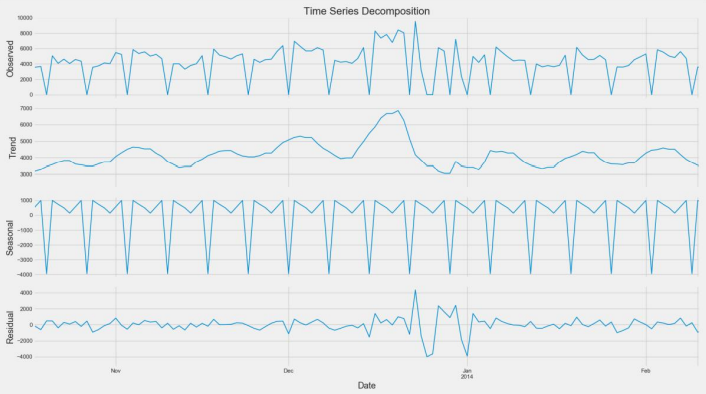
****

Рисунок 1.4 - Декомпозиция продаж розничного магазина на составляющие: Тренд, Сезон и Остаток

# Методы исследования временных рядов

## Как прогнозировать временные ряды

Прогнозирование временных рядов отличается от стандартных задач регрессии, поскольку у нас есть важное ограничение, которое необходимо учитывать: порядок. Хронологический порядок данных затрудняет для оценщика изучение общей модели, которую можно использовать в течение длительного времени, поскольку закономерности могут появляться на некоторое время, а затем исчезать, или может измениться все распределение данных. Прогнозирование цены акций является подходящим примером этой проблемы, поскольку поведение акций может измениться в зависимости от нового регулирования, которое может изменить все распределение данных в дальнейшем.

### Прогнозирование временных рядов

Прогнозирование является основной причиной, по которой мы проводим анализ временных рядов, основная идея состоит в том, чтобы попытаться использовать прошлые наблюдения для прогнозирования будущего. Модель, которая лучше всего описывает данные, позже будет использоваться для прогнозирования будущего на основе прошлых записей. Модель прогнозирования — это функциональное представление, описывающее временной ряд, на основе которого мы будем прогнозировать будущее. Для одномерных временных рядов у нас есть две основные переменные:

* Эндогенные переменные, которые представляют собой прошлые значения ряда;
* Экзогенные переменные (или экспликативные переменные), которые являются внешними факторами, которые могут быть соотнесены со значением временного ряда, эти переменные должны быть детерминированными (например, календарные данные), иначе мы будем использовать другие временные ряды для прогнозирования нашего собственного, что является случай многомерного прогнозирования временных рядов.

В этом примере мы будем использовать только эндогенные переменные.

После определения наилучшей модели, которая может соответствовать данным, мы можем начать прогнозировать будущие значения ряда. Здесь у нас есть два подхода, мы можем прогнозировать на один шаг вперед или на несколько шагов вперед.

### На шаг впереди

yt+1 =f(y0, y1, ..., yt−1, yt) (2.1)

Стратегия на шаг вперед очень прямолинейна, используя прошлые значения ряда, задача будет состоять в том, чтобы предсказать следующее значение сигнала.

### Многоступенчатый

Однако для многоэтапной стратегии все становится немного сложнее, поскольку мы сталкиваемся с двумя вариантами выбора: либо мы прогнозируем все значения ряда до yt+h, либо прогнозируем только последнее значение горизонта yt+h.

Итеративный

yt+1 =f(y0, y1, ..., yt)

yt+2 =f(y0, y1, ..., yt,ˆyt+1)

.

. (2.2)

.

yt+h−1=f(y0, y1, ..., yt,ˆyt+1, .., ˆyt+h−2)

yt+h=f(y0, y1, ..., yt,ˆyt+1, .., ˆyt+h−2,ˆyt+h−1)

Итеративный подход обычно используется авторегрессионными моделями, где каждое предсказание основано на прошлых записях, что затрудняет для модели прямое предсказание значений, которые находятся далеко в будущем (это случай модели AR). , описываемый уравнением (2.2). Слабость этого метода заключается в том, что он распространяет ошибку, допущенную в более ранних прогнозах, на будущее, что может сделать качество долгосрочных прогнозов ненадежным.

Прямой

yt+h=f(y0, y1, ..., yt) (2.3)

Прямой подход использует прошлые значения, собранные до настоящего времени, для непосредственного прогнозирования будущего значения, этот процесс описан выше в уравнении (2.3). Этот подход будет использоваться в наших готовых оценщиках глубокого обучения.

### Перекрестная проверка временных рядов

Традиционный подход к прогнозированию для временных рядов заключается в обучении модели/ оценщика на части данных (обучение набора данных), затем прогнозировании на горизонте фиксированного размера, а затем вычислении производительности оценщика на тестовом наборе данных (рисунок 2.1).

Вышеупомянутая стратегия не подходит для временных рядов, особенно для тех, у которых огромное количество точек данных. Это происходит из-за того, что мы не можем предположить, что распределение ряда не изменится со временем.

С другой стороны, тот факт, что некоторые внешние/экзогенные переменные имеют эффект, который мы не можем обязательно смоделировать/использовать, когда делаем наши прогнозы, делает задачу прогнозирования на относительно широком горизонте ненадежной.

Поэтому, чтобы предотвратить такие проблемы, мы используем подход, который называется: Walk Forward Validation.

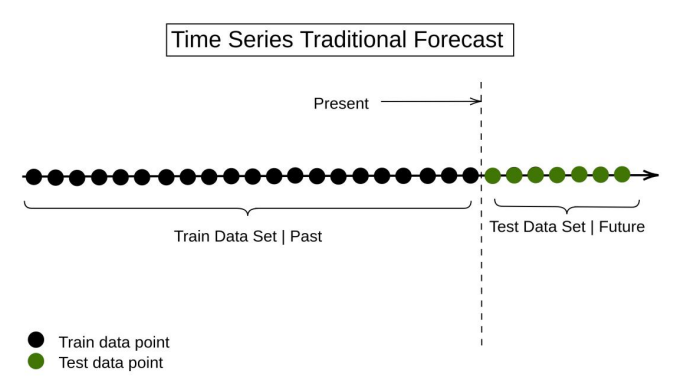
****

Рисунок 2.1 - Традиционный подход к прогнозированию испытаний обучению

Эта стратегия учитывает детализацию/частоту временных рядов при прогнозировании невидимых значений. С деловой/практической точки зрения это современный подход/реализация проблемы прогнозирования временных рядов, поскольку он способен давать более качественные прогнозы (используя максимальный объем информации, который у него есть до «настоящего»). Поскольку мы предполагаем, что у нас могут быть хорошие характеристики от наших оценок до определенного подгоризонта, это можно вывести из детализации набора данных, а затем переобучить модель на том же наборе данных поезда и наблюдениях, включенных в этот подгоризонт что мы уже предсказывали.

Это увеличивает знания оценщиков, поскольку они делают свои прогнозы и противодействуют проблеме, которая широко распространена в традиционном подходе, который адаптирует модель к изменению общей тенденции ряда.

Эта стратегия основана на кампании, в которой использовались длинные временные ряды. Внимательно изучив результат, мы обнаружили, что для такого рода наборов данных даже при коэффициенте проверки 5% мы по-прежнему прогнозируем огромное количество точек за раз (3000 точек для одного случая). По этой причине мы подумали об этом подходе, при котором мы предоставляем такое же количество прогнозируемых точек, но мы предполагаем, что после предполагаемого подгоризонта, основанного на сроке (краткосрочном, среднесрочном или долгосрочном), на который мы прогнозируем, реальные данные доступны для использования. Таким образом, оценщик может использовать эти знания для настройки изученной модели и предоставления более качественных прогнозов для следующего подгоризонта. В нашей реализации мы добавили раздел конфигурации для перекрестной проверки временных рядов [TimeSeriesCV], мы выдвинули 3 термина: краткосрочный, среднесрочный и долгосрочный. Основываясь на этом выборе, мы выберем соответствующий подгоризонт, по которому мы будем делать прогнозы перед повторным обучением (пример: если временной ряд является ежедневным, и мы указываем, что хотим сделать краткосрочный прогноз, тогда наш метод перекрестной проверки выведет подгоризонт 7 дней, таким образом прогнозируя 7 значений перед повторным обучением модели). Кроме того, мы добавили параметр окна, который помогает сохранять одинаковое время прогноза при каждом сгибе. Эта опция сохраняет количество очков, подаваемых в каждом поезде.

Фазовая постоянная (рисунок 2.2). Этот подход используется для имитации реального процесса прогнозирования в бизнесе и для получения представления о надежности/адаптируемости.

С практической/деловой точки зрения эта стратегия имеет большой смысл, поскольку предприятия обычно делают прогнозы на основе небольшого числа точек данных. В нашей реализации мы предоставим 3 варианта расчета для подгоризонта: Краткосрочный, среднесрочный, долгосрочный.

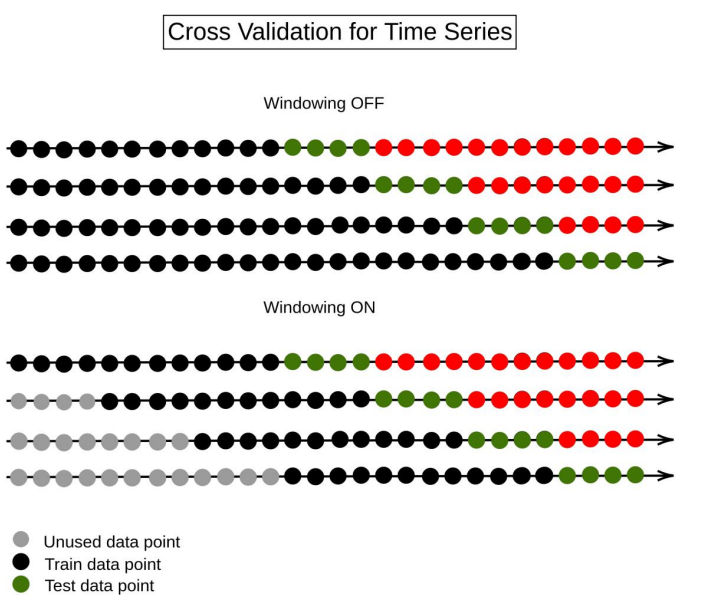
****

Рисунок 2.2- Перекрестная проверка для временных рядов

## Прогнозирование временных рядов SAP (SAPTF)

SAP использует собственную модель прогнозирования временных рядов, которая будет использоваться в качестве эталона при изучении остальных моделей-кандидатов с точки зрения производительности и скорости. Целью данного исследования является сравнение SAPTF с наиболее часто используемыми оценщиками в прогнозировании временных рядов, чтобы выявить их слабые и сильные стороны. SAPTF — это автоматизированная модель, которая может использовать экзогенные переменные для получения более качественных прогнозов. Однако в контексте данного исследования мы будем использовать только эндогенные переменные. SAPTF использует аддитивную модель для декомпозиции временных рядов следующим образом:

y(t) = T(t) + S(t) + F(t) + ε(t) (2.4)

Где y(t) — исходный временной ряд, T(t) — тренд, S(t) — сезонный компонент, F(t) описывает колебания, а ε(t) — остаточный компонент модели.

В отличие от других оценщиков, SAPTF моделирует компоненты последовательно, начиная с тренда, сезонного компонента, затем флуктуаций, которые дают меньшую возможную ошибку. SAPTF проверяет несколько комбинаций гипотез для трендовых и сезонных/ циклических компонентов, чтобы найти ту, которая лучше всего соответствует данным поезда. SAPTF может обрабатывать субдневные периоды (секунды), объяснять модель и указывать важность каждой функции, которую она использует на этапе оценки.

## ARMA и его варианты

### АР(р)

Предположение: временной ряд является стационарным

yt=c+φ1yt−1+φ2yt−2+· · · +φpyt−p (2.5)

p

yt=c+ ∑ φkyt−k (2.6)

k=1

Авторегрессионная модель — это простейшая модель для временных рядов, она исходит из идеи, что любое значение временного ряда можно предсказать на основе прошлых записанных значений. Математически эта формулировка означает, что будущее значение временного ряда yt будет линейной комбинацией прошлых значений с точкой пересечения (2.5, 2.6), где φi∈[1..p] вес каждого наблюдения. Параметр p AR(p) определяет количество лагов, которые будут использоваться при прогнозировании значения yt.

### MA(q)

Предположение: временной ряд является стационарным

yt=µ+εt+θ1εt−1+θ2εt−2+· · · +θqεt−q (2.7)

yt=µ+ εt−k (2.8)

µ=1/q (2.9)

Модель скользящего среднего похожа на авторегрессионную модель, за исключением того, что точка пересечения представляет собой среднее из самых последних значений q, и вместо авторегрессионной модели на наблюдаемых записях мы регрессируем по остаткам (разнице между средние и наблюдаемые значения). Как описано в уравнениях (2.7, 2.8, 2.9), компонент МА представляет собой линейную комбинацию последних q невязок с θi∈[1..q] весом каждого исправления ошибок. Компонента εt−i, i ∈ [1, q], называется случайным скачком, который, как предполагается, следует нормальному распределению с нулевым средним и постоянной дисперсией σ2.

Полезной характеристикой модели MA(1) является то, что она эквивалентна бесконечной модели AR(∞).

### ARMA(p,q)

Предположение: временной ряд является стационарным

yt=c+εt+yt-i+ (2.10)

Модель ARMA(p,q) представляет собой комбинацию ранее упомянутых AR(p) и MA(q), математически модель представляет собой просто сумму этих моделей, как описано в (2.10). Но слабость этих моделей в том, что они предполагают, что временные ряды, над которыми мы работаем, являются стационарными, что не так в большинстве реальных случаев. Таким образом, нам нужно убедиться, что любой временной ряд, смоделированный с помощью этой функции, должен быть стационарным, для этой цели мы используем операцию разности, чтобы стационаризировать ряд. Эта операция обрабатывается моделью ARIMA, которая обобщает модель ARMA для нестационарных временных рядов [2].

### ARIMA(p,d,q)

С помощью этого оператора мы теперь можем переписать модели AR(p), MA(q) и ARMA(p,q) (для простоты мы будем игнорировать постоянный член). Переписывание этих формул с использованием оператора обратного сдвига упростит математическое представление модели ARIMA.

Наиболее важной характеристикой модели ARIMA является то, что она может обрабатывать нестационарные временные ряды благодаря параметру d, который определяет количество разностей, выполняемых для точек данных . Таким образом, модель ARMA будет применяться к дифференцированной переменной, а не к исходной записи.

Несмотря на то, что модель ARIMA больше подходит для обработки большего массива временных рядов (несезонных и нестационарных), ей по-прежнему не хватает способности обрабатывать сезонные данные, поскольку она не моделирует этот компонент ряда в своей математической формулировке[4].

### SARIMA(p, d, q)(P, D, Q)S

Модель SARIMA является обобщением модели ARIMA, представленной в 1970 году Боксом и Дженкинсом, которая предназначена для обработки сезонного компонента периодических временных рядов. В своем обобщенном предложении Бокс и Дженкинс проводят второе дифференцирование сезонной составляющей временного ряда. Таким образом, мы добавляем в модель еще четыре параметра, которые будут (p, d, q)(P, D, Q)S. Модель будет представлена следующим образом:

Модель SARIMA является обобщением всех ранее упомянутых моделей, которые обрабатывают сезонные и нестационарные временные ряды, поскольку мы можем вернуться к любой оценке с правильным выбором параметров (p,d,q)(P,D,Q)S:

SARIMA(p, 0,0)(0,0,0)0 описывает модель AR(p)

SARIMA(0, 0,q)(0,0,0)0 описывает модель MA(q)

SARIMA(p, 0,q)(0,0,0)0 описывает оценку ARMA(p,q)

SARIMA(p, d,q)(0,0,0)0 описывает оценку ARIMA(p, d, q)

Краткий обзор семейства авторегрессионных оценок. Ограничение: одиночная сезонность

Таблица 2.1 - Методология авторегрессионных моделей

|  |  |  |  |
| --- | --- | --- | --- |
|  | Стационарный | Нестационарный | Сезонность |
| AR | ✔ | x | x |
| MA | ✔ | x | x |
| ARMA | ✔ | x | x |
| ARIMA | ✔ | ✔ | x |
| SARIMA | ✔ | ✔ | ✔ |

### Функция автокорреляции и частичной автокорреляции

Определение временной зависимости/корреляции между недавними и прошлыми записями является важным шагом в моделировании и прогнозировании временных рядов, поскольку помогает разработать адекватную модель, которая наилучшим образом описывает имеющиеся у нас записи и дает надежные прогнозы.

Функция автокорреляции (рисунок 2.4) вычисляет корреляции, существующие между временными рядами и их задержками в различные моменты времени. Результатом этой функции является график, называемый коррелограммой (рисунок 2.3). Этот график является важной частью настройки параметров модели ARIMA, поскольку он может выявить наличие периодического паттерна, который может быть скрыт шумом [11]. Мы также используем функцию частичной автокорреляции как еще один способ определения временных зависимостей между недавними и прошлыми значениями, что дает нам частичную корреляцию между записями на разных временных шагах (рисунок 2.3).

Поскольку мы стремимся автоматизировать процесс прогнозирования, мы реализовали вспомогательную функцию, которая будет искать хорошие гиперпараметры, которые обеспечат наилучшие результаты в разумные сроки. Во время этого процесса мы ограничим набор данных, на котором мы будем подбирать наши модели, максимальным значением между 100 точками и последним 0,6-кратным размером набора данных поезда, это делается для ограничения времени, которое каждый кандидат тратит на подбор данных , тем самым ускоряя процесс.

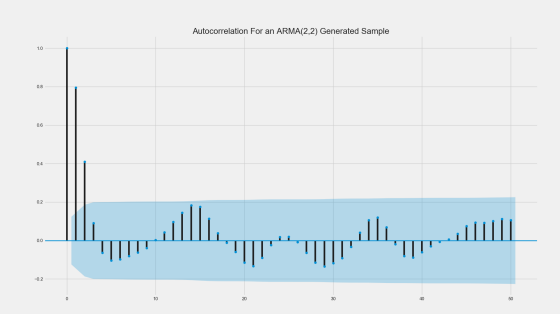


Рисунок 2.3 - Пример автокорреляционной функции для выборки, сгенерированной процессом ARMA(2,2)

Этот выбор гарантирует, что у нас есть минимальное количество точек, по которым можно подогнать модель. Эта вспомогательная функция, которую мы назовем AutoSARIMA, сначала попытается найти сезонный компонент во временном ряду на основе детализации предоставленных данных, используя поле даты. Если это поле не указано, то для извлечения периода используется метод обработки сигналов. Это делается путем извлечения периодограммы ряда. Простая инвертирующая функция вернет период. Как только это значение обнаружено, мы проводим дополнительные тесты, чтобы убедиться, что это значение не является ошибочным, где мы фиксируем порог в 3,5 периода, который должен появиться в наборе данных поезда, чтобы сделать вывод, что это значение действительно является периодом. Как только это будет сделано, и мы узнаем, что будем использовать SARIMA, а не ARIMA, мы запускаем функцию PACF, чтобы вывести наиболее оптимальный порядок AR для сезонной и несезонной части SARIMA. Сезонный компонент AR обнаруживается путем ограничения максимального количества лагов в функции PACF к значению периода, а для несезонного компонента AR p мы используем более высокий параметр максимального запаздывания, который составляет примерно четверть размера данных поезда. Кроме того, мы фиксируем значение порядка MA как для сезонного, так и для несезонного компонента равным 1. Наличие MA более высокого порядка практически не дает улучшения во время проведенных нами экспериментов. Этот подход дополнительно укрепляется путем тестирования 3 значений разностного порядка d и D [12]. Таким образом, вместо выполнения поиска по огромной сетке, который займет несколько часов, чтобы получить какие-либо результаты (что мы сделали сначала, и привели к модели SARIMA, которая может работать часов, прежде чем предоставить прогноз), мы используем мало эвристик и тестируем меньшее количество комбинаций параметров, что ускоряет процесс. С этой вспомогательной функцией среднее время для автоматизированной модели SARIMA составляет около 90 секунд, а средняя производительность теста MAPE составляет около 20%. Модель SARIMA не была повторно реализована с нуля, поскольку мы использовали версию реализации оценщика из библиотеки statsmodels.

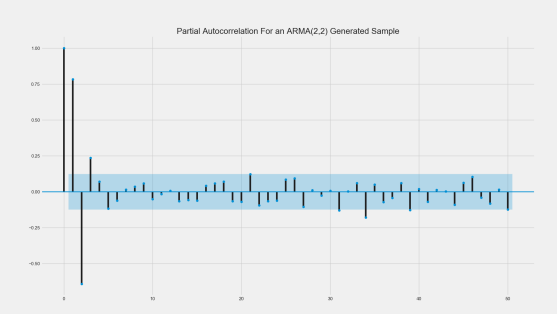


Рисунок 2.4: Пример функции частичной автокорреляции для выборки, сгенерированной процессом ARMA(2,2)

## Прогнозист на основе преобразования Фурье

Прогнозист, основанный на преобразовании Фурье, представляет собой модель, основанную на декомпозиции. Он разлагает сигнал на его строительные блоки, которые позже будут использоваться для прогнозирования и воссоздания сигнала на указанном горизонте H.

### Оценщик БПФ, как он работает

Фундаментальная идея оценки БПФ заключается в том, что мы будем использовать дискретное разложение Фурье для извлечения наиболее важных компонентов ряда, затем воссоздать обобщенную версию наблюдений на основе извлеченных компонентов, а затем применить обратное преобразование Фурье к этим компонентам [5].

1. Детренд временного ряда;
2. Запустите разложение БПФ на временном ряду без тренда;
3. Отфильтруйте низкоамплитудные высокочастотные компоненты;
4. Прогноз по каждому отдельному компоненту;
5. Запустите обратное БПФ для новых компонентов;
6. Добавьте тренд обратно в прогнозы.

Второй шаг модели прогнозирования БПФ используется, чтобы избежать переобучения и избежать переноса шума, который мог попасть в исходные записи, в прогнозы.

### Прогнозирование числа ежедневных рождений в Квебеке набор данных с БПФ

В этом примере мы будем использовать БПФ-оценку количества ежедневных рождений в наборе данных Квебека с удержанием 5%, что эквивалентно горизонту 255. Этот набор данных представляет собой большой набор данных в контексте одномерное прогнозирование временных рядов с 5000 точками. Часть тестового набора изображена на рисунке (рисунок 2.5), где мы видим, что в целом прогнозы не очень далеки от целей.

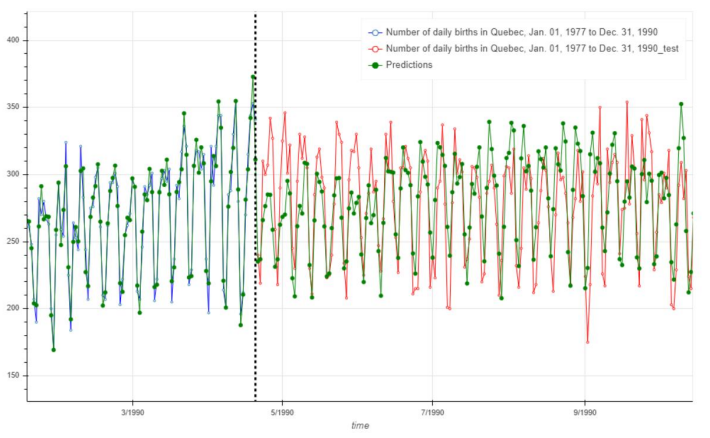
****

Рисунок 2.5: Результаты прогноза количества ежедневных рождений в наборе данных Квебека

Таблица 2.2: Результаты по количеству ежедневных рождений в наборе данных Квебека с разным отклонением [0,025, 0,05, 0,1]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Обучаемый | | | Тестовый | | | Горизонт |
|  | MAPE | RMSE | R2 | MAPE | RMSE | R2 |
| Ho=0.025 | 3.4 | 10.37 | 0.938 | 23.684 | 67.825 | -1.438 | 127 |
| Ho=0.050 | 3.418 | 10.4 | 0.937 | 10.137 | 32.54 | 0.41 | 225 |
| Ho=0.100 | 3.437 | 10.437 | 0.936 | 23.345 | 67.496 | -1.41 | 552 |

Предложенный выше метод не является оценочным, поскольку прогнозирующая часть модели выполняется другим оценщиком временных рядов (по каждой гармонике). Это просто метод, использующий дискретное разложение Фурье для упрощения модели и разделения сложности сигнала на несколько гармоник.

## Facebook Prophet

Prophet — проект с открытым исходным кодом для анализа и прогнозирования временных рядов, разработанный Facebook. Он предоставляет полезный API на Python и R для готового прогнозирования для людей, которым не хватает глубоких знаний в моделировании временных рядов. Prophet предоставляет полуавтоматическую модель прогнозирования с возможностью настройки модели на основе предварительных знаний и опыта пользователя. Этот подход в статье, подробно описывающей модель, используемую в Prophet, называется «аналитик в цикле».

Prophet использует аддитивную модель временных рядов с тремя компонентами:

1. Кусочно-линейная или логистическая функция роста для моделирования трендовой составляющей в ряду;
2. Периодическая составляющая, моделируемая функцией s(t);
3. Праздничный компонент, который должен быть добавлен пользователем вручную для активации. Этот компонент добавляет внешние функции (не только праздники) во временные ряды, которые могут помочь пользователю получить более точные прогнозы и правильно прогнозировать редкие события. Этот компонент предоставляет prophet возможность моделировать «условные» сезонности;
4. Составляющая ошибок, учитывающая ошибки, допущенные при прогнозировании.

### Полезные дополнения к традиционной аддитивной модели регрессии

Prophet использует модифицированную модель обнаружения тренда, в которой используется кусочно-линейный компонент, чтобы иметь возможность корректировать уровень временного ряда при сдвиге наклона тренда, чтобы не упустить локальный паттерн в пользу общего тренда. шаблон в серии и использует последнюю часть тренда в качестве тренда для будущего. Кроме того, тренд использует функцию, подобную логистическому росту, для моделирования непериодического развития сигнала во времени. Эта функция имеет полезный параметр C(t), который ограничивает значение наблюдений максимальным или минимальным значением. Примером полезности этого компонента является прогнозирование рядов в реальном времени, таких как изменение количества пользователей конкретной услуги, это значение не может превышать количество людей на земле; или при прогнозировании цены акции мы точно знаем, что цена акции не может быть ниже 0. Эти небольшие ограничения на модель могут иметь большое значение, чтобы убедиться, что модель имеет смысл и дает объяснимые прогнозы. Инструмент Facebook также обрабатывает мультипликативную сезонность и ежедневную частоту, но эти два параметра должны быть предоставлены модели вручную. Одна слабость prophet заключается в том, что он не может обрабатывать временные ряды без временных меток, поскольку он использует даты для определения частоты и построения модели. Эта слабость устраняется в нашем коде путем возврата среднего значения набора данных поезда в качестве прогноза. Эта проблема является проблемой для этой реализации модели prophet, которая постоянно обеспечивает хорошие прогнозы, поскольку она не может обрабатывать записи без временных меток, которых в нашем случае довольно много.

## Регрессия опорных векторов (SVR)

### Теоретическая формулировка СВР

Обычно модели машины опорных векторов используются для бинарной классификации, поскольку основная идея заключается в том, чтобы найти гиперплоскость, которая может максимально разделить два класса. Однако другой вариант модели, называемый SVR (или SVM для регрессии), полученный из математической формулировки SVM, обобщает область применения модели для регрессии. Тонкое различие между формулировками SVM и SVR заключается в том, что выходные данные теперь являются непрерывными, а не дискретными, и функцией потерь, используемой в задаче квадратичного программирования, ε-интенсивной функцией потерь [3].

В целях наших экспериментов, поскольку мы манипулируем одномерными временными рядами, нам необходимо создать функции, которые будут переданы в модель для прогнозирования. Мы будем использовать скользящее окно размера N, которое будет определяться нашей моделью автоматически исходя из некоторых характеристик временного ряда и размера горизонта прогнозирования, это окно будет подаваться в модель с целевым значением yt+H (Рисунок 3.6). Во всех наших экспериментах мы использовали реализацию SVR от sklearn с линейным ядром и ε = 10−2. Еще одним важным фактором для создания модели SVR для временных рядов является размер окна прокрутки, выбор этого параметра осуществляется так же, как и в моделях глубокого обучения, в следующем разделе мы подробно опишем процесс выбора этого значения.

## Подход к глубокому обучению

### Нейронная сеть

Нейронные сети сейчас используются гораздо чаще при прогнозировании временных рядов, особенно для длинных рядов. Это связано с тем, что НС могут моделировать сложные нелинейные модели и самостоятельно подстраиваться под данные. NN - это универсальные оценщики, используемые для нескольких задач обучения, это свойство делает первостепенной задачей правильную настройку нашей сети.

### Рекуррентная нейронная сеть (RNN)

Внутренняя структура RNN лучше подходит для обработки временных рядов, поскольку она может использовать преимущества порядка в данных. Целью нашей методологии для архитектуры глубокого обучения является предоставление готовой модели RNN, которая автоматически настраивается и настраивается на основе предоставленного набора данных с использованием эвристики, разработанной на этапе исследования, которая обеспечивает наилучшие результаты. Из единиц RNN, представленных в литературе, мы выбрали две единицы, которые составляют слои наших моделей. Эти две единицы являются наиболее распространенными и используемыми: Gated Recurrent Unit и Long Short Term Memory.

Долгосрочная кратковременная память (LSTM)

ft = Ϭg (Wf xt + Uf ht-1 + bf) (2.11)

it = Ϭg (Wi xt + Ui ht-1 + bi)

ot = Ϭg (Wo xt + Uo ht-1 + bo)

Ячейка LSTM имеет два компонента, которые обрабатывают ее состояние: скрытое состояние ht и внутреннее состояние ct. Скрытое состояние соответствует кратковременной памяти, тогда как состояние ячейки соответствует долговременной памяти. Одна ячейка LSTM состоит из ячейки и трех ворот (ввод, забвение и вывод). Ворота используют сигмовидную функцию активации σg, а вход и состояние ячейки обычно преобразуются с помощью tanh σh. Механизм стробирования может хранить информацию в течение длительного времени, это устройство поддерживает карусель постоянных ошибок (CEC), который решает проблему исчезающих и взрывающихся градиентов.

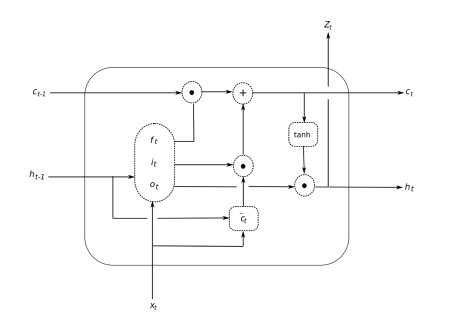


Рисунок 2.7 - Модуль LSTM

Добавление CEC к блоку LSTM позволяет ему изучать долгосрочные модели (рисунок 2.7).В уравнении (2.11) ht обозначает вектор скрытого состояния ячейки на временном шаге t, который фиксирует информацию, которую необходимо распространять дальше в будущем. ct и ˜ct — состояние ячейки и состояние ячейки-кандидата на временном шаге t. Uf , Ui , Uo — весовые матрицы вентилей забывания, ввода, вывода и состояния ячейки соответственно. Точно так же Wf , Wi , Wo и bf , bi , bo, bc являются весовыми матрицами и векторами смещения текущего входа. В выражении ct он и ft определяют, какую часть предыдущей информации сохранить в текущем состоянии ячейки и какую часть текущего контекста передать в будущее. Если ворота забывания установлены на 1, то предыдущее состояние ячейки должно быть сохранено, а если они установлены на 0, то ничего из текущего состояния не должно сохраняться.

Закрытая рекуррентная установка (GRU)

GRU - это еще один вариант рекуррентной единицы, который относительно проще, чем ячейка LSTM, что делает его быстрее в вычислениях. Блок ГРУ имеет только два шлюзовых механизма: ворота обновления и сброса. Шлюз обновления сочетает в себе роль шлюза забывания и входа модуля LSTM. Кроме того, в отличие от LSTM, GRU имеет только один компонент состояния ht. Ворота сброса решают, какую часть предыдущей информации о предыдущем скрытом состоянии внести в потенциальное текущее состояние. Затем для вычисления текущего скрытого состояния значение шлюза обновления будет использоваться для вычисления средневзвешенного значения текущего скрытого состояния-кандидата и предыдущего скрытого состояния (рисунок 2.8).

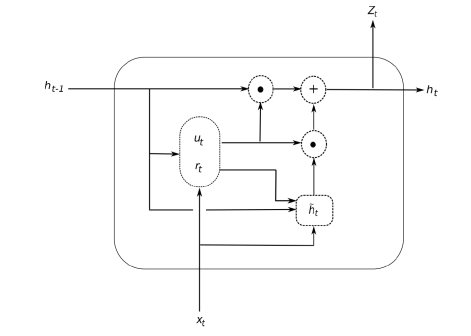


Рисунок 2.8 - Блок GRU

### Архитектура моделей RNN

В наших экспериментах мы выбрали архитектуру Sequence to Sequence с плотным слоем, которая использует движущееся окно в качестве входных данных и yt+h в качестве цели. Мы используем эту архитектуру для прогнозирования значений до горизонта h (прогнозируем h точек), этот выбор мотивирован тем, что мы хотим изучить прогностическую способность модели до горизонта h без какой-либо дополнительной ошибки, которая может распространяться тот факт, что мы повторно вводим предсказанные значения в качестве входных данных для следующего предсказания.

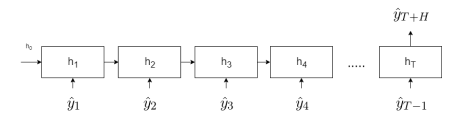


Рисунок 2.9 - Последовательность к последовательности с архитектурой с плотным уровнем

На этапе сравнительного анализа мы экспериментировали с четырьмя основными моделями. Все они автоматически настраиваются на основе эвристики. После того, как модель выбрана, мы начинаем с настройки архитектуры, чтобы она лучше всего обрабатывала данный набор данных, начиная с выбора размера движущегося окна, количества единиц RNN на слой и количества нейронов в плотных слоях. Как только модель правильно настроена, мы обучаем модель, а затем прогнозируем до фиксированного горизонта h.

### Вейвлеты + нейронные сети

В этом разделе исследования мы добавили специальную архитектуру глубокого обучения, которая использует различный уровень вейвлет-декомпозиции, чтобы обогатить функции, которые мы можем извлечь из временного ряда. Мы извлекаем максимум 4 уровня декомпозиции, они используются для извлечения более глобальных шаблонов, которые мы можем упустить из виду при использовании небольших движущихся окон.

### Обзор архитектуры

Основной принцип этой архитектуры прост: мы разлагаем временные ряды на 4 различных уровня субдискретизации, чтобы извлечь глобальные закономерности, которые может быть трудно воспринять сетью (аналогично свертке при работе с изображениями). После разложения временного ряда на аппроксимацию компонентов уровня 4 A4 и 4 деталей D1, D2, D3, D4 мы передаем эти входные данные в виде движущегося окна в их соответствующую рекуррентную нейронную сеть для извлечения полезных функций. В то же время в модель также передаются статистические характеристики каждого движущегося окна (записи поезда временного ряда) и исходные записи в виде движущихся окон. Затем, после того, как все функции были извлечены (оранжевая область), мы объединяем выходные данные всех кодировщиков и передаем их в модель RNN (простая LSTM + Dense Layer), чтобы предоставить прогноз. В этой реализации мы использовали пять статистических признаков: среднее значение, среднее среднее отклонение, дисперсию, эксцесс, асимметрию. Для всех наших намеченных целей мы использовали библиотеку pywt v:1.0.3 для всех наших операций разложения и преобразования вейвлетов.

## Анализ и выбор модели

В этом разделе исследования мы представим результаты кампаний, проведенных с использованием вышеупомянутой техники и моделей. Все модели были запущены на 189 наборах данных с различными характеристиками и объемом (финансовые, деловые, метеорологические, макроэкономические и т. д.). Все протестированные оценщики автоматизированы и не настраивались вручную во время любого из этих экспериментов. Здесь мы представим производительность каждой модели на нескольких наборах данных, а затем общую производительность моделей на всех 189 наборах данных с точки зрения производительности (метрика MAPE) и с точки зрения времени прогнозирования. Последняя метрика важна, потому что нам нужно предоставить хороший прогноз, но также и за разумное время, иначе модель бесполезна, даже если она дает наилучшие результаты, но для ее вычисления требуются часы [10]. Все графики, относящиеся к следующему опыту, будут добавлены в раздел приложений.

Ниже мы найдем результаты, представленные в таблицах, где каждая таблица представляет собой опыт, в котором мы используем каждую из предложенных моделей для наборов данных с использованием традиционного метода прогнозирования. Ho относится к удержанной части данных, по которым будет производиться прогноз (размер удержанной части — это количество прогнозов вне выборки, выполненных моделью, мы называем это значение горизонтом). В этом разделе все значения показателя MAPE выражены в процентах.

### Эксперименты и результаты

Таблица 2.3 - Результаты оценки по набору данных о продажах колы с разным отставанием [0,1, 0,15, 0,2]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Источник | Обучение | | |  | Тест |  | Горизонт |
|  | MAPE | RMSE | R2 | MAPE | RMSE | R2 |
| Ho=0.10 | SAPTF | 21.267 | 137.658 | 0.8 | 39.495 | 256.818 | -0.825 | 3 |
| Prophet | 5.45 | 27.156 | 0.99 | 13.696 | 94.86 | 0.75 |
| SARIMA | 12.415 | 90.403 | 0.914 | 4.415 | 28 | 0.978 |
| SVR | 32.756 | 237.656 | 0.415 | 29.23 | 260.32 | -0.8757 |
| GRU | 3.42 | 254.525 | 0.329 | 41.09 | 312.51 | -1.7 |
| LSTM | 27.59 | 269.419 | 0.248 | 41.062 | 317.689 | -1.79 |
| Ho=0.15 | SAPTF | 27.826 | 176.319 | 0.528 | 33.609 | 418.41 | -0.606 | 5 |
| Prophet | 4.837 | 28.58 | 0.987 | 12.34 | 127.24 | 0.85 |
| SARIMA | 9.935 | 68.096 | 0.932 | 0.9 | 7.42 | 0.999 |
| SVR | 28.42 | 212.016 | 0.349 | 39.279 | 494.24 | -1.24 |
| GRU | 23.508 | 186.82 | 0.494 | 41.81 | 524.528 | -1.52 |
| LSTM | 22.378 | 190.33 | 0.475 | 45.58 | 553.775 | -1.81 |
| Ho=0.20 | SAPTF | 28.216 | 158.34 | 0.582 | 34.03 | 412.52 | -0.935 | 7 |
| Prophet | 2.43 | 10.936 | 0.998 | 18.02 | 166.16 | 0.685 |
| SARIMA | 14.014 | 97.034 | 0.844 | 8.559 | 88.662 | 0.91057 |
| SVR | 17.45 | 138.48 | 0.706 | 17.147 | 214.6 | 0.476 |
| GRU | 17.3 | 110.916 | 0.811 | 24.589 | 267.42 | 0.186 |
| LSTM | 40.577 | 263.398 | -0.06 | 38.13 | 492.83 | -1.763 |

Таблица 2.4 - Результаты оценки количества ежедневных рождений в наборе данных Квебека с разным отклонением [0,1, 0,15, 0,2]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pipelines |  | Train |  |  | Test |  | Horizon |
|  | MAPE | RMSE | R2 | MAPE | RMSE | R2 |
| Ho=0.10 | SAPTF | 6.33 | 19.27 | 0.797 | 8.397 | 26.23 | 0.61 | 182 |
| Prophet | 6.35 | 19.2 | 0.8 | 6.848 | 22.627 | 0.709 |
| SARIMA | 6.76 | 20.58 | 0.768 | 7.775 | 25.98 | 0.617 |
| SVR | 8.83 | 27.387 | 0.6 | 10.72 | 35.85 | 0.27 |
| GRU | 8.7 | 25.975 | 0.64 | 9.647 | 31.288 | 0.445 |
| LSTM | 8.92 | 27.119 | 0.608 | 12.15 | 38.59 | 0.156 |
| Ho=0.15 | SAPTF | 6.4 | 19.45 | 0.779 | 8.667 | 27.949 | 0.59 | 273 |
| Prophet | 6.33 | 19.016 | 0.789 | 6.7 | 22.58 | 0.73 |
| SARIMA | 6.8 | 20.548 | 0.753 | 9.55 | 30.418 | 0.5166 |
| SVR | 6.32 | 19.92 | 0.768 | 7.418 | 24.84 | 0.677 |
| GRU | 6.867 | 20.73 | 0.749 | 7.23 | 25.457 | 0.66 |
| LSTM | 10.63 | 30.857 | 0.44 | 12.955 | 45.477 | -0.08 |
| Ho=0.20 | SAPTF | 6.395 | 19.48 | 0.775 | 6.867 | 23.17 | 0.71 | 364 |
| Prophet | 6.35 | 19.01 | 0.786 | 6.508 | 21.816 | 0.745 |
| SARIMA | 6.83 | 20.543 | 0.75 | 10.949 | 36.627 | 0.28 |
| SVR | 0.28 | 21.39 | 0.73 | 7.884 | 27.14 | 0.606 |
| GRU | 6.9 | 20.72 | 0.748 | 7.25 | 24.866 | 24.866 |
| LSTM | 7.074 | 21.23 | 0.736 | 0.736 | 24.766 | 0.67 |

Таблица 2.5 - Средняя производительность кандидатов на наборе данных 189, метрика производительности — MAPE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Средняя  производ обучения | Тест средней производ | Средняя разница произв  Тестовое обучение | Рейтинг (на основе результатов теста) |
| SAPTF | 12.263 | 16.955 | 4.691 | 1 |
| Prophet | 18.818 | 22.277 | 3.458 | 4 |
| SARIMA | 15.519 | 22.982 | 7.463 | 5 |
| SVR | 13.449 | 21.158 | 7.708 | 2 |
| GRU | 16.623 | 21.914 | 3 | 3 |
| LSTM | 17.41 | 23.187 | 5.776 | 6 |

Таблица 2.6 - Средняя относительная производительность по времени прогноза Время прогноза-кандидата/Время прогноза SAPTF для небольших наборов данных

|  |  |
| --- | --- |
|  | Среднее время прогноза |
| Prophet | 0.86 |
| SARIMA | 23.62 |
| SVR | 0.469 |
| GRU | 6.17 |
| LSTM | 4.875 |

## Обоснование выбора среды разработки и языка программирования

Язык программирования питон дает большой выбор библиотек для работы с временными рядами, поэтому был выбран этот язык программирования, библиотеки будут перечислены далее. Создание модели временных рядов в Python позволяет получить более сложные данные и включает все элементы данных, которые могут быть важны. Это также позволяет вносить коррективы в различные измерения, настраивая модель, чтобы сделать ее потенциально более точной.

Sktime

Это библиотека Python с открытым исходным кодом, предназначенная исключительно для анализа временных рядов. Он предоставляет расширение API scikit-learn для решений временных рядов и содержит все необходимые алгоритмы и инструменты, необходимые для эффективного решения проблем регрессии, прогнозирования и категоризации временных рядов.

Darts

Darts — это еще одна библиотека Python для временных рядов, которая вошла в список 10 лучших библиотек Python для анализа временных рядов в 2022 году. Darts, разработанный Unit8, широко известен тем, что позволяет легко манипулировать временными рядами и прогнозировать их. Он может довольно хорошо обрабатывать большие данные и поддерживает как одномерный, так и многомерный анализ и модели временных рядов

Pyflux

Еще одна библиотека временных рядов Python с открытым исходным кодом, заслуживающая упоминания, — это Pyflux. Подход, используемый этой библиотекой, в основном полезен для таких задач, как предсказание. Здесь пользователи могут построить стохастический шаблон, в котором данные и скрытые значения обрабатываются как случайные опасности с использованием совместной вероятности.

Arrow

Arrow — это не что иное, как удобный для человека подход к созданию, манипулированию, форматированию и преобразованию дат, времени и временных меток. Эта библиотека Python реализует и обновляет тип DateTime, заполняя пробелы в функциональности, а также предоставляя API интеллектуального модуля.

Pastas

Эта библиотека временных рядов Python с открытым исходным кодом предназначена для обработки, моделирования и анализа гидрогеологических моделей временных рядов. Он состоит из встроенных инструментов для статистического анализа, визуализации и оптимизации моделей временных рядов.

Jupyter notebook обладают тремя особенно важными преимуществами:

Они отлично подходят для демонстрации вашей работы. Вы можете видеть как код, так и результаты.

Легко использовать работу других людей в качестве отправной точки. Вы можете запустить ячейку за ячейкой, чтобы лучше понять, что делает код.

Очень легко размещать серверную часть, что полезно в целях безопасности. Многие данные являются конфиденциальными и должны быть защищены, и одним из шагов к этому является отсутствие хранения данных на локальных машинах. Установка Jupyter Notebook на стороне сервера дает вам это бесплатно.

При прототипировании подход ноутбуков Jupyter на основе ячеек великолепен. Но вы быстро заканчиваете тем, что программируете несколько шагов — вместо того, чтобы смотреть на объектно-ориентированное программирование.

# Программная реализация

## Описание работы моделей

В этом разделе мы представим описание всех моделей который мы будем использовать для реализации нашей программы. Модель с точки зрения времени прогнозирования — это автоматизированные SARIMA, поскольку он выполняют поиск гиперпараметров, который требует значительных вычислительных ресурсов. В итоге я решил использовать авторегрессионные модели, так как буду работать с простыми данными временных рядов. Я опишу процессы прогнозирования таких моделей временных рядов как AR, MA, ARIMA, SARIMA.

Для работы модели AR и MA мы импортируем библиотеки:

Pandas — это библиотека Python для обработки и анализа структурированных данных.

Statsmodels— содержит классы моделей и функции, полезные для анализа временных рядов. Базовые модели включают одномерные авторегрессионные модели (AR), векторные авторегрессионные модели (VAR) и одномерные авторегрессионные модели скользящего среднего (ARMA).

Matplotlib — это обширная библиотека для создания статических, анимированных и интерактивных визуализаций на Python.

Загрузим данные CSV, которые представляют собой ежедневное общее количество рождений женщин в городе Сакраменто.

Для работы модели ARIMA мы импортируем библиотеки warnings, matplotlib, numpy, pandas, statsmodels, pmdarima, sklearn, statsmodels.

Warnings **-** предупреждающие сообщения обычно выдаются в ситуациях, когда полезно предупредить пользователя о каком-либо состоянии в программе, где это условие (обычно) не требует возбуждения исключения и завершения программы.

Numpy - это библиотека Python, используемая для работы с массивами. Он также имеет функции для работы в области линейной алгебры, преобразования Фурье и матриц.

pmdarima - (первоначально пирамида-арима, для анаграммы «py» + «arima») - это статистическая библиотека, предназначенная для заполнения пробела в возможностях анализа временных рядов Python, переносит любимый auto.arima из R на Python, что еще раз доказывает, почему вам не нужен R для науки о данных. pmdarima на 100% состоит из Python + Cython и не использует какой-либо код R, но реализован в виде мощного, но простого в использовании набора функций и классов.

sklearn - бесплатная библиотека машинного обучения для Python. Он включает в себя различные алгоритмы, такие как машина опорных векторов, случайные леса и k-neighbours, а также поддерживает числовые и научные библиотеки Python, такие как NumPy и SciPy и загрузим данные CSV, которые представляют собой данные об акциях Facebook.

Для работы модели SARIMA мы импортируем те же библиотеки warnings, matplotlib, numpy, pandas, statsmodels, pmdarima, sklearn, statsmodels эти библиотеки были описаны выше. Модель SARIMA является обобщением модели ARIMA. В своем обобщенном предложении Бокс и Дженкинс проводят вторую дифференциацию сезонной составляющей временного ряда. Таким образом, мы добавляем в модель еще четыре параметра, которые будут (p, d, q)(P, D, Q) s. Так же загрузим данные CSV, которые представляют собой данные об акциях Facebook.

## ****Реализация моделей прогнозирования****

### Авторегрессионные модели

Модель авторегрессии (AR) предсказывает будущее поведение на основе прошлых данных. Это когда данные коррелируют с последовательной последовательностью временного ряда и значениями до и после последовательности. Авторегрессионная модель использует только данные о прошлом поведении для прогнозирования значения.

**Модели AR используют прошлые значения для прогнозирования, как показано здесь:**

**Ŷt = μ + ϕ1Yt-1 +** **ϕ2Yt-2 ... +** **ϕpYt-p** **+ ɛt (2.12)**

где ɛt — белый шум. Эта модель известна как модель AR(p), где p — порядок авторегрессионной модели. Модель AR проста в использовании для обработки широкого спектра моделей временных рядов. Рисунок 3.1-3.2 иллюстрирует поведение AR (1) и AR (2) (рисунок 3.1), ) (рисунок 3.2).

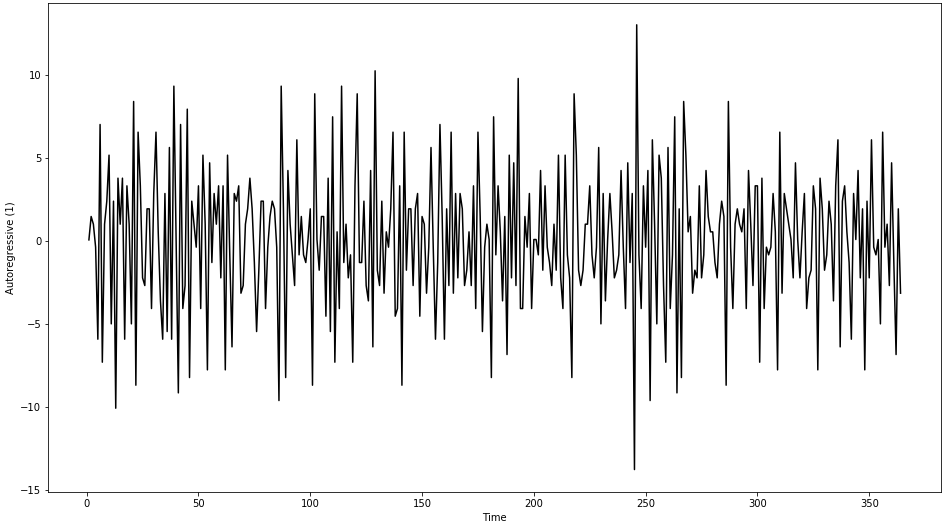


Рисунок 3.1 - Модель авторегрессии AR (1)

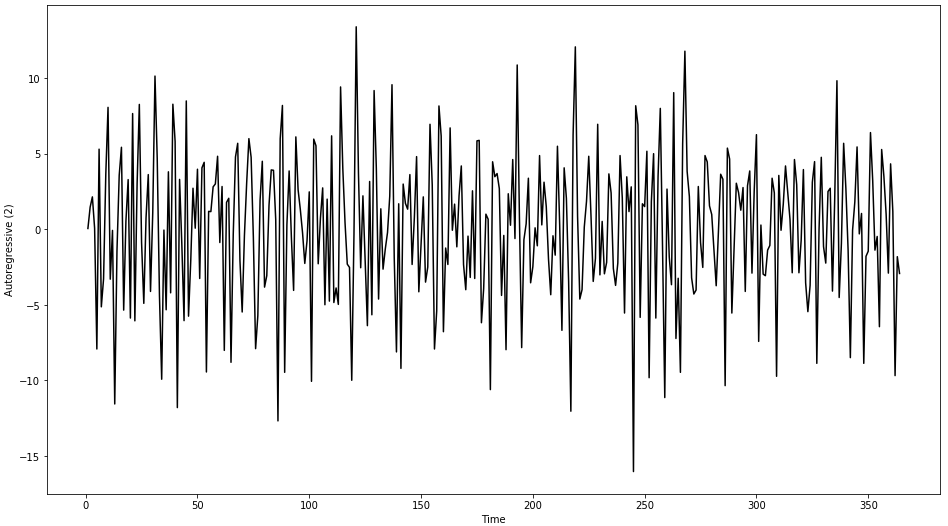
****

Рисунок 3.2 - Модель авторегрессии AR (2)

Модели авторегрессии ограничены стационарными данными, и требуются некоторые параметры, как определено здесь:

* **AR (1) model:** -1 < ϕ**1** < 1
* **AR (2) model:** -1 < ϕ**1** < 1, ϕ**1** + ϕ**2** < 1, ϕ**1** - ϕ**2** < 1

Скользящее среднее

Скользящее среднее (MA) — это метод получения всех трендов во временном ряду. Это среднее значение любой подкатегории чисел. Он используется для долгосрочного прогнозирования тенденций. По сути, скользящее среднее прогнозирует будущие точки, используя среднее значение нескольких прошлых точек данных [9].

Модель скользящего среднего использует прошлые ошибки прогноза:

Ŷt = μ**0**+ εt – ω**1** ε**t-2** - ω**1** ε**t-2 …** ω**q** ε**t-q**

где ɛt — белый шум. Эта модель известна как модель MA(q), где q упорядочено для модели скользящего среднего. Модель MA проста в использовании для работы с широким спектром моделей временных рядов. Рисунок 3.3-3.4 иллюстрирует поведение MA (1) и MA (2) (рисунок 3.3), ) (рисунок 3.4).

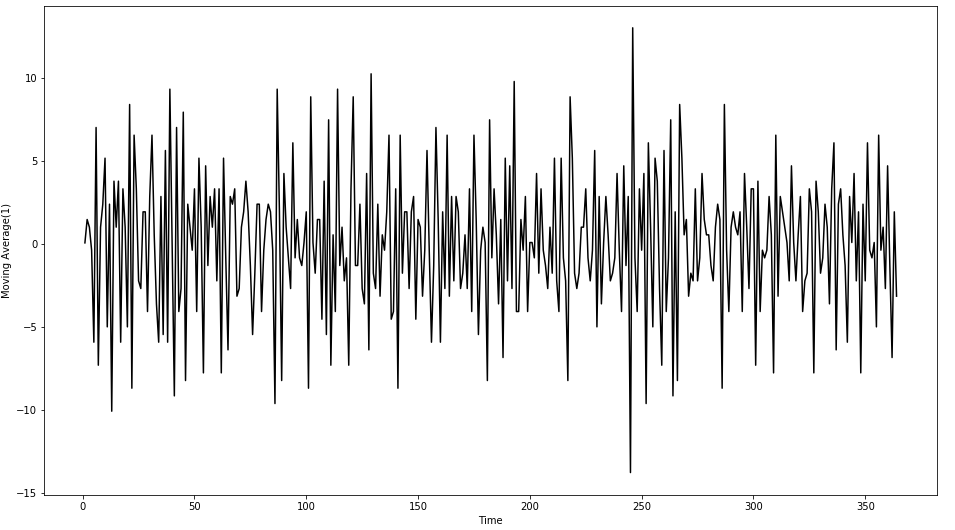


Рисунок 3.3 - Модель скользящее среднее MA (1)

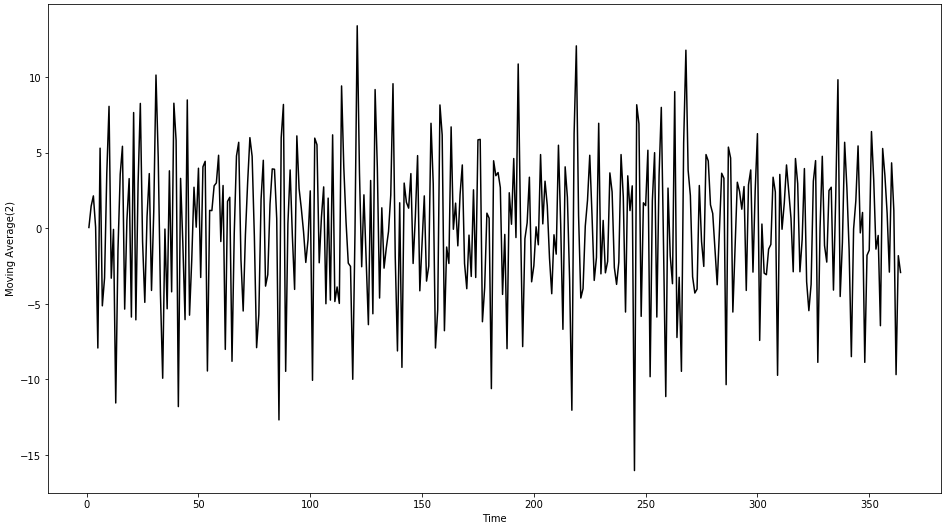


Рисунок 3.4 - Модель скользящее среднее MA (2)

В предыдущем разделе вы узнали об AR и MA, а также об интерпретации AR, I и MA. Теперь давайте реализуем это на наборе данных временных рядов.

Давайте взглянем на данные об акциях Facebook (рисунок 3.5).



Рисунок 3.5 - Данные об акциях Facebook

Выполним некоторый базовый исследовательский анализ данных, используя линейную, гистограммную и ядерную оценку плотности цены закрытия акций (рисунок 3.6). Исследовательский анализ данных — это процесс использования графического представления для обнаружения и исследования закономерностей в данных (рисунок 3.7).

Оценка плотности ядра (KDE) — это непараметрический способ оценки функции плотности вероятности (PDF) случайной величины.

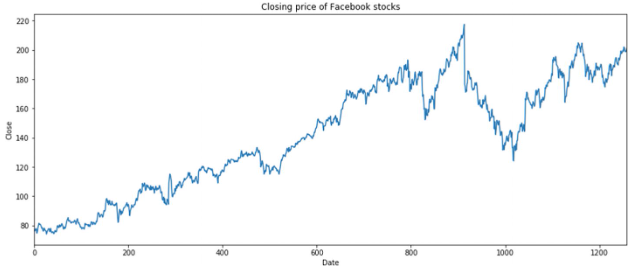


Рисунок 3.6 - Цена закрытия акций Facebook

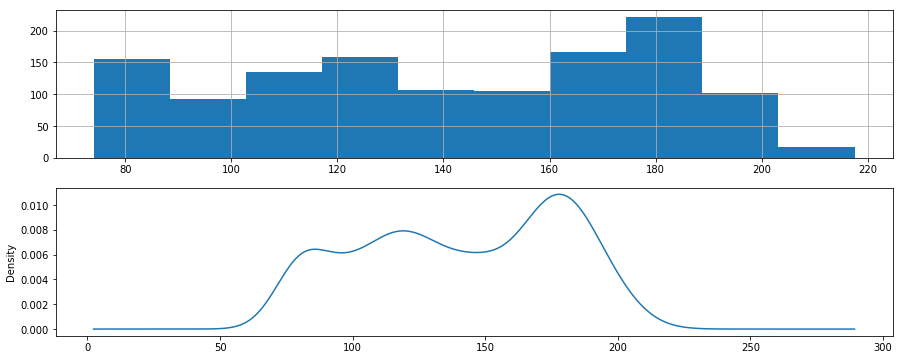


Рисунок 3.7 – Линейные и гистограммные данные об акциях Facebook

Мы видим, что функция Close нестационарна, и auto-arima обрабатывает это внутри себя (рисунок 3.8):

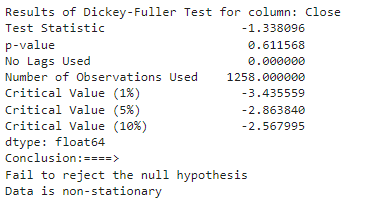


Рисунок 3.8 – Результат теста Дикки-Фуллера

Модуль pmdarima поможет нам определить p, d и q, не глядя на график (рисунок 3.9).

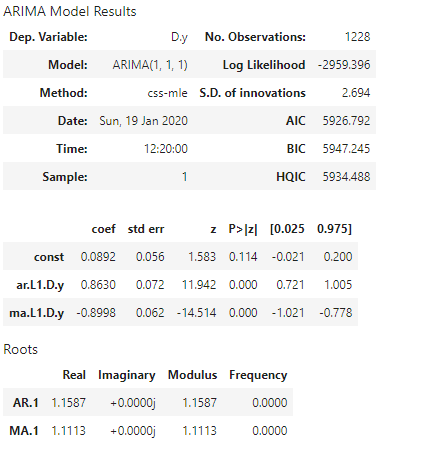


Рисунок 3.9 – Результат модели ARIMA

Спрогнозируйте как результаты, так и достоверность на следующие 30 дней и сохраните их в DataFrame, как показано здесь (рисунок 3.10):

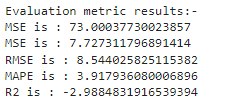


Рисунок 3.10 – Прогноз и достоверность на следующие 30 дней

Построим результаты с доверительными границами, как показано здесь (рисунок 3.11):

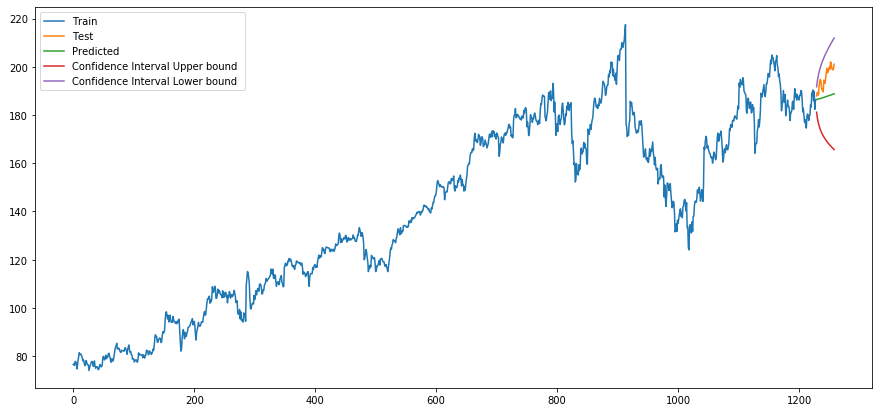


Рисунок 3.11 – Результаты с доверительными границами

Вот диагностический график (рисунок 3.12):

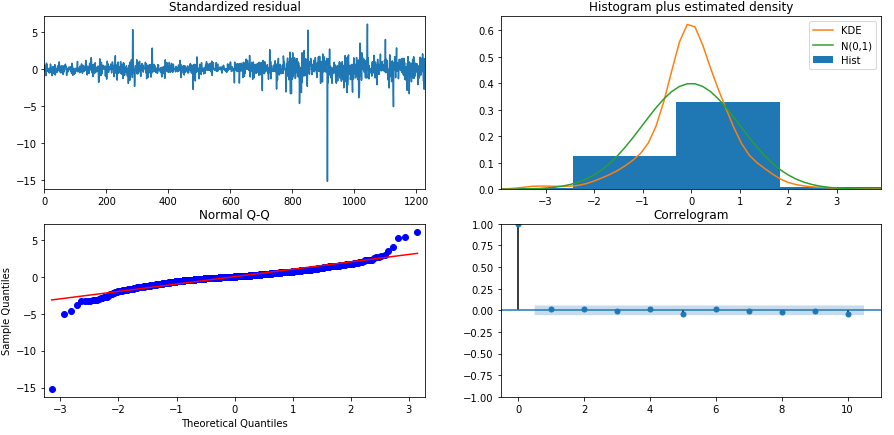


Рисунок 3.12 – Диагностический график

Реализация SARIMA.

В предыдущем разделе вы узнали о математике высокого уровня, лежащей в основе SARIMA, где вы можете выполнять моделирование данных для сезонных компонентов. Теперь давайте реализуем это на наборе данных временных рядов.

Давайте взглянем на данные (рисунок 3.13).



Рисунок 3.13 – Данные об акциях Facebook

Определим функцию оценки временных рядов, как показано здесь (рисунок 3.14):

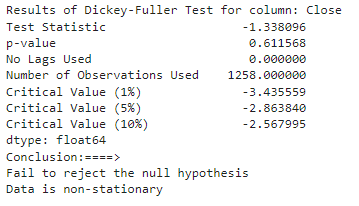


Рисунок 3.14 – Результат оценки временных рядов

Давайте настроим и запустим сезонную ариму для параметров, заданных в цикле for, и проверим оптимальное количество периодов в каждом сезоне, подходящее для нашего набора данных.

Вот место для поиска:

p → от 1 до 7.

q → от 1 до 7.

d → None означает найти оптимальное значение.

P → от 1 до 7.

Q → от 1 до 7.

D → None означает найти оптимальное значение.

m относится к количеству периодов в каждом сезоне.

7 → Ежедневно

12 → Ежемесячно

52 → Еженедельно

4 → Ежеквартально

1 → Годовой (несезонный)

Подгонка SARIMA по сезонной стоимости m = 1 (рисунок 3.15).

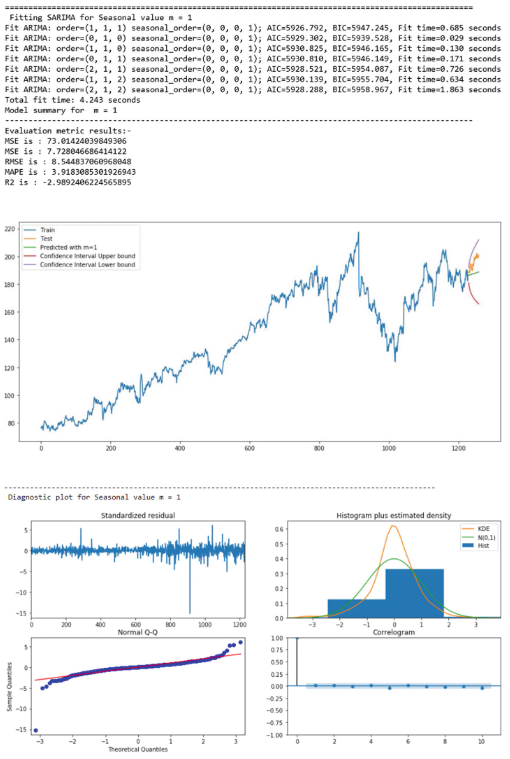


Рисунок 3.15 – SARIMA по сезонной стоимости m = 1

Подгонка SARIMA по сезонной стоимости m = 4 (рисунок 3.16).

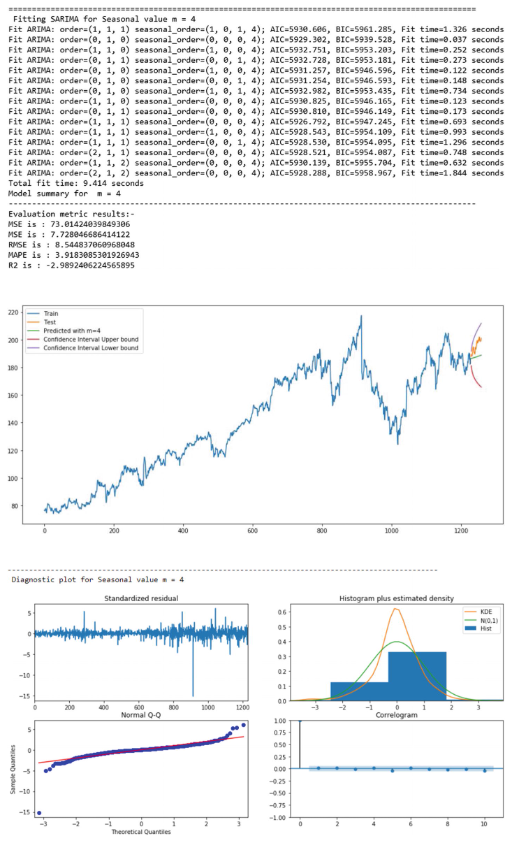


Рисунок 3.16 – SARIMA по сезонной стоимости m = 4

Подгонка SARIMA по сезонной стоимости m = 7 (рисунок 3.17).

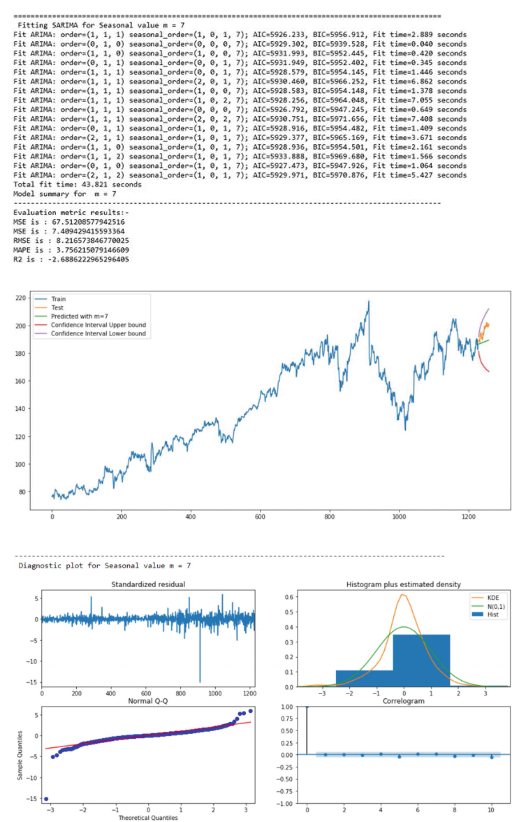


Рисунок 3.17 – SARIMA по сезонной стоимости m = 7

Подгонка SARIMA по сезонной стоимости m = 52 (рисунок 3.18).

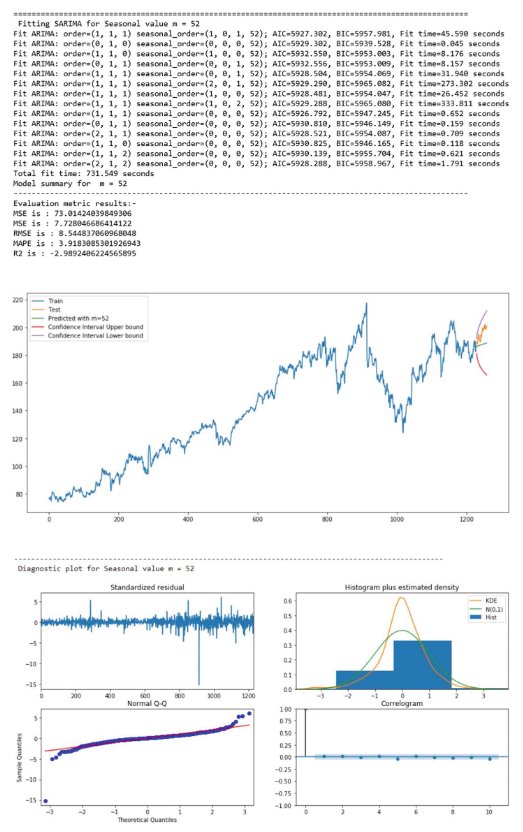


Рисунок 3.18 – SARIMA по сезонной стоимости m = 52

# Заключение

Цель данной работы состояла в разработке моделей прогнозирования временных рядов, которая позволяет прогнозировать будущие событии путем анализа прошлых тенденций, основанный на предположении, что будущие тенденции будут аналогичны историческим тенденциям..

В первом разделе мы познакомились с основные понятиями временных рядов, что такое временные ряды, с типами временных рядов, почему мы проводим анализ временных рядов.

Во втором разделе мы изучили что, ключевым моментом, который следует усвоить из этих экспериментов, является то, что стохастические модели (SARIMA) плохо справляются с обработкой временных рядов с множественными сезонными изменениями без каких-либо экзогенных переменных, и набор данных о температуре является хорошим примером этой слабости. Для временных рядов, где трудно извлечь сезонный компонент SARIMA (который сводится к ARIMA) не могут обеспечить среднесрочные или долгосрочные прогнозы, поскольку прогнозируемые значения имеют тенденцию сходиться к среднему значению последних наблюдаемых значений. Но когда временной ряд представляет собой уникальную сезонную составляющую, SARIMA, как правило, дают превосходные результаты (показатель: таблица 2.3). Prophet имеет производительность, очень сравнимую с моделью SAPTF, но не определяет автоматически субдневные периоды и не может предоставлять прогнозы для наборов данных без индексов в формате даты и времени.

Что касается моделей глубокого обучения, то они изо всех сил пытаются обеспечить хорошие прогнозы для коротких временных рядов, поскольку эти модели созданы для захвата сложных закономерностей. RNN способны напрямую моделировать сезонность, если ряд в наборе данных содержит однородные сезонные закономерности. Таким образом, для моделей глубокого обучения важно, чтобы у нас была достаточная длина временных рядов и однородная модель (отсутствие частых внезапных точек изменения). Однако SAPTF, SARIMA, Prophet являются единственными объяснимыми моделями. Это сильная сторона стохастических моделей по сравнению с моделями глубокого или даже некоторых моделей машинного обучения, где мы можем давать хорошие прогнозы, но не всегда можем объяснить, почему.

В целом, как показано в Таблице 2.4, SAPTF обеспечивает в среднем лучший прогноз (среднее значение TestMAPE = 16,9%), даже несмотря на то, что у него могут возникнуть проблемы с некоторыми наборами данных, за которыми следуют Prophet, SARIMA, SVR и GRU, которые имеют очень похожие показатели (около 22% MAPE в среднем).

При обработке коротких временных рядов мы видим, что SVR на 80% быстрее, чем SAPTF (эталон), тогда как остальные оценщики медленнее, чем этот оценщик, а prophet идет вторым и в 3 раза медленнее. Но эти значения (таблица 2.5), даже если они кажутся огромными, отражают лишь выигрыш в несколько секунд, поскольку базовый уровень уже мал. Но для более крупных временных рядов, где количество точек данных на временной ряд может превышать 1000, здесь мы снова видим SVR наверху (55%-е увеличение времени прогноза), за ним следует Prophet (14%-е увеличение времени прогноза), затем SAPTF.

В третьем разделе мы узнали о стационарных данных, включая типы стационарных данных, интерпретацию p-значений и способы сделать данные стационарными. Мы также углубились в математику и реализацию AR, MA, ARIMA, SARIMA. После проверки различных значений m мы видим, что m не оказывает никакого влияния на результаты.

# Список использованных источников

1. Эйлин Нильсен: Практический анализ временных рядов.Прогнозирование со статистикой и машинное обучение - <https://www.labirint.ru/books/786694/>
2. Модель авторегрессии и скользящего среднего ARMA(p,q) - <https://www.youtube.com/watch?v=pQm7ZDgB1tA&ab_channel=DataMininginAction>
3. Модель прогнозирования временных рядов по выборке максимального подобия - <https://www.mbureau.ru/articles/dissertaciya-model-prognozirovaniya-vremennyh-ryadov-glava-1>
4. ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ ARIMA В PYTHON 3 - <https://www.8host.com/blog/prognozirovanie-vremennyx-ryadov-s-pomoshhyu-arima-v-python-3/>
5. ADF тест для анализа временных рядов - <https://russianblogs.com/article/4449640394/>
6. Введение во временные ряды - <https://www.ibm.com/docs/ru/spss-statistics/SaaS?topic=forecasting-introduction-time-series>
7. Временной ряд - <https://blog.skillfactory.ru/glossary/vremennoj-ryad-2/>
8. Time Series Forecasting: What, Why, and, How? - <https://www.projectpro.io/article/time-series-forecasting-models/559#faq-q-1>
9. 11 Classical Time Series Forecasting Methods in Python (Cheat Sheet) - <https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/>
10. Базовые модели временных рядов - <https://studme.org/72681/ekonomika/bazovye_modeli_vremennyh_ryadov>
11. Временные ряды. Простые решения - <https://habr.com/ru/post/553658/>
12. Time Series Analysis in Python | Time Series Forecasting | Data Science with Python | Edureka - <https://www.youtube.com/watch?v=e8Yw4alG16Q&t=1752s&ab_channel=edureka%21>

# Приложение А. Программный код приложения

**Программный код моделей AR и MA**

**import pandas as pd**  
  
**df = pd.read\_csv(r'\Data\daily-total-female-births-CA.csv')**  
  
from statsmodels.tsa.arima\_model import ARIMA

import matplotlib.pyplot as plt

model = ARIMA(df['births'], order=(1, 1, 0))

results\_AR = model.fit(disp=-1)

plt.plot(results\_AR.fittedvalues, color='black')

plt.ylabel('Autoregressive (1)')

plt.xlabel('Time')

plt.rcParams["figure.figsize"] = [16,9]

plt.show()

from statsmodels.tsa.arima\_model import ARIMA

import matplotlib.pyplot as plt

model = ARIMA(df['births'], order=(2, 1, 0))

results\_AR = model.fit(disp=-1)

plt.plot(results\_AR.fittedvalues, color='black')

plt.ylabel('Autoregressive (2)')

plt.xlabel('Time')

plt.rcParams["figure.figsize"] = [16,9]

plt.show()

from statsmodels.tsa.arima\_model import ARIMA

import matplotlib.pyplot as plt

model = ARIMA(df['births'], order=(1, 1, 0))

results\_AR = model.fit(disp=-1)

plt.plot(results\_AR.fittedvalues, color='black')

plt.ylabel('Moving Average(1)')

plt.xlabel('Time')

plt.rcParams["figure.figsize"] = [16,9]

plt.show()

from statsmodels.tsa.arima\_model import ARIMA

import matplotlib.pyplot as plt

model = ARIMA(df['births'], order=(2, 1, 0))

results\_AR = model.fit(disp=-1)

plt.plot(results\_AR.fittedvalues, color='black')

plt.ylabel('Moving Average(2)')

plt.xlabel('Time')

plt.rcParams["figure.figsize"] = [16,9]

plt.show()

**Программный код модели ARIMA**

**import warnings**

**import matplotlib.pyplot as plt**

**import numpy as np**

**import pandas as pd**

**import statsmodels.api as sm**

**from pmdarima import auto\_arima**

**from sklearn import metrics**

**from statsmodels.tsa.seasonal import seasonal\_decompose**

**from statsmodels.tsa.stattools import adfuller**

**warnings.filterwarnings("ignore")**  
  
df = pd.read\_csv(r'\Data\FB.csv', parse\_dates = True)

df.head(10)

df["Close"].plot(figsize=(15, 6))

plt.xlabel("Date")

plt.ylabel("Close")

plt.title("Closing price of Facebook stocks")

plt.show()

plt.figure(1, figsize=(15,6))

plt.subplot(211)

df["Close"].hist()

plt.subplot(212)

df["Close"].plot(kind='kde')

plt.show()

def timeseries\_evaluation\_metrics\_func(y\_true, y\_pred):

def mean\_absolute\_percentage\_error(y\_true, y\_pred):

y\_true, y\_pred = np.array(y\_true), np.array(y\_pred)

return np.mean(np.abs((y\_true - y\_pred) / y\_true)) \* 100

print('Evaluation metric results:-')

print(f'MSE is : {metrics.mean\_squared\_error(y\_true, y\_pred)}')

print(f'MSE is : {metrics.mean\_absolute\_error(y\_true, y\_pred)}')

print(f'RMSE is : {np.sqrt(metrics.mean\_squared\_error(y\_true, y\_pred))}')

print(f'MAPE is : {mean\_absolute\_percentage\_error(y\_true, y\_pred)}')

print(f'R2 is : {metrics.r2\_score(y\_true, y\_pred)}',end='\n\n')

def Augmented\_Dickey\_Fuller\_Test\_func(series , column\_name):

print (f'Results of Dickey-Fuller Test for column: {column\_name}')

dftest = adfuller(series, autolag='AIC')

dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','No Lags Used','Number of Observations Used'])

for key,value in dftest[4].items():

dfoutput['Critical Value (%s)'%key] = value

print (dfoutput)

if dftest[1] <= 0.05:

print("Conclusion:====>")

print("Reject the null hypothesis")

print("Data is stationary")

else:

print("Conclusion:====>")

print("Fail to reject the null hypothesis")

print("Data is non-stationary")

Augmented\_Dickey\_Fuller\_Test\_func(df['Close' ],'Close')

X = df[['Close' ]]

train, test = X[0:-30], X[-30:]

stepwise\_model = auto\_arima(train,start\_p=1, start\_q=1,

max\_p=7, max\_q=7, seasonal=False,

d=None, trace=True,error\_action='ignore',suppress\_warnings=True, stepwise=True)

stepwise\_model.summary()

forecast,conf\_int = stepwise\_model.predict(n\_periods=30,return\_conf\_int=True)

forecast = pd.DataFrame(forecast,columns=['close\_pred'])

df\_conf = pd.DataFrame(conf\_int,columns= ['Upper\_bound','Lower\_bound'])

df\_conf["new\_index"] = range(1229, 1259)

df\_conf = df\_conf.set\_index("new\_index")

timeseries\_evaluation\_metrics\_func(test, forecast)

forecast["new\_index"] = range(1229, 1259)

forecast = forecast.set\_index("new\_index")

import matplotlib.pyplot as plt

%matplotlib inline

plt.rcParams["figure.figsize"] = [15,7]

plt.plot( train, label='Train ')

plt.plot(test, label='Test ')

plt.plot(forecast, label='Predicted ')

plt.plot(df\_conf['Upper\_bound'], label='Confidence Interval Upper bound ')

plt.plot(df\_conf['Lower\_bound'], label='Confidence Interval Lower bound ')

plt.legend(loc='best')

plt.show()

stepwise\_model.plot\_diagnostics();

**Программный код модели SARIMA**

**import warnings**

**import matplotlib.pyplot as plt**

**import numpy as np**

**import pandas as pd**

**import statsmodels.api as sm**

**from pmdarima import auto\_arima**

**from sklearn import metrics**

**from statsmodels.tsa.seasonal import seasonal\_decompose**

**from statsmodels.tsa.stattools import adfuller**

**warnings.filterwarnings("ignore")**

**df = pd.read\_csv(r'\Data\FB.csv')**

**df.head(10)**

**def timeseries\_evaluation\_metrics\_func(y\_true, y\_pred):**

**def mean\_absolute\_percentage\_error(y\_true, y\_pred):**

**y\_true, y\_pred = np.array(y\_true), np.array(y\_pred)**

**return np.mean(np.abs((y\_true - y\_pred) / y\_true)) \* 100**

**print('Evaluation metric results:-')**

**print(f'MSE is : {metrics.mean\_squared\_error(y\_true, y\_pred)}')**

**print(f'MSE is : {metrics.mean\_absolute\_error(y\_true, y\_pred)}')**

**print(f'RMSE is : {np.sqrt(metrics.mean\_squared\_error(y\_true, y\_pred))}')**

**print(f'MAPE is : {mean\_absolute\_percentage\_error(y\_true, y\_pred)}')**

**print(f'R2 is : {metrics.r2\_score(y\_true, y\_pred)}',end='\n\n')**

**def Augmented\_Dickey\_Fuller\_Test\_func(series , column\_name):**

**print (f'Results of Dickey-Fuller Test for column: {column\_name}')**

**dftest = adfuller(series, autolag='AIC')**

**dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','No Lags Used','Number of Observations Used'])**

**for key,value in dftest[4].items():**

**dfoutput['Critical Value (%s)'%key] = value**

**print (dfoutput)**

**if dftest[1] <= 0.05:**

**print("Conclusion:====>")**

**print("Reject the null hypothesis")**

**print("Data is stationary")**

**else:**

**print("Conclusion:====>")**

**print("Fail to reject the null hypothesis")**

**print("Data is non-stationary")**

**Augmented\_Dickey\_Fuller\_Test\_func(df['Close' ],'Close')**

**X = df[['Close' ]]**

**train, test = X[0:-30], X[-30:]**

**for m in [1, 4,7,12,52]:**

**print("="\*100)**

**print(f' Fitting SARIMA for Seasonal value m = {str(m)}')**

**stepwise\_model = auto\_arima(train, start\_p=1, start\_q=1,**

**max\_p=7, max\_q=7, seasonal=True, start\_P=1, start\_Q=1, max\_P=7, max\_D=7, max\_Q=7, m=m,**

**d=None, D=None, trace=True, error\_action='ignore', suppress\_warnings=True, stepwise=True)**

**print(f'Model summary for m = {str(m)}')**

**print("-"\*100)**

**stepwise\_model.summary()**

**forecast ,conf\_int= stepwise\_model.predict(n\_periods=30,return\_conf\_int=True)**

**df\_conf = pd.DataFrame(conf\_int,columns= ['Upper\_bound','Lower\_bound'])**

**df\_conf["new\_index"] = range(1229, 1259)**

**df\_conf = df\_conf.set\_index("new\_index")**

**forecast = pd.DataFrame(forecast, columns=['close\_pred'])**

**forecast["new\_index"] = range(1229, 1259)**

**forecast = forecast.set\_index("new\_index")**

**timeseries\_evaluation\_metrics\_func(test, forecast)**

**import matplotlib.pyplot as plt**

**%matplotlib inline**

**plt.rcParams["figure.figsize"] = [15, 7]**

**plt.plot(train, label='Train ')**

**plt.plot(test, label='Test ')**

**plt.plot(forecast, label=f'Predicted with m={str(m)} ')**

**plt.plot(df\_conf['Upper\_bound'], label='Confidence Interval Upper bound ')**

**plt.plot(df\_conf['Lower\_bound'], label='Confidence Interval Lower bound ')**

**plt.legend(loc='best')**

**plt.show()**

**print("-"\*100)**

**print(f' Diagnostic plot for Seasonal value m = {str(m)}')**

**display(stepwise\_model.plot\_diagnostics());**

**print("-"\*100)**