НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.855.5:519.876.2

До захисту допущено
В. о. завідувача кафедри ММСА
Тимощук О. Л.
«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 Комп'ютерні науки
на тему: «Інформаційна система для прогнозування котирування акцій методами машинного навчання»

Виконав:
студент ІІ курсу, групи КА-93мп
Ночовний Олексій Олександрович _____

Керівник: доцент кафедри ММСА
к.т.н., доц. Жиров О.Л. _____

Рецензент: доцент кафедри системного проектування
КПІ ім. Ігоря Сікорського,
к.т.н., доц. Кисельов Г. Д. _____

Засвідчую, що в цій магістерській дисертації
немає запозичень із праць інших авторів
без відповідних посилань

Студент _____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ММСА
О. Л. Тимощук
«___» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студента Ночовного Олексія Олександровича

1. Тема дисертації: «Інформаційна система для прогнозування котирування акцій методами машинного навчання», науковий керівник дисертації Жиров Олександр Леонідович к.т.н., доцент, затверджені наказом по університету від «02» листопада 2020р. № 3182-с.

2. Термін подання студентом дисертації: 18 грудня 2020 р.

3. Об'єкт дослідження: прогноз котирування акцій;

4. Предмет дослідження: моделі для прогнозування котирування акцій;

5. Перелік завдань, які потрібно розробити:

1) дослідити сучасний стан та особливості методів прогнозування котирування акцій;
2) провести огляд сучасних підходів для побудови моделей прогнозування котирування акцій;
3) обрати та обґрунтувати вибір декількох моделей;
4) обробка вхідних даних;
5) застосувати оброблені дані на реалізованих моделях та проаналізувати отримані результат;
6) розробити стартап-проект виведення на ринок результатів дослідження;
7) розробити концептуальні висновки за результатами наукового дослідження.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

1. Рисунки схем роботи деяких алгоритмів машинного навчання;
2. Таблиці порівняння результатів роботи;
3. Рисунки та графіки на яких зображено результати роботи;

4. Таблиці у розділі стартап-проекту.

7. Дата видачі завдання: 01 вересня 2020 р.

Календарний план

з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1	Концептуальний вступ дисертації. Формулювання об’єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	05.09.2020—13.09. 2020
2	Перший розділ. Огляд літературно-інформаційних джерел, формування нормативної бази. Характеристика об’єкта.	16.09.2020—27.09.2020
3	Другий розділ. Математичні розрахунки та сегментація показників оцінювання платоспроможності. Застосування математичних моделей для прогнозування дефолту позичальника.	30.09.2020—18.10.2020
4	Третій розділ. Обробка вхідних даних. Реалізація та застосування моделей. Збір та аналіз результатів.	21.10.2020—29.10.2020
5	Четвертий розділ. Стартап-проект	30.10.2020—17.11.2020
6	Концептуальні висновки. Перспективи розвитку отриманих рішень	22.11.2020—26.11.2020

Студент Ночовний О.О.

Науковий керівник дисертації Жиров О.Л.

РЕФЕРАТ

Магістерська дисертація: 80 с., 22 рис., 19 табл., 1 додаток, 18 джерел.

ЧАСОВІ РЯДИ, СТОХАСТИЧНА МОДЕЛЬ АВТОРЕГРЕСІЇ З ІНТЕГРОВАНИМ КОВЗНИМ СЕРЕДНІМ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, LSTM, LIGHTGBM,FACEBOOK PROPHET

Моделювання та прогнозування часових рядів має принципове значення для різного практичного застосування. В зв'язку з цим, протягом останніх років у цій темі було безліч наукових робіт. У літературі запропоновано багато важливих моделей для підвищення точності та ефективність моделювання та прогнозування часових рядів, а також огляд самих моделей для прогнозування часових рядів.

Актуальність дисертації зумовлена надзвичайним розвитком машинного навчання, а саме нейромережевих технологій та штучного інтелекту, потребою в покращенні результатів прогнозування.

Мета дипломної робот - опис та порівняльний аналіз такого надзвичайно популярного методу для прогнозування котирування акцій як АРІКС з новітніми розробками в галузі прогнозування - Facebook Prophet та такими методами машинного навчання як LightGBM та LSTM.

Предметом дослідження є нейронна мережа на основі рекурентної архітектури, прогнозування даних на основі адитивної моделі, дерево рішень і їх можливість та перспективи у сфері фінансового прогнозування.

Об’єктом мого дослідження є котирування акцій представлені у вигляді часових рядів на основі статистичних даних стосовно їхньої динаміки.

ABSTRACT

The master’s thesis: 80 pages 22 images 19 tables 18 sources

The theme: «Information system for prediction stock quotes using machine learning».

TIME SERIES, STOCHASTIC AUTOREGRESSION MODEL WITH INTEGRATED SLIDING AVERAGE, ARTIFICIAL NEURAL NETWORKS, LSTM, LIGHTGBM, FACEBOOK PROPHET

Modeling and forecasting of time series is of fundamental importance for various practical applications. In this regard, in recent years there have been many scientific papers on this topic. The literature offers many important models for improving the accuracy and efficiency of modeling and forecasting time series, as well as an overview of the models themselves for forecasting time series.

The relevance of the dissertation is due to the extraordinary development of machine learning, namely neural network technologies and artificial intelligence, the need to improve the results of forecasting.

The purpose of the thesis is to describe and compare such an extremely popular method for forecasting stock quotes as ARIMA with the latest developments in the field of forecasting - Facebook Prophet and such machine learning methods as LightGBM and LSTM.

The subject of research is neural networks based on recurrent architecture, data forecasting based on the additive model, the decision tree and their capabilities and prospects in the field of financial forecasting.

The object of my research is stock quotes presented in the form of time series based on statistics on their dynamics.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

AR(АР) – авторегресійне рівняння

ARMA(АРКС) – авторегресійне рівняння з ковзним середнім

ARIMA(АРІКС) – авторегресійне рівняння з інтегрованим ковзним середнім

SES(ЕЗ) – експоненціальне згладжування

DW – критерій Дарбіна-Уотсона

ACF(АКФ) – автокореляційна функція

PACF(ЧАКФ) – часткова автокореляційна функція

SSE(СКП) – сума квадратів похибок моделі

СП – середня похибка прогнозу

СПП – середня похибка в процентах

RMSE – середньоквадратична похибка

MAPE(СаПП) – середня абсолютна похибка у процентах

R2 – коефіцієнт детермінації

AIC – інформаційний критерій Акайке

LSTM – Long short-term memory

LightGBM – Light Gradient Boosting Machine

ВСТУП

На сьогоднішній день зростає актуальність ефективного вирішення практичних проблем, що включають в себе обробку даних із прихованими кореляціями. У цей клас входить широкий спектр задач, зокрема прогнозування часових рядів, які широко застосовуються у економіці, фізиці та багатьох практичних дисциплінах, неантичний аналіз тексту та ідентифікація природньої мови, нейроаналіз текстовий даних, задачі динамічного регулювання та задачі теорії автоматичного управління.

Варто зауважити, що на сьогоднішній день вже сформовані класичні методи вирішення проблем цього класу у кожній з предметних областей. Наприклад, для прогнозування часових рядів вже більше 30 років застосовують авторегресійний аналіз і метод Бокса-Дженсінса, для регулювання перехідних процесів застосовують ПІД-регулятори, а для аналізу тексту — теорію автоматів та граматик. Проте класичні методи часто програють застосування нейронних мереж не тільки за швидкістю реалізації, а й за якісними характеристиками, такими як точність прогнозування, час перехідного процесу, точність розпізнавання тексту тощо.

Інтерес наукового суспільства і бізнесу до машинного навчання зростає з кожним роком. Все більше і більше з’являється прикладів успішного застосування штучного інтелекту у найрізноманітніших галузях діяльності людини, значно зростає кількість підприємств що впроваджують нейронні мережі у свою операційну діяльність. Розв’язання задач управління, прогнозування а також класифікації все частіше покладають на штучний інтелект у багатьох галузях діяльності людини.

Підготовка високоякісних прогнозів - непросте завдання як для машин, так і для більшості аналітиків. Під час дослідження джерел на цю тематику виявлено два основних напрями в практиці створення різноманітних бізнес-прогнозів:

1. Повністю автоматичні методи прогнозування можуть бути крихкими і часто дуже не гнучкими, щоб включати в себе корисні припущення або евристику.
2. Аналітики, які можуть розробляти високоякісні прогнози, зустрічаються досить рідко, оскільки прогнозування - це спеціалізований навик в області науки про дані, що вимагає значного досвіду.

Результат цих досліджень полягає в тому, що попит на високоякісні прогнози часто набагато перевищує темпи росту, з якими аналітики можуть їх робити. Це спостереження є мотивацією для моєї роботи.

Чи серед вказаних моделей АРІКС є першим, про що Ви думаєте, коли чуєте про часові ряди? Можливо, прийшов час вивчити інші авантюри і методики. В даний час активно розвивається безліч нових інновацій і сучасних методик, і деякі з них перевершують традиційні моделі АРІКС.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Специфіка економічного прогнозування

Економічне прогнозування - це процес складання прогнозів про економіку. Прогнози можуть бути зроблені на високому рівні агрегування - наприклад, по ВВП, інфляції, безробіття або бюджетного дефіциту - або на більш дезагреговані рівні, по конкретним секторам економіки або навіть по конкретним фірмам. Економічне прогнозування є тим, що дозволяє з'ясувати майбутнє процвітання моделі інвестування, і є ключовим видом діяльності в економічному аналізі.

Багато установ займаються економічним прогнозуванням: національні уряди, банки і центральні банки, консультанти та організації приватного сектора, такі як аналітичні центри, компанії та міжнародні організації, такі як Міжнародний валютний фонд, Світовий банк і ОЕСР. Деякі прогнози складаються щорічно, але багато хто з них оновлюються частіше.

Економіст, як правило, розглядає ризики (тобто події або умови, які можуть привести до того, що результат буде відрізнятися від своїх початкових оцінок). Ці ризики допомагають проілюструвати процес обґрунтування, використовуваний для отримання остаточних цифр прогнозів. Економісти зазвичай використовують коментарі поряд з інструментами візуалізації даних, такими як таблиці і діаграми, для передачі свого прогнозу. При підготовці економічних прогнозів у спробі підвищити точність використовувалася різноманітна інформація.

Для отримання більш точних прогнозів використовувалося все, починаючи від макроекономічних, мікроекономічних, ринкових даних майбутнього, машинного навчання (штучні нейронні мережі), і закінчуючи поведінковими дослідженнями людини . Прогнози використовуються для різних цілей. Уряду і бізнес використовують економічні прогнози, щоб допомогти їм визначити свою стратегію, багаторічні плани і бюджети на майбутній рік. Аналітики фондового ринку використовують прогнози, щоб допомогти їм оцінити вартість компанії і її акцій.

Економісти вибирають, які змінні важливі для обговорюваного матеріалу. Економісти можуть використовувати статистичний аналіз історичних даних для визначення очевидних зв'язків між конкретними незалежними змінними і їх зв'язку з досліджуваної залежною змінною. Наприклад, в якій мірі зміни цін на житло вплинули на чисту вартість житла в цілому по населенню в минулому? Цей зв'язок потім може бути використана для прогнозування майбутнього. Тобто, якщо ціни на житло, як очікується, будуть змінюватися певним чином, який вплив це матиме на майбутню чисту вартість житла для населення? Прогнози, як правило, ґрунтуються на вибіркових даних, а не на повній сукупності, що вносить невизначеність. Економіст проводить статистичні тести і розробляє статистичні моделі (часто з використанням регресійного аналізу) для визначення того, які стосунки найкраще описують або пророкують поведінку досліджуваних змінних. Історичні дані і припущення про майбутнє застосовуються до моделі при побудові прогнозу для конкретних змінних.

Процес економічного прогнозування аналогічний аналізу даних і призводить до розрахункових значень ключових економічних змінних в майбутньому. Економіст застосовує методи економетрики в процесі їх прогнозування. Типові етапи можуть включати в себе:

1) область застосування: Ключові економічні змінні і теми для прогнозних коментарів визначаються виходячи з потреб аудиторії прогнозу;
2) огляд літератури: Коментарі з джерел з короткою перспективою, таких як МВФ, ОЕСР, Федеральна резервна система США і СВО допомагають визначити ключові економічні тенденції, проблеми та ризики. Такий коментар також може допомогти синоптикам з їх власними припущеннями, а також дати їм інші прогнози для порівняння;
3) вихідні дані: Історичні дані збираються з ключових економічних змінних. Ці дані містяться як в друкованих, так і в електронних джерелах, таких як FRED база даних або Eurostat, які дозволяють

користувачам дізнаватися історичні значення змінних до яких є інтерес;

4) визначити історичні зв'язки: Історичні дані використовуються для визначення зв'язків між однією або декількома незалежними змінними і досліджуваної залежної змінної, часто за допомогою регресійного аналізу;
5) модель: Історичні дані і припущення використовуються для розробки економетричної моделі. Моделі зазвичай застосовують обчислення до ряду вхідних даних для створення економічного прогнозу для однієї або декількох змінних;
6) звіт: Результати моделі включаються в звіти, які зазвичай включають інформаційні графіки і коментарі, щоб допомогти читачеві зрозуміти прогноз.

Аналітики можуть використовувати обчислювальні моделі загальної рівноваги або динамічні стохастичні моделі загальної рівноваги. Останні часто використовуються центральними банками.

Методи прогнозування включають в себе економетричні моделі, консенсус-прогнози, аналіз економічної бази, аналіз Shift-акцій, модель входу-виходу і модель Грінольда і Кронер.

1.2 Приклади часових рядів

У цьому розділі будуть наведені приклади часових рядів, і можливі застосування аналізу часових рядів. приклади містять як типові приклади з економічних досліджень, так і більш технічні застосування.

1.2.1 Обмінний курс долара до євро

Перший приклад - щоденний міжбанківський обмінний курс долара США до євро. зображений на рисунку 1.1. Це типовий економічний часовий ряд, в якому аналіз можна було б використовувати для формулювання моделі прогнозування майбутньої ціни обмінного курс.

Рисунок 1.1 - Щоденний міжбанківський обмінний курс долара США до євро.

1.2.2 Кількість щомісячних авіапасажирів

Далі розглянемо показане кількість щомісячних авіапасажирів в США на рис. 1.2. Для даної серії спостерігається явна річна варіативність. Знову ж це може бути корисним при побудові моделі для складання прогнозів про майбутню кількість пасажирів авіакомпаній. Моделі і методи аналізу часових рядів з урахуванням сезонності.

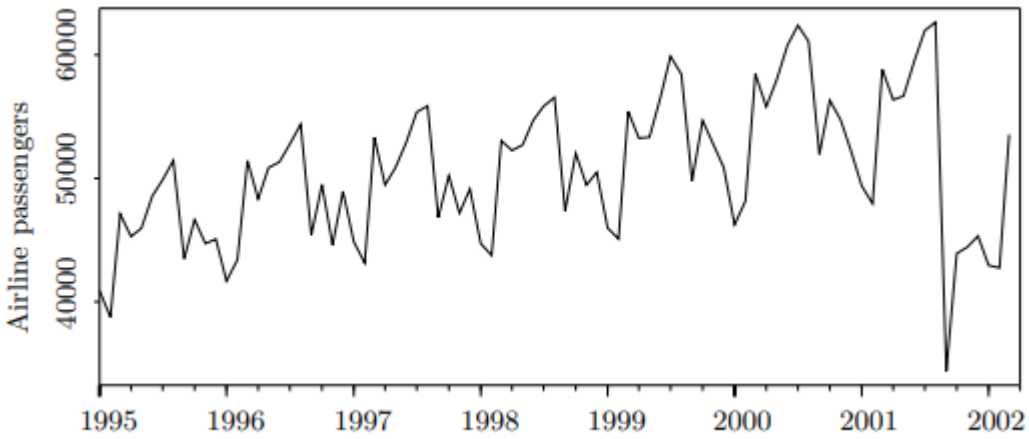

Рисунок 1.2 - Кількість щомісячних авіапасажирів в США. В ряді можна побачити чітку річну варіацію.

1.2.3 Теплова динаміка будівлі

Тепер розглянемо більш технічний приклад. На рисунку 1.3 на наступній сторінці показано вимірювання з тестового будівлі без мешканців. Дані на нижньому графіку показують температуру повітря в приміщенні, а на верхньому графіку - температуру навколишнього повітря, теплопостачання і сонячне випромінювання.

Для цього прикладу може бути цікаво охарактеризувати теплову поведінку будівлі. В рамках цього можна оцінити так званий опір тепловому потоку зсередини назовні. Опір характеризує ізоляцію будівлі. Також може бути корисним створення динамічної моделі будівлі і оцінка констант часу. Знання констант часу може бути використано для проектування оптимальних контролерів теплопостачання.

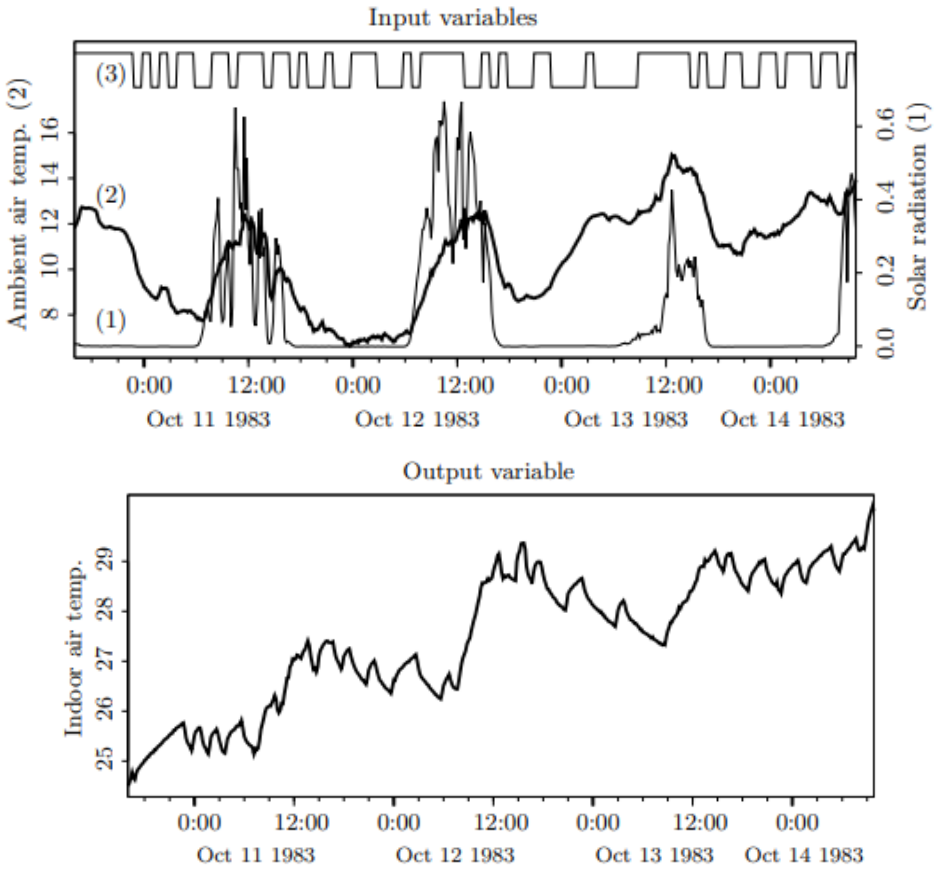

Рисунок 1.3 - Вимірювання з випробувальної будівлі без мешканців.

Вхідними змінними є (1) сонячне випромінювання, (2) температура навколишнього повітря і (3) теплової введення. Змінної на виході є температура повітря в приміщенні.

1.2.4 Взаємостосунки хижак-здобич

Цей приклад ілюструє типовий багатовимірний часовий ряд, оскільки неможливо класифікувати один з рядів як вхідний, а інший - як вихідний. На рисунку 1.4 показаний широко вивчений випадок хижак-здобич, а саме, серія щорічно продаваних шкур ондатри і норки компанії Hudson's Bay Company.

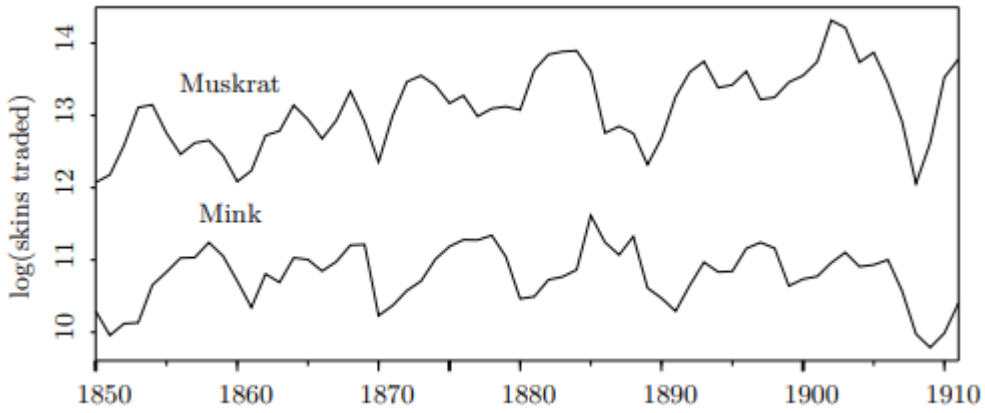

Рисунок 1.4 - Щорічна кількість шкури ондатра і норки Гудзонової затоки після логарифмічною трансформації. Неможливо класифікувати одну з серій як вхідну, а іншу серію як вихідну.

Протягом 62 років з 1850 по 1911 рік. Фактично, популяція ондатри залежить від популяції норки, а популяція норки залежить від кількості ондатри. У таких випадках обидва ряди повинні бути включені в багатовимірний часовий ряд.

1.3 Загальні кроки в процесі прогнозування

5 основних кроків в завданні прогнозування узагальнені Хіндманом і Атанасопулосом в їх книзі "Прогнозування: принципи і практика". Цими кроками є:

1) визначення проблеми. Ретельний огляд того, кому потрібен прогноз і як він буде використовуватися. Це описується як найважча частина процесу, швидше за все, тому, що вона є повністю проблемною і суб'єктивною;
2) збір інформації. Збір історичних даних для аналізу і моделювання. Сюди також входить отримання доступу до експертів в області і збір інформації, яка може допомогти найкращим чином інтерпретувати історичну інформацію і, в кінцевому рахунку, прогнози, які будуть зроблені;

3) попередній дослідний аналіз. Використання простих інструментів, таких як графіки та зведені статистичні дані, для кращого розуміння даних. Перегляд графіків, визначення очевидної часової структури, такі як сезонність трендів, аномалії, такі як відсутність даних, відхилення, і будь-які інші структури, які можуть впливати на прогнозування;
4) вибір і настройка моделей. Оцінка двух, трьох або набір моделей різного типу. Моделі можуть бути обрані для оцінки, ґрунтуючись на припущеннях, які вони роблять, і на те, чи відповідає набір даних. Моделі конфігуруютья і підганяються під історичні дані;
5) використання і оцінка моделі прогнозування. Модель використовується для складання прогнозів, і продуктивність цих прогнозів оцінюється, і оцінюється майстерність моделей. Це може включати в себе зворотне тестування з історичними даними або очікування нових спостережень для порівняння.

Цей 5-ступінчастий процес забезпечує достатній огляд, починаючи з ідеї або постановки завдання і закінчуючи моделлю, яка може бути використана для складання прогнозів. У центрі уваги процесу знаходиться розуміння проблеми і підгонка оптимальної моделі.

1.4 Огляд Літератури

Вілсон і Шарда [1] вивчали прогнозування банкрутства фірми, використовуючи нейронні мережі і класичний множинний дискримінантний аналіз, де нейронні мережі виконували значно краще, ніж множинний дискримінантний аналіз. Вони оцінювали методи, засновані на Support Vector Machine, множині дискримінантному аналізі, логістичному регресійному аналізі і тришарових повністю підключених нейронних мережах зворотного поширення. Чи намагався передбачити кредитний рейтинг компанії, що використовує векторні машини підтримки, використовуючи різні фінансові показники і

коефіцієнти, такі як процентне покриття, звичайний дохід до загальної суми як чистий прибуток до власного капіталу зацікавлених сторін, коефіцієнт поточних зобов'язань і т.д., а також досягли точності близько 60%. Прогнозування кредитного рейтингу компаній також вивчалося з використанням нейронних мереж, досягаючи точності від 75% до 80% для ринків США і Тайваню.

Цай і Ван [2] провели дослідження, в якому вони спробували передбачити ціни акцій за допомогою навчання з використанням з дерев прийняття рішень і штучних нейронних мереж. Вони створили набір даних по тайванським фондовим ринкам, беручи до уваги фундаментальні індекси, технічні індекси і макроекономічні індекси. Продуктивність Дерево рішень + штучна нейронна мережа, навчена за даними Тайванської фондової біржі, показала продуктивність Fscore 77%. Моноалгоритми показали ефективність F-score до 67% .

Кім і Хань [3] використовували генетичний алгоритм для перетворення безперервних вхідних значень в інтродіскретние. Генетичний алгоритм був використаний для зниження складності функціоналу, в даній роботі запропоновано новий еволюційний метод обчислень, званий генетичним квантовим алгоритмом. Генетичний квантовий алгоритм заснований на концепції і принципи квантових обчислень, таких як кубіти і суперпозиції станів. Замість бінарного, числового чи символічного уявлення, прийнявши в якості уявлення бітову хромосому, генетичний квантовий алгоритм може являти собою лінійне накладення рішень завдяки своєму вероятностному поданням. Асгенетіческіе оператори, квантові затвори використовуються для пошуку найкращого рішення, існує безліч інструментів і програмного забезпечення, які забезпечують прогнозування об'єктів фондового ринку, кількості і вартості акцій для даної фінансової організації. Більшість з них претендують на прогнозування фондового ринку з точністю до 100%, але думки користувачів варіюються. Деякі з популярних інструментів і програмного забезпечення з їх методологіями згадуються.

Висновки до розділу 1

Можна зробити висновок що дослідження в даній сфері проводяться і тема є надзвичайно популярною серед дослідників. Аналіз застосування традиційних методологій прогнозування показує, що вони не відповідають сучасним потребам ринку у прогнозуванні фінансових активів. Більшість методів розглядають вирішення задачі прогнозування лише з одного боку, беручі до уваги лише частину факторів, що впливають на зміну курсу фінансового активу і не враховують надважливі фактори, такі як реакцію людей у соціальних мережах і фінансові новини. У той час як фундаментальний аналіз розглядає зовнішні фактори, його застосування у системах прогнозування досить обмежене через складність реалізації і підтримки, або ж навпаки його реалізація досить неповна і орієнтується на зміну основних економічних і політичних індексів і звітів. Існує потужний методологічний апарат для обробки часових рядів, методи авторегресії, ковзного середнього та інші. Проте економічні процеси керуються подіями, а часовий ряд (графік зміни курсу) зазвичай вже є наслідком цих подій.

При побудові правильної моделі часових рядів необхідно враховувати принцип Парсімона. Відповідно до цього принципу завжди вибирається модель з найменшою кількістю параметрів, щоб забезпечити адекватне уявлення вихідних даних часових рядів. З безлічі відповідних моделей слід розглядати найпростішу, зберігаючи при цьому точний опис властивостей, властивих часовому ряду. Ідея модельної стриманості схожа на відомий бритвений принцип Оккама.

Таким чином, підбиваючи підсумок, можна сказати, що, слід приділяти увагу вибору найбільш скупої моделі серед всіх інших варіантів.

РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ РОБОТИ

2.1 Методологія Бокса-Дженкінса

Після опису різних моделей часових рядів наступним питанням, яке турбує, є те, як вибрати належну модель, яка може дати точний прогноз, заснований на описі історичних подій, шаблон даних і як визначити оптимальні моделі. Статистики Джордж Бокс. і Гвілім Дженкінс розробили практичний підхід до побудови моделі ARIMA, яка найкращим чином підходить для вирішення цього завдання до певного часового ряду, а також задовольняють принципом скупості. Їх концепція має фундаментальне значення в області аналізу і прогнозування часових рядів.

Методологія Боксу-Дженкінса не передбачає будь-якої конкретної закономірності в історичних даних ряду, які необхідно спрогнозувати. Швидше можна сказати, що вона використовує триступеневий ітеративний підхід ідентифікації моделі, оцінки параметрів і діагностичної перевірки для визначення кращої економною моделі із загального класу АРІКС моделей. Цей триступеневий процес повторюється скільки разів, поки остаточно не буде обрана задовільна модель. Ця модель може бути використана для прогнозування майбутніх значень часового ряду.

Метод прогнозування Бокса-Дженкінса схематично показаний на рисунку 2.1

Рисунок 2.1 - Метод прогнозування Бокса-Дженкінса

Найважливішим кроком при виборі відповідної моделі є визначення оптимальних параметрів даної моделі. Одним з критеріїв є те, що вибірка АКФ і ЧАКФ, розрахована на основі даних навчання повинні збігатися з відповідними теоретичними або фактичними значеннями. Інші широко відомі критерії для ідентифікації моделі є критерій Акайке (AIC):

$$AIC(p) = n * ln\left(\frac{\widehat{\sigma_e}^2}{n}\right) + 2p$$

і Байєсівський інформаційний критерій (БІК):

$$BIC(p) = n * ln\ ln\left(\frac{\widehat{\sigma_e}^2}{n}\right) + p + pln(n),$$

де n - кількість ефективних спостережень, які використовуються для підгонки моделі, p - число в моделі, $\widehat{\sigma_e}^2$ - це сума квадратних залишків зразка.

Оптимальний порядок розташування моделей вибирається за кількістю параметрів моделі, що мінімізує або AIC, або BIC. Інші аналогічні критерії також були запропоновані в літературі для оптимальної ідентифікації моделі.

2.2 Прогнозування з використанням штучних нейронних мереж

У попередньому абзаці ми обговорили важливі стохастичні методи моделювання та прогнозування часових рядів. Основним інструментом є штучні нейронні мережі (ШНМ) були запропоновані в якості альтернативи прогнозування часових рядів і в останні роки широко використовуються. Подібно роботі людського мозку, ШНМ намагаються розпізнати закономірності у вхідних даних, винести уроки з досвіду і потім надати узагальнені результати, засновані на їх отриманих раніше знаннях.

Найбільш характерними особливостями ШНМ, які роблять їх вельми улюбленими для аналізу і прогнозування часових рядів наведено нижче.

По-перше, ШНМ за своєю природою засновані на даних і є самоадаптивними. Немає необхідності вказувати конкретну форму моделі або робити апріорні припущення про статистичному розподілі даних, бажана модель адаптивно формується на основі особливостей, представлених на основі даних. Такий підхід дуже корисний в багатьох практичних ситуаціях, коли теоретичних рекомендацій немає необхідних для відповідного процесу генерації даних.

По-друге, ШНМ за своєю природою є нелінійним перетворювачем, що робить їх більш практичними і точними в наступних областях моделювання складних моделей даних на відміну від різних традиційних лінійних підходів, таких як методи ARIMA. Існує безліч прикладів, які свідчать про те, що ШНМ зробили набагато кращий аналіз і прогнозування, ніж різні лінійні моделі.

Нарешті, на думку Хорніка і Стінчкомб, ШНМ є універсальними функціональними апроксиматорами. Вони показали, що мережа може наблизити будь-яку неперервну функцію до будь-якої необхідної точності. ШНМ використовують паралельну обробку інформації з даних для апроксимації великого класу функцій з високим ступенем точності. Крім того, вони можуть мати справу з ситуацією, коли вхідні дані помилкові, неповні або нечіткі.

2.3 Prophet

Сьогодні Facebook Procphet це інструмент прогнозування з відкритим вихідним кодом, доступний на Python і R. Проблеми масштабу, які можна спостерігати на практиці, пов'язану зі складністю, яку привносить різноманітність проблем прогнозування, і побудовою довіри до великої кількості прогнозів після того, як вони були зроблені. Prophet був ключовим елементом в поліпшенні здатності Facebook створювати велику кількість надійних прогнозів, які використовуються для прийняття рішень.

Не всі проблеми прогнозування можуть бути вирішені однієї і тієї ж процедурою. Prophet оптимізований під завдання бізнес-прогнозу, з якими зіткнулися в Facebook, які, як правило, мають одну з наведених нижче характеристик:

1) погодинні, щоденні або щотижневі спостереження з не менш ніж кількома місяцями (бажано річними) історії;
2) сильна множинна "людська" сезонність: день тижня і час року;
3) важливі свята, які відбуваються з нерегулярними, заздалегідь відомими інтервалами (наприклад, Суперкубок);
4) розумну кількість пропущених спостережень або великих відхилень;
5) історичні зміни трендів, наприклад, в зв'язку з запуском нових продуктів або занесенням в журнал змін;

6) тренди, які є нелінійними кривими зростання, коли тренд досягає природної межі або насичується.

Було знайдено настройки за замовчуванням Prophet, щоб виробляти прогнози, які часто бувають точними, як і ті, які виробляються кваліфікованими синоптиками, з набагато меншими зусиллями. З Prophet ви не застрягнете з результатами повністю автоматичної процедури, якщо прогноз незадовільний - аналітик, що не навчений методам часових рядів, може поліпшити або підкоригувати прогнози, використовуючи різні параметри з легкою інтерпретацією. Було виявлено що, комбінуючи автоматичне прогнозування з прогнозами аналітика в особливих випадках, можна охопити широкий спектр бізнес-кейсів. Наступна діаграма, представлена на рисунку 2.2, ілюструє процес прогнозування, який ми виявили для роботи в масштабі:

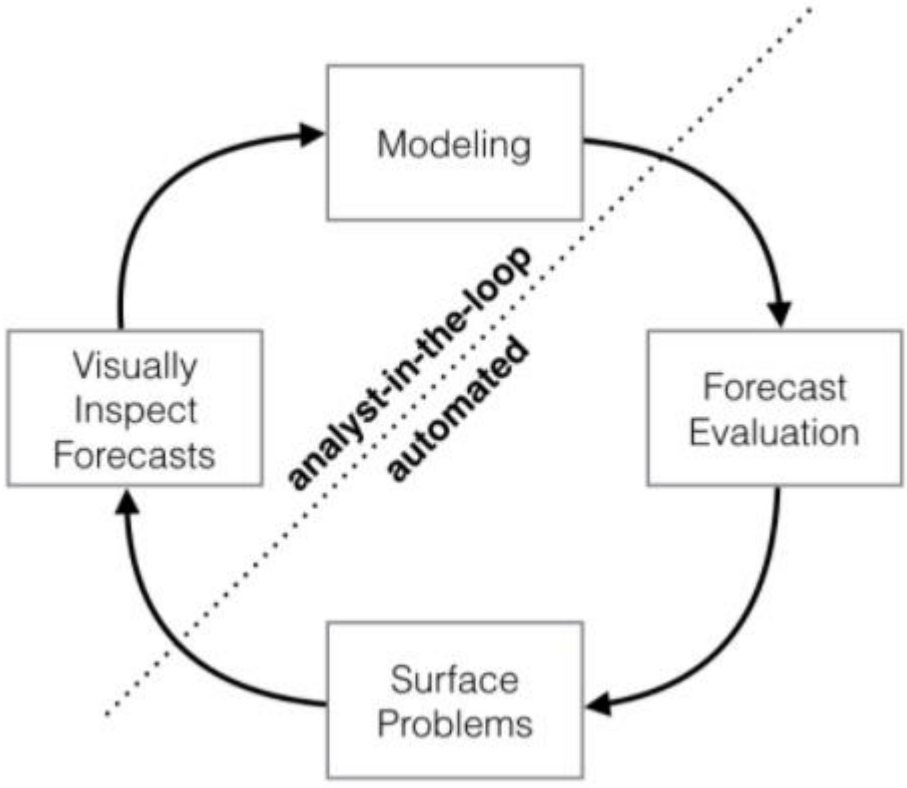

Рисунок 2.2 - Принцип роботи методів прогнозування

Для етапу моделювання процесу прогнозування в даний час є лише обмежена кількість інструментів. Відмінний пакет прогнозів Rob Hyndman в R, мабуть, найпопулярніший варіант, а Google і Twitter випустили пакети з більш

специфічним функціоналом часових рядів - CausalImpact і AnomalyDetection відповідно. Наскільки можена судити, на Python мало програм з відкритим вихідним кодом для прогнозування.

Дослідники часто використовували Prophet в якості заміни пакета прогнозів у багатьох налаштуваннях через дві основні переваги:

1. Prophet робить створення розумного і точного прогнозу набагато простіше. Пакет прогнозів включає в себе безліч різних методик прогнозування (ARIMA, експоненціальне згладжування і т.д.), кожна з яких має свої сильні і слабкі сторони, а також параметри налаштування. Було виявилено, що вибір неправильної моделі або параметрів часто це може призвести неякісне, і навряд чи навіть досвідчені аналітики зможуть ефективно вибрати правильну модель і параметри з урахуванням цього масиву варіантів.
2. Прогнози Prophet налаштовуються інтуїтивно зрозумілим для неспеціалістів способом. Існують параметри згладжування для сезонності, які дозволяють підлаштовуватися під історичні цикли, а також згладжують параметри для трендів, які дозволяють підлаштовуватися під те, наскільки агресивно можна стежити за історичними змінами трендів. Для кривих зростання можна вручну вказати "можливості" або верхня межа кривої зростання, дозволяючи вам вводити власну попередню інформацію про те, як ваш прогноз буде рости (або знижуватися). Нарешті, можна вказати нерегулярні свята для моделювання, наприклад, дати Суперкубка, Дня Подяки і Чорної П'ятниці.

2.4 Як працює Prophet

В основі процедури Prophet лежить аддитивная регресійна модель з чотирма основними компонентами:

1) кусково-лінійний або логістичний тренд кривої зростання. Prophet автоматично виявляє зміни трендів, вибираючи з даних точки зміни;
2) річний сезонний компонент, змодельований з використанням ряду Фур'є;
3) щотижневий сезонний компонент з використанням фіктивних змінних;
4) наданий користувачем список важливих свят.

Як приклад можна привести характерний прогноз: перегляд сторінки в Вікіпедії Peyton Manning, яку можна завантажити за допомогою пакета wikipediatrend. Оскільки Peyton Manning є американським футболістом, то можна бачити, що щорічна сезонність грає важливу роль, в той час як щотижнева періодичність також чітко присутня зображено на рисункку 2.3. Нарешті, видно, що деякі події (наприклад, ігри плей-офф, в яких він з'являється) також можуть бути змодельовані.

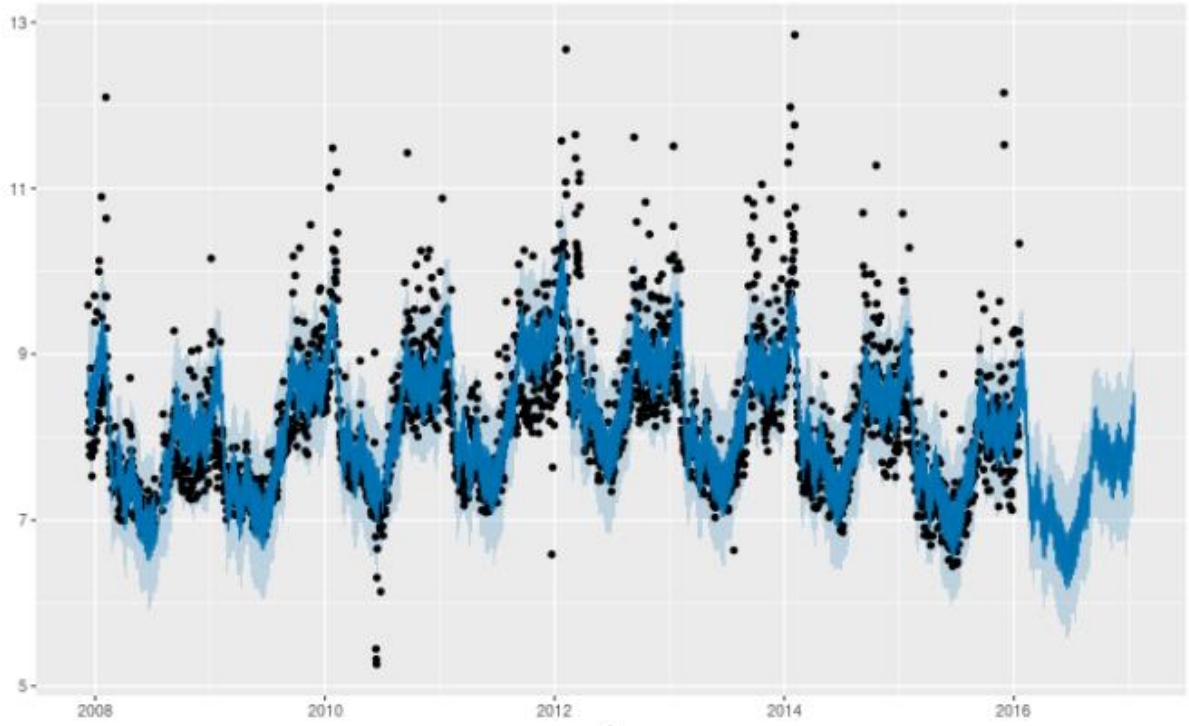

Рисунок 2.3 – щорічна сезонність гравця Peyton Manning

Prophet надасть графік компонентів зображених на рисунку 2.4, який графічно описує модель, в яку він вписується:

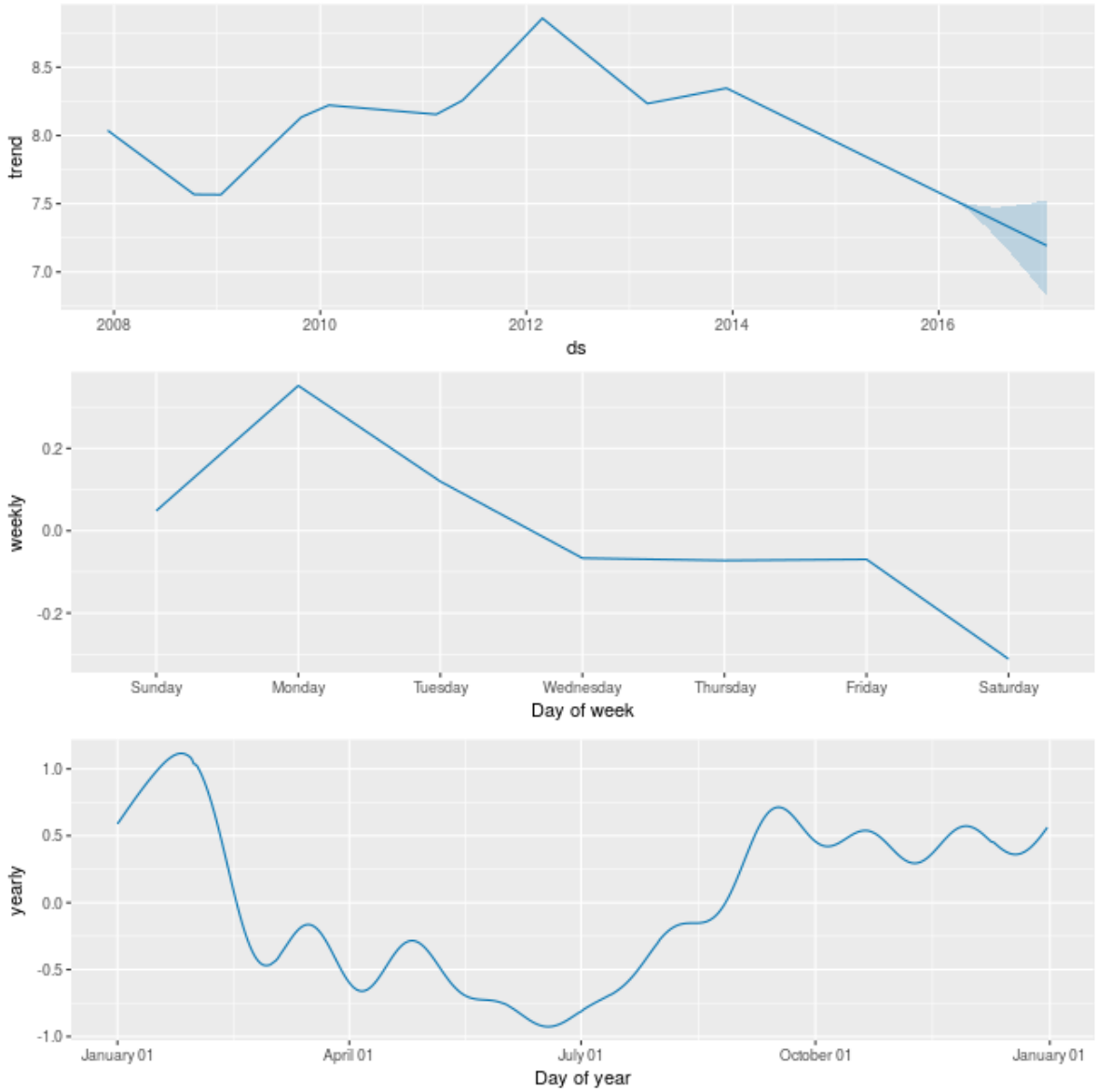

Рисунок 2.4 – графік компонентів

Ці графіки більш наочно показують річну сезонність, пов'язану з переглядом сторінки Peyton Manning (футбольний сезон і плей-офф), а також щотижневу сезонність: більше відвідувань в день і після ігор (неділя і понеділок). Також можна помітити спадну коригування трендового складової з тих пір, як він недавно вийшов на пенсію.

Важливою ідеєю в "Пророка" є те, що, роблячи більш гнучку підгонку трендової складової, ми більш точно моделюємо сезонність, і в результаті отримуємо більш точний прогноз. Дослідники вважають за краще

використовувати дуже гнучку регресійну модель (щось на зразок підгонки під криву) замість традиційної моделі часових рядів для цього завдання, оскільки вона дає нам велику гнучкість моделювання, полегшує підгонку моделі, а також більш витончено обробляє відсутні дані або відхилення.

За замовчуванням Prophet надасть інтервали невизначеності для компонента тренда, моделюючи майбутні зміни тренда в вашому часовому ряду. Якщо є бажання змоделювати невизначеність майбутніх сезонних або святкових ефектів, то можна провести кілька сотень ітерацій (що займе кілька хвилин), і прогнози будуть включати в себе оцінки сезонної невизначеності.

Модель Prophet була підігнана за допомогою мови програмування Stan і реалізували ядро процедури Пророка на імовірнісному мові програмування Stan. Stan дуже швидко (<1 секунди) проводить оптимізацію настроюваної карти для параметрів, дає нам можливість оцінити невизначеність параметрів за допомогою алгоритму Гамільтона Монте-Карло, а також дозволяє повторно використовувати процедуру підгонки на декількох мовах інтерфейсу. В даний час надається перевага реалізації Prophet як на Python, так і на R. Вони мають точно такі ж можливості.

2.5 GBDT

GBDT є ансамблевий модель дерев рішень, які навчаються послідовно. У кожній ітерації GBDT навчається деревах рішень шляхом підгонки негативних градієнтів (також відомих як залишкові помилки). Основна вартість в GBDT полягає у вивченні дерев рішень, а сама трудомістка частина в вивченні дерев рішень - це знаходження найкращих точок поділу. Одним з найпопулярніших алгоритмів пошуку точок поділу є алгоритм попереднього сортування, який перераховує всі можливі точки поділу на попередньо відсортованих значеннях ознак. Цей алгоритм простий і може знайти оптимальні точки поділу, однак він неефективний як з точки зору швидкості навчання, так і з точки зору витрат

пам'яті. Іншим популярним алгоритмом є алгоритм на основі гістограми. Замість того, щоб знаходити точки дроблення на відсортованих значеннях ознак, алгоритм, заснований на гістограмі, розбиває безперервні значення ознак на дискретні біни і використовує ці біни для побудови гістограм ознак під час навчання. Оскільки алгоритм, заснований на гістограмах, більше ефективно як за споживанням пам'яті, так і по швидкості тренування, Light GBM був розвинений на її основі.

2.6 Light GBM як частина GBDT

Дерево рішень, що підвищує градієнт (GBDT) є широко використовуваним алгоритмом машинного навчання, завдяки своїй ефективності, точності і інтерпритованості. GBDT досягає найсучасніших результатів у багатьох задачах машинного навчання, таких як класифікація по декількох класах, прогнозування клацань миші і навчання ранжирування. В останні роки, з появою великих даних (як за кількістю функцій, так і за кількістю тестових даних), GBDT стикається з новими викликами, особливо в області компромісу між точністю і ефективністю. Традиційні реалізації GBDT повинні для кожної функції сканувати всі екземпляри даних для оцінки інформаційного виграшу за всіма можливими точкам поділу. Тому їх обчислювальна складність буде пропорційна як кількості функцій, так і кількості примірників. Це робить ці реалізації дуже трудомісткими при роботі з великими даними.

Для вирішення цього завдання простою ідеєю є зменшення кількості тестових даних і кількості функцій. Однак це виявляється вкрай нетривіальним. Наприклад, незрозуміло, як виконувати вибірку даних для GBDT. Незважаючи на те, що є деякі роботи, в яких вибірка даних відповідно до їх вагами прискорює тренувальний процес форсування, вони не можуть бути безпосередньо застосовані до GBDT, оскільки в GBDT взагалі немає ваги вибірки.

Градієнтна одностороння вибірка. Хоча в GBDT немає вбудованої ваги для екземпляра даних, можна помітити, що екземпляри даних з різними градієнтами грають різну роль в обчисленні отримання інформації. Зокрема, відповідно до визначення посилення інформації, ті екземпляри даних з великими градієнтами (тобто недостатньо підготовлені екземпляри) будуть робити більший внесок в посилення інформації.

Тому при подальшій вибірці даних, щоб зберегти точність оцінки посилення інформації, необхідно краще залишати ці екземпляри з великими градієнтами (наприклад більшими, ніж попередньо задані значення порогу, або серед верхніх значень), і тільки випадковим чином опускати ті екземпляри з малим зменшенням.

Варто зазначити, що такий метод може призвести до більш точній оцінці коефіцієнта посилення, ніж рівномірно випадкова оцінка. вибірка з тієї ж цільової частотою дискретизації, особливо в тих випадках, коли значення коефіцієнта отримання інформації має велике значення.

Ексклюзивна комплектація (EFB). Зазвичай в реальних додатках, хоча їх велику кількість. Особливості, простір елементів досить мізерний, що дає нам можливість проектувати практично беззбитковий підхід до скорочення кількості ефективних функцій. Зокрема, в розрідженому просторі функцій, багато функцій є (майже) ексклюзивними, тобто вони рідко беруть ненульові значення одночасно. Приклади включати одноразові функції (наприклад, відображення одноразових слів в процесі пошуку тексту). Такі функції можемо безпечно об'єднувати такі ексклюзивні функції. Для цього є ефективний алгоритм, який скорочує оптимальні зв'язування завдання з завданням розфарбовування графа (приймаючи властивості як вершини і додаючи ребра для кожних двух характеристики, якщо вони не є взаємовиключними), і вирішити це жадібним алгоритмом за допомогою постійного співвідношення апроксимації.

2.7 Recurrent neural networks (RNN)

Мережі RNN є нейронними мережами для послідовних даних - таким чином можна застосовувати їх до часових рядів. Основна ідея рекурентних нейромереж полягає у використанні не тільки вхідних даних, але і попередніх виходів для складання поточного прогноза. Ця ідея має великий сенс - є можливість побудувати нейронні мережі, що передають значення вперед в часі. Однак такі прості рішення, як правило, працюють не так, як очікувалося. Їх складно тренувати і вони забудькуваті. Швидше, є необхідність в системі з деякою пам'яттю.

Є дві популярні і ефективні RNN моделі, які дійсно добре працюють: довгострокова короткочасна пам'ять і закритий рекурентний блок.

2.8 LSTM

Довга короткочасна пам'ять - це закритий блок пам'яті для нейронних мереж. Він має 3 ворота, які керують вмістом пам'яті. Ці ворота є простими логістичними функціями зважених сум, де ваги можуть бути вивчені шляхом зворотного розповсюдження. Це означає, що, навіть якщо це здається трохи складним, LSTM прекрасно вписується в нейронну мережу і процес її навчання. Він може дізнатися, що йому потрібно дізнатися, згадати, що йому потрібно згадувати, і згадувати те, що йому потрібно, без спеціальної підготовки і оптимізації. Вхідні ворота (1) і ворота "забуття" (2) керують осередком. стан (4), яке є довготривалою пам'яттю. Вихідний затвор (3) виробляє вихідний вектор або прихований стан (5), який є пам'яттю, сфокусованої для використання. Ця система пам'яті дозволяє мережі запам'ятовувати на довгий час, що сильно не вистачає ванільним (стандартним) рекурентним нейронним мережам.

$$i_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = sigmoid(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \odot tanh(c_t) \quad (5)$$

2.9 GRU

Закріплена рекуррентная одиниця по суті є спрощеним LSTM. Він має точно таку ж роль в мережі. Основна відмінність полягає в кількості воріт і вага - GRU дещо простіше. У нього 2 воріт. Окільки у нього є немає вихідного шлюзу, немає контролю над вмістом пам'яті.

Оновлення затвора (6) управляє потоком інформації від попередньої активації, і додавання нової інформації також (8), поки вставлені ворота скидання (7). в активацію кандидата. В цілому, це досить схоже на LSTM. З них Одні тільки відмінності, важко сказати, який з них є кращим вибором для заданої проблеми.

$$x_t = sigmoid(W_z x_t + U_z h_{t-1} + b_z) \quad (6)$$

$$r_t = sigmoid(W_r x_t + U_r h_{t-1} + b_r) \quad (7)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (8)$$

2.10 Рекурентні нейронні мережі для прогнозування

Хоча це, ймовірно, не є їх основним застосуванням, мережі LSTM і GRU часто використовуються для прогнозування часових рядів з воротами для вивчення часових відстаней. Використані стекові мережі LSTM для виявлення аномалій у часових рядах. запропонували метод адаптивного градієнтного

навчання для мережі, щоб робити надійні прогнози для часових рядів з відхиленнями і точками зміни.

Включено автокодирувальник в LSTM для поліпшення його продуктивності по прогнозуванню. Запропопоновано механізм підвищеної уваги до захоплювати періоди і моделювати відсутні значення у часових рядах. для конволюціонних ЛСТМ з використанням бутстраппінга.

2.11 Прогнозні показники ефективності

Після ознайомлення з різними моделями прогнозування часових рядів виникає наступне важливе питання - реалізація, тобто застосування цих методів для складання прогнозів. При застосуванні конкретної моделі до якогось реального чи імітованої часового ряду, спочатку необроблені дані діляться на дві частини, а саме: тренувальний і тестовий набори. Спостереження в навчальному комплекті використовуються для побудови бажаної моделі. Часто невелика частина навчального комплекту зберігається для цілей валідації і відома як валідаційні комплект. Іноді препроцесинг виконується шляхом нормалізації даних, логарифмічних або інших перетворень. Одним з таких відомих методів є метод Box-Cox Transformation. Як тільки модель побудована, вона використовується для створення прогнозів. Спостереження тестового набору проводяться для перевірки точності передбачення цих значень за допомогою відповідної моделі. При необхідності на прогнозовані значення застосовується зворотне перетворення для їх перетворення у вихідну шкалу. Для того, щоб оцінювати точність прогнозування конкретної моделі або для оцінки і порівняння різних моделей розглядається їх відносна продуктивність в наборі тестових даних.

У зв'язку з фундаментальної важливістю прогнозування часових рядів у багатьох практичних ситуаціях, при виборі конкретної моделі слід проявляти

належну обережність. З цієї причини, різні показники для оцінки роботи пропонуються в літературі для оцінки прогнозу і порівняти різні моделі. Вони також відомі як показники продуктивності. Кожна з цих перевірок є функцією фактичних і прогнозованих значень часового ряду.

Тепер варто обговорити дані показники ефективності і їх важливі властивості. У кожному з наступних визначень y_t- фактичне значення, f_t – прогнозоване,

$$e_t = y_t - f_t$$

значення похибка прогнозу і n - розмір тестового набору. Також,

$$\underline{y} = \frac{1}{n}\sum_{t=1}^{n} \quad y_t$$

середнє значення та

$$\sigma^2 = \frac{1}{n-1}\sum_{t=1}^{n} \quad (y_t - \underline{y})^2$$

тестова дисперсія.

Помилка середнього прогнозу (The Mean Forecast Error (MFE))

$$MFE = \frac{1}{n}\sum_{t=1}^{n} \quad e_t$$

Властивості MFE такі:

- це міра середнього відхилення прогнозних значень від фактичних;
- він показує напрямок похибки і тому також називається зміщенням прогнозу;
- у MFE вплив позитивних і негативних похибок зводиться нанівець, і немає ніякої можливості точно знати їх кількість;

- нульовий MFE не означає, що прогнози ідеальні, тобто не містять помилок; скоріше, він тільки вказує на те, що прогнози націлені на потрібну мету;
- MFE не виводить на панель екстремальні помилки;
- залежить від масштабу виміру, а також залежить від трансформації даних;
- для хорошого прогнозу, тобто для забезпечення мінімальної похибки, бажано, щоб MFE був якомога ближче до нуля, наскільки це можливо.

Абсолютна помилка (The Mean Absolute Error (MAE))

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t|$$

Властивості MAE такі:

- вимірює середнє абсолютне відхилення прогнозованих значень від вихідних.
- також називається Середнім абсолютним відхиленням (MAD).
- показує величину загальної помилки, що виникла в результаті прогнозування.
- у MAE вплив позитивних і негативних помилок не зводиться нанівець.
- на відміну від MFE, MAE не дає уявлення про направлення помилок.
- для хорошого прогнозу отримане значення MAE має бути якомога менше.
- як і MFE, MAE також залежить від масштабу вимірювань і перетворення даних.
- екстремальні помилки прогнозу не відображаються в панелі MAE.

Середня абсолютна процентна похибка (The Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left|\frac{e_t}{y_t}\right| \times 100$$

Властивості MAPE такі:

- цей показник являє собою відсоток від середньої абсолютної похибки, що мала місце;
- він не залежить від масштабу виміру, але схильний до впливу трансформації даних;
- він не показує напрямок помилки;
- панелі MAPE не призначені для панелей з екстремальними відхиленнями;
- при цьому зустрічні знакові похибки не компенсуються одна одною.

Середня відсоткова похибка (The Mean Percentage Error (MPE))

$$MPE = \frac{1}{n}\sum_{t=1}^{n} \left(\frac{e_t}{y_t}\right) \times 100$$

Властивості MPE такі:

- MPE є відсоток від середньої помилки, що виникла при прогнозуванні;
- він має ті ж властивості, що й MAPE, за винятком того, що показує напрямок помилки;
- протилежні помилки впливають один на одного і нівелюють одна одну;
- таким чином, як і MFE, отримуючи значення MPE близьке до нуля, ми не можемо зробити висновок, що відповідна модель працює дуже добре;
- бажано, щоб для хорошого прогнозу отриманий MPE був невеликим.

Середньоквадратична похибка (The Mean Squared Error (MSE))

$$MSE = \frac{1}{n}\sum_{t=1}^{n} {e_t}^2$$

Властивості MSE такі:

- це міра середнього квадратного відхилення прогнозованих значень;
- оскільки тут протилежні знакові помилки не компенсують один одного, MSE дає загальне уявлення про те чи виникли помилки під час прогнозування;

- у ньому відображаються екстремальні помилки, що виникають при прогнозуванні;

- MSE підкреслює той факт, що на загальну похибку прогнозування в значній мірі впливають такі фактори як індивідуальні помилки, тобто великі помилки коштують набагато дорожче дрібних помилок;

- MSE не дає уявлення про направлення загальної помилки;

- MSE чутливо реагує на зміну масштабу і трансформацію даних.

Хоча MSE - хороший показник загальної похибки прогнозування, але не такий інтуїтивно зрозумілий і зрозумілий і щоб його було легко інтерпретувати як інші приклади, що обговорювалися раніше.

Сума квадратичної помилки (The Sum of Squared Error (SSE))

$$SSE = \sum_{t=1}^{n} \quad {e_t}^2$$

Властивості SSE такі:

- він вимірює сумарне квадратичне відхилення прогнозованих спостережень від фактичних значень;

- властивості SSE такі ж, як і у MSE.

Підписана середня квадратична помилка (The Signed Mean Squared Error (SMSE))

$$SMSE = \frac{1}{n}\sum_{t=1}^{n} \quad \left(\frac{e_t}{|e_t|}\right) {e_t}^2$$

Властивості SMSE такі:

- це те ж саме, що і MSE, за винятком того, що тут зберігається вихідний знак для кожної окремої квадратної помилки;

- SMSE систематизує екстремальні помилки, що виникають при прогнозуванні;

- на відміну від MSE, SMSE також показує напрямок загальної помилки;

- при розрахунку SMSE позитивні і негативні помилки компенсують один одного;
- як і MSE, SMSE також чутливі до змін масштабу і трансформації даних;

Коренева середня квадратична помилка (The Root Mean Squared Error (RMSE))

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} \quad {e_t}^2}$$

Властивості RMSE такі:

- RMSE - це не що інше, як квадратний корінь обчисленого MSE;
- всі властивості MSE зберігаються і для RMSE;

Нормалізована середня квадратична похибка (The Normalized Mean Squared Error (NMSE))

$$NMSE = \frac{MSE}{\sigma^2} = \frac{1}{\sigma^2 n}\sum_{t=1}^{n} \quad {e_t}^2$$

Властивості NMSE такі:

- NMSE нормалізує отримане значення MSE після поділу його на дисперсію тесту;
- це виважена міра похибки і дуже ефективна при оцінці точності прогнозу моделі;
- чим менше значення NMSE, тим краще прогноз;
- інші властивості NMSE такі ж, як і у MSE.

U-статистика Тейла (The Theil’s U-statistics)

$$U = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n} \quad e_t^2}}{\sqrt{\frac{1}{n}\sum_{t=1}^{n} \quad f_t^2}\sqrt{\frac{1}{n}\sum_{t=1}^{n} \quad y_t^2}}$$

Властивості U-статистики Тейлора

- це нормалізована міра загальної похибки прогнозу;
- $0 \leq U \leq 1$, $U = 0$ середня арифметична ідеально підходить;
- на цей метод впливає зміна масштабу і трансформація даних;
- для оцінки хорошою точності прогнозу бажано, щоб U-статистика була близька до нуля.

Представлено десять важливих показників для оцінки точності підігнаної моделі. Кожен з цих методів має унікальні властивості, відмінні від інших. В експериментах краще розглядати більш ніж один критерій ефективності. Це допоможе отримати розумні знання про величину, напрямок загальної помилки прогнозу. З цієї причини для часових рядів аналітики зазвичай використовують більше одного виду методів для винесення суджень.

Висновок до розділу 2

До сих пір ми обговорювали важливі мережеві архітектури, а саме: RNN, LSTM та GRU які широко використовуються для прогнозування часових рядів. Після визначення конкретної структури наступним найбільш важливим питанням є визначення оптимальних мережевих параметрів. Кількість нейронних параметрів дорівнює загальній кількості зв'язків між нейронами і термінами зміщення.

Бажана модель повинна давати відносно невелику помилку не тільки в межах вибірки (тренувальної), але також і за вибірковими (тестовим) даними. З цієї причини величезна обережність проявляється в тому, що необхідно

правильно обирати правильне налаштування моделі (як наприклад кількість прихованих шарів нейронів для нейронної мережі). Однак, для прикладу з кількістю прихованих шарів нейронів, це непросте завдання. оскільки для вибору цих параметрів відсутні теоретичні вказівки, а часто і самі ці параметри.

Також були розглянуті методи машинного навчання GBDT та Light GBM. Було визначено, що для цілей часових рядів найкраще підходить модель дерев рішень Light GBM.

З цією метою проводяться експерименти, такі як перехресна валідація. Інша серйозна проблема полягає в тому, що неадекватне або велика кількість параметрів може привести до перенавчання даних. Перетренування дає помилково хороший результат всередині тестової вибірки, що не дає кращого прогнозу. Щоб покарати додавання додаткових параметрів, необхідно виконати наступні дії, як наприклад порівняння моделей за допомогою AIC і BIC. Мережева обрізка і в цьому відношенні вельми популярний також байесовский алгоритм регуляризації МакКей.

Підводячи підсумок, можна сказати, що машинне навчання - дивно просте, але потужні технології для часу. серійне прогнозування. Вибір відповідних параметрів має вирішальне значення при використанні моделей для прогнозування. Крім того, відповідне перетворення або масштабування даних тренування часто виконується в такий спосіб необхідні для досягнення найкращих результатів.

РОЗДІЛ 3 АРХІТЕКТУРА ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

3.1 Короткий огляд

Майже кожна проблема часового ряду матиме деякі зовнішні функції або деякі внутрішні функції, щоб допомогти моделі. Додамо деякі базові функції, такі як значення запізнювання доступних числових функцій, які широко використовуються в задачах прогнозування. Оскільки нам потрібно передбачити ціну акцій на день, ми не можемо використовувати значення функцій в той же день, оскільки вони будуть недоступні в реальному часі виведення. Нам необхідно використовувати такі статистичні дані, як середнє, стандартне відхилення їх запізнілих значень. Будемо використовувати три набори запізнілих значень, один попередній день, один - 7 днів і інший - 30 днів в якості проксі-сервера для метрик за минулий тиждень і останній місяць. Для підвищення ефективності моделей дуже корисно додати в них функції, пов'язані з датою, такі як годину, день, місяць, в залежності від обставин, для надання інформації про часову складову в даних моделі. Для моделей часових рядів явно передавати цю інформацію не потрібно, але це можна зробити для того, щоб всі моделі порівнювалися на одному і тому ж наборі функцій.

Для роботи було обрано датасет nifty-50. Індекс NIFTY 50 є базовим широким індексом фондового ринку Індії для індійського фондового ринку. NIFTY 50 позначає Національний індекс Fifty і являє собою середньозважене значення 50 акцій індійських компаній в 17 секторах. Це один з двох основних фондових індексів, які використовуються в Індії, другий - BSE Sense. Реальні значення які й будуть використовувати для прогнозування є VWAP зображено на рисунку 3.1.

Поділ даних на тестову і валідаційну вибірку.

тестова: Дані з 26 травня 2008 року по 31 грудня 2018 року.

валідаційна: Дані з 1 січня 2019 року по 31 грудня 2019 року.

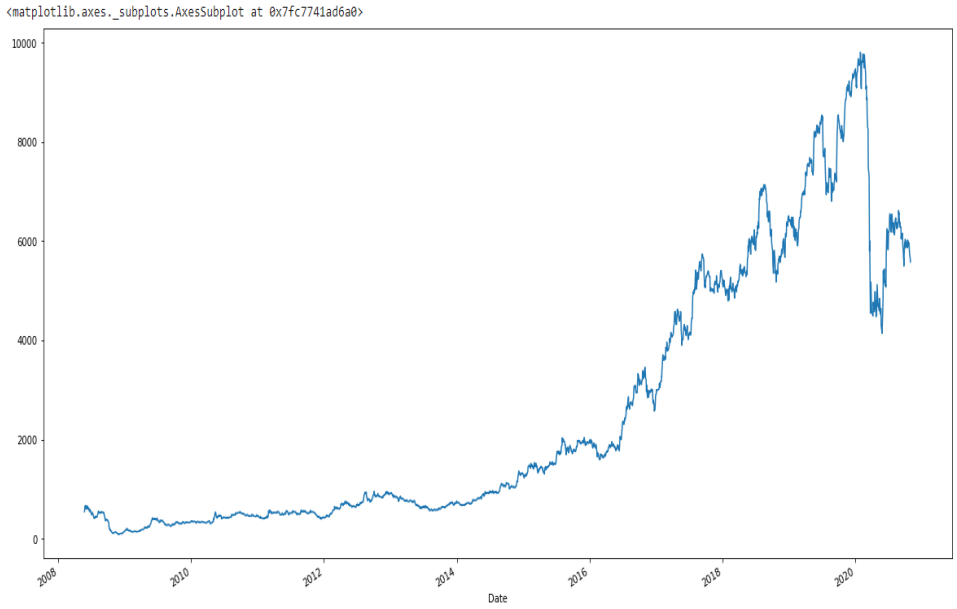

Рисунок 3.1 – складова VWAP акцій Bajaj Finsv на проміжку 2000-2020 роки

3.2 Огляд застосованої архітектури Keras

Keras користується бібліотекою Theano або TensorFlow для ефектичних розрахунків з тезнорами. Що таке тензор? Це просто багатовимірний масив або матриця. Обидві бібліотеки вміють ефективно виконувати символічні розрахунки з тензорами, а це основний будівельний блок для створення нейронних мереж.

Регуляризація надзвичайно добре розроблена в даній архітектурі. Її ціль – унеможливити перенавчання. В шарах різних типів маються параметри регуляризації. Наведено часто використовувані в модулях функції:

1. kernel_regularizer: застосовується до матриці ваг;
2. bias_regurizer: застосовується до векторів зміщення;
3. activity_regularizer: застосовується до вихідного шару (його функції активації).

Контрольні точки – це процес періодичного збереження миттєвого знімка стану додатку, так щоб програму модна було перезапускати з останнього

збереженого стану за умови відмови. Це може бути корисним при навчанні глибинних моделей навчання в будь-який момент часу.

3.3 Прогнозування за допомогою АРКС

Виконаємо тест Дікі-Фулера для перевірки стаціонарності часового ряду. Результати зображено на рисунку 3.2

```
adf:  -1.1425022139058953
p-value:  0.6978284856133512
Critical values:  {'1%': -3.432495840047687, '5%': -2.862488095901948, '10%': -2.567274695404461}
Unit roots exist, time series is not stationary
```

Рисунок 3.2 – результати перевірки теста Дікі-Фулера

Можна зробити висновок, що тест знайшов одиничні корені, а отже ряд не є стаціонарним.

Отже, необхідно виконати мультиплікативну декомпозицію ряду і таким чином зробити дослідження про наявність тренду, сезонність та викиди. Відповідні графіки розміщені на рисунку 3.3

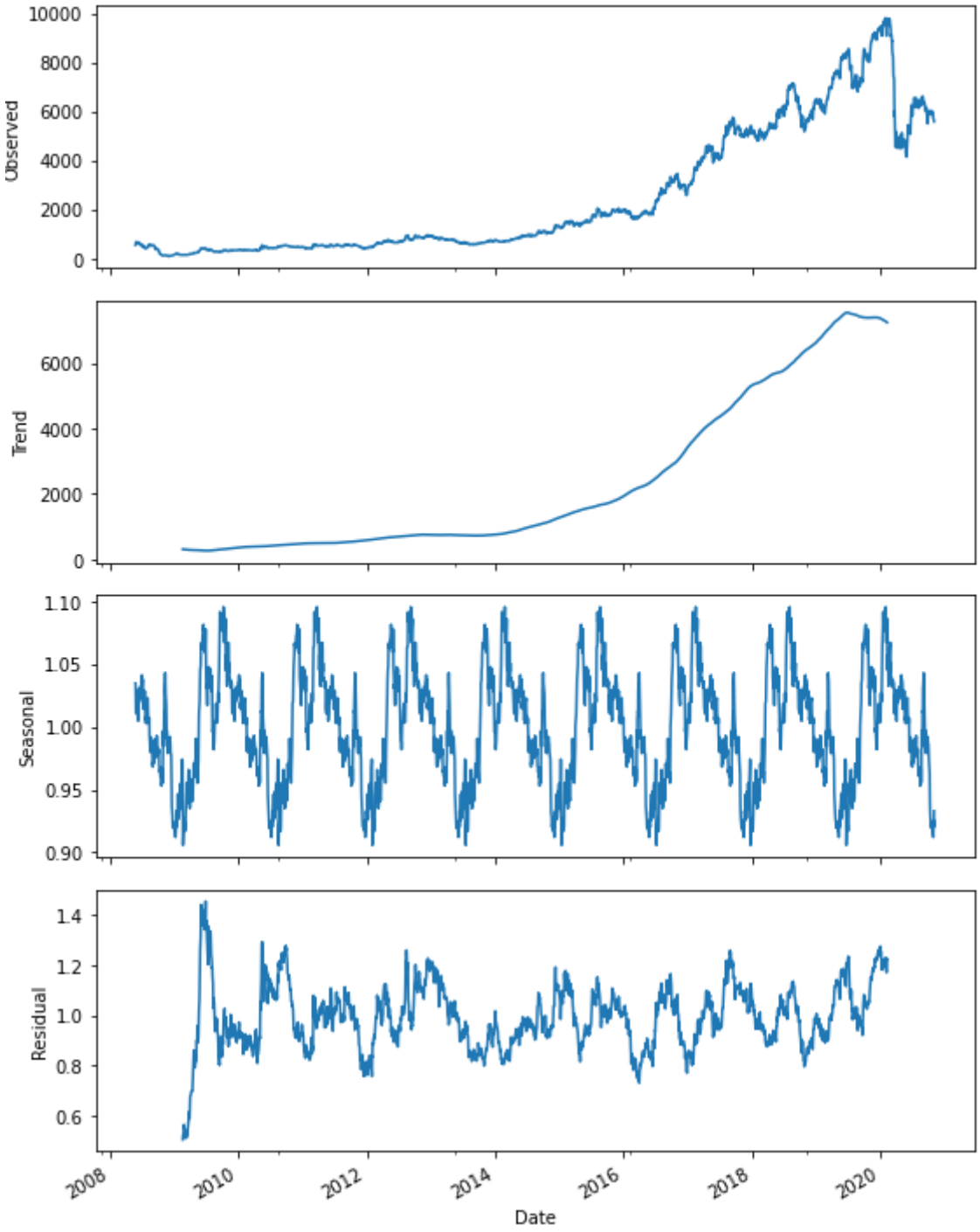

Рисунок 3.3 – Тренд, сезонність, викиди експерементального ряду

Бачимо, що у нашому ряді явно присутній лінійний тренд. Розкид сезонної та шумової компонент не є великим, середнє значення цих компонент дорівнює 1, отже їх вплив на ряд є досить слабким.

Для моделювання часового ряду за допомогою АРКС моделі, нам необхідно зробити перетворення ряду, щоб він став стаціонарним. Для цього візьмемо логарифм від нього і далі порахуємо різниці порядку 1. Після цього експерементальний ряд має такий вигляд, який можна побачити на рисунку 3.4

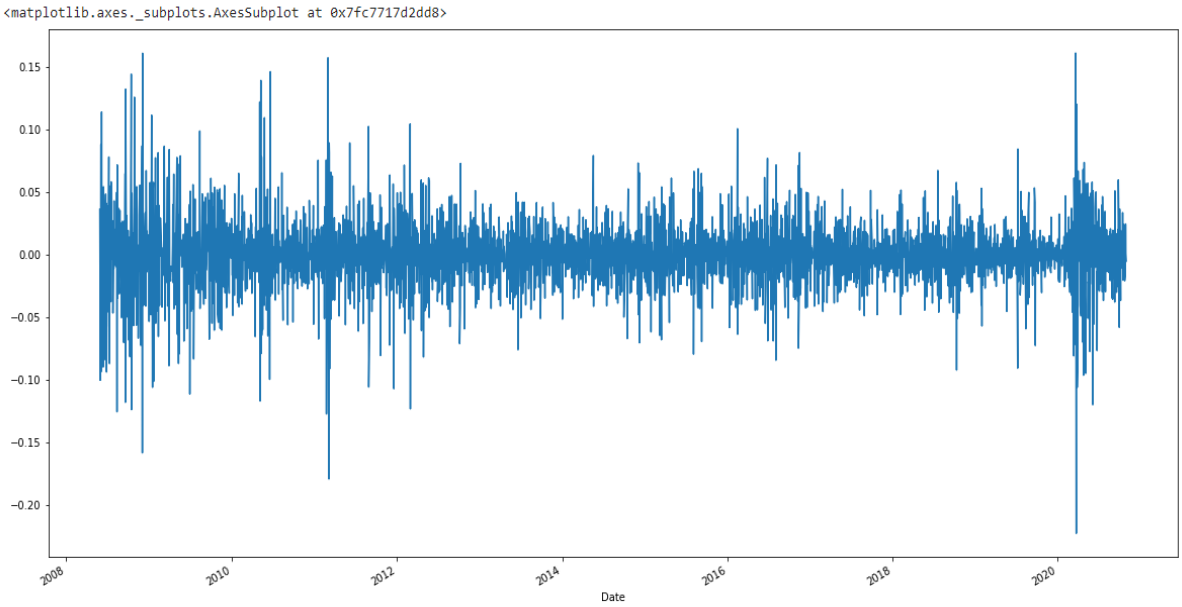

Рисунок 3.4 – графік часового ряду, після перетворення

Зробимо повторну перевірку на стаціонарність – проведемо тест Дікі Фулера. Результати можна побачити на рисунку 3.5

```
adf:  -17.029805061083753
p-value:  8.341004408083955e-30
Critical values:  {'1%': -3.432497249267698, '5%': -2.862488718312822, '10%': -2.5672750267676925}
Unit roots doesn't exist, time series is stationary
```

Рисунок 3.5 – результати тесту Дікі Фулера

Бачимо, що все гаразд і можна застосовувати АРКС модель.

Для цього необхідно коректно підібрати значення коефіцієнтів p та q, а для цього на рисунку 3.6 побудовано графіки АКФ та ЧАКФ.

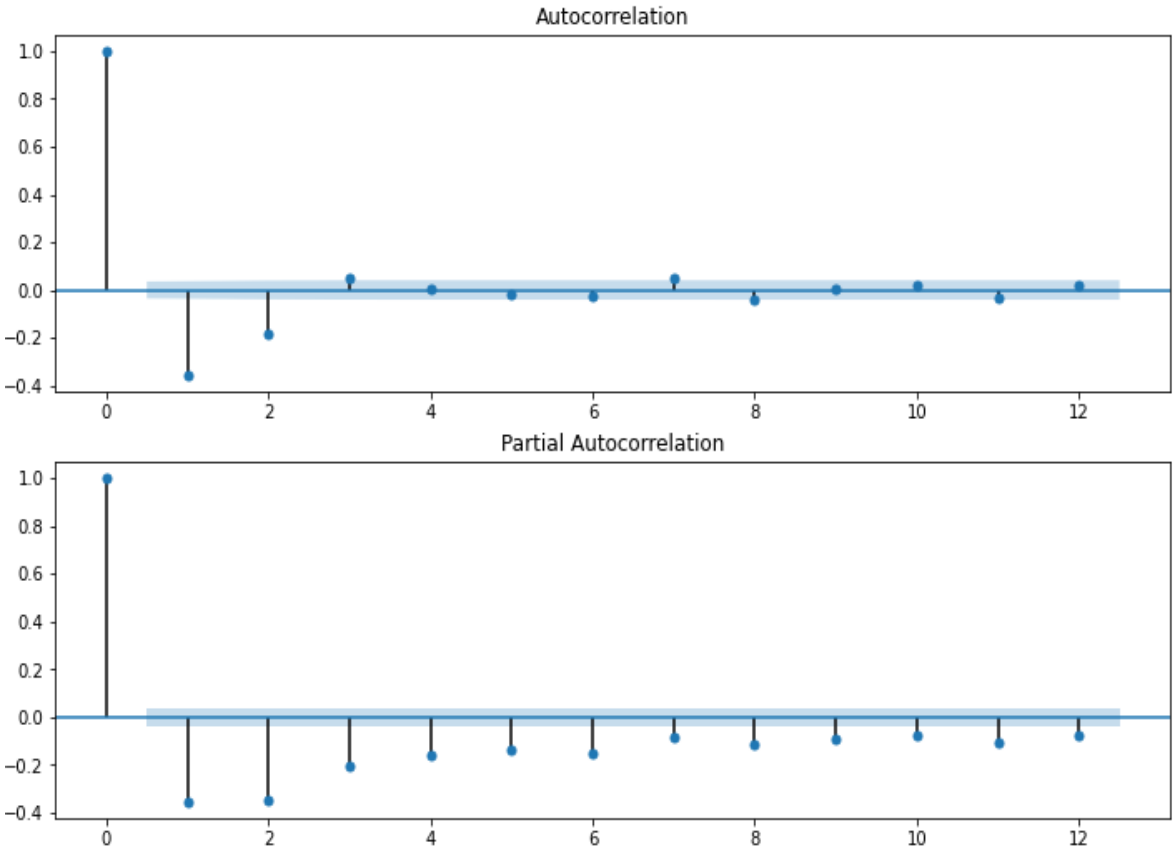

Рисунок 3.6 – графік АКФ та ЧАКФ

Візьмемо p = 2, як номер коефіцієнта, що найбільше відрізняється від 0 у ЧАКФ, та q = 3, як кількість коефіцієнтів, що найбільше відрізняються від 0 у АКФ.

Результати АРКС(2, 3) моделі:

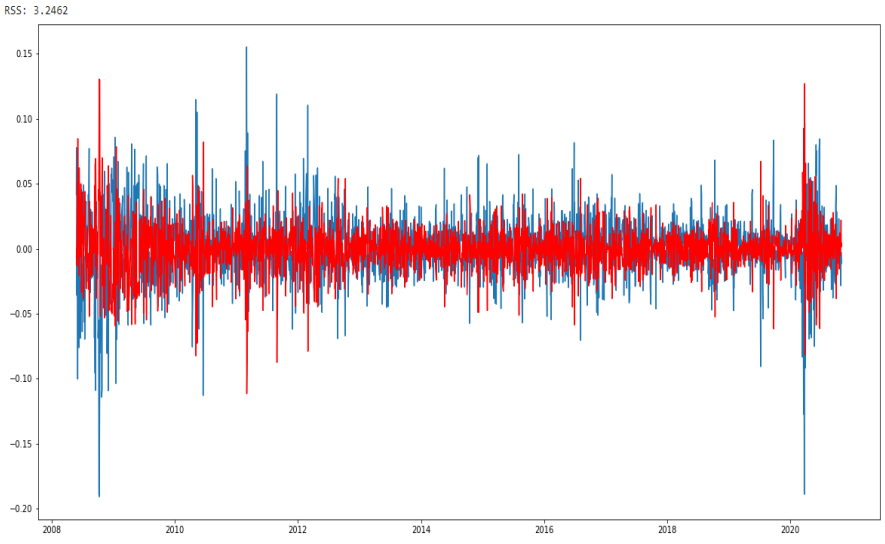

Рисунок 3.7 – результати моделі АРКС(2, 3)

Результуюча модель АРКС виглядає наступним чином – рисунок 3.8

ARMA Model Results

Dep. Variable:	VWAP	**No. Observations:**	3077
Model:	ARMA(2, 3)	**Log Likelihood**	7267.834
Method:	css-mle	**S.D. of innovations**	0.023
Date:	Thu, 03 Dec 2020	**AIC**	-14521.667
Time:	20:57:54	**BIC**	-14479.445
Sample:	0	**HQIC**	-14506.500

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.795e-07	7.76e-07	0.360	0.719	-1.24e-06	1.8e-06
ar.L1.VWAP	0.7991	0.017	46.054	0.000	0.765	0.833
ar.L2.VWAP	0.0596	nan	nan	nan	nan	nan
ma.L1.VWAP	-1.5261	nan	nan	nan	nan	nan
ma.L2.VWAP	0.2902	nan	nan	nan	nan	nan
ma.L3.VWAP	0.2359	nan	nan	nan	nan	nan

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.1523	+0.0000j	1.1523	0.0000
AR.2	-14.5496	+0.0000j	14.5496	0.5000
MA.1	1.0000	+0.0000j	1.0000	0.0000
MA.2	1.2263	+0.0000j	1.2263	0.0000
MA.3	-3.4564	+0.0000j	3.4564	0.5000

Рисунок 3.8 – результати моделі АРКС

Після конвертації результату АРКС моделі для початкового ряду – тобто без логарфиму різниць рисунок так відповідні помилки зображені на рисунку 3.9

Рисунок 3.9 – результат моделі АРКС

3.4 Прогнозування за допомогою АРІКС

АРІКС модель може працювати з нестаціонарними рядами шляхом взяття різниць деякого порядку від початкового ряду. Для АРКС ми взяли різницю першого порядку, знайшли найкращі параметри p та q, і побудували модель. Тобто для АРІКС ми не будемо брати перший порядок d=1, оскільки ми отримаємо ідентичні результати попередньої моделі.

Тому візьмемо параметр d=2. Побудуємо для цього випадку АКФ та ЧАКФ, оберемо кращі p, q зображені АКФ та ЧАКФ на рисунку 3.10.

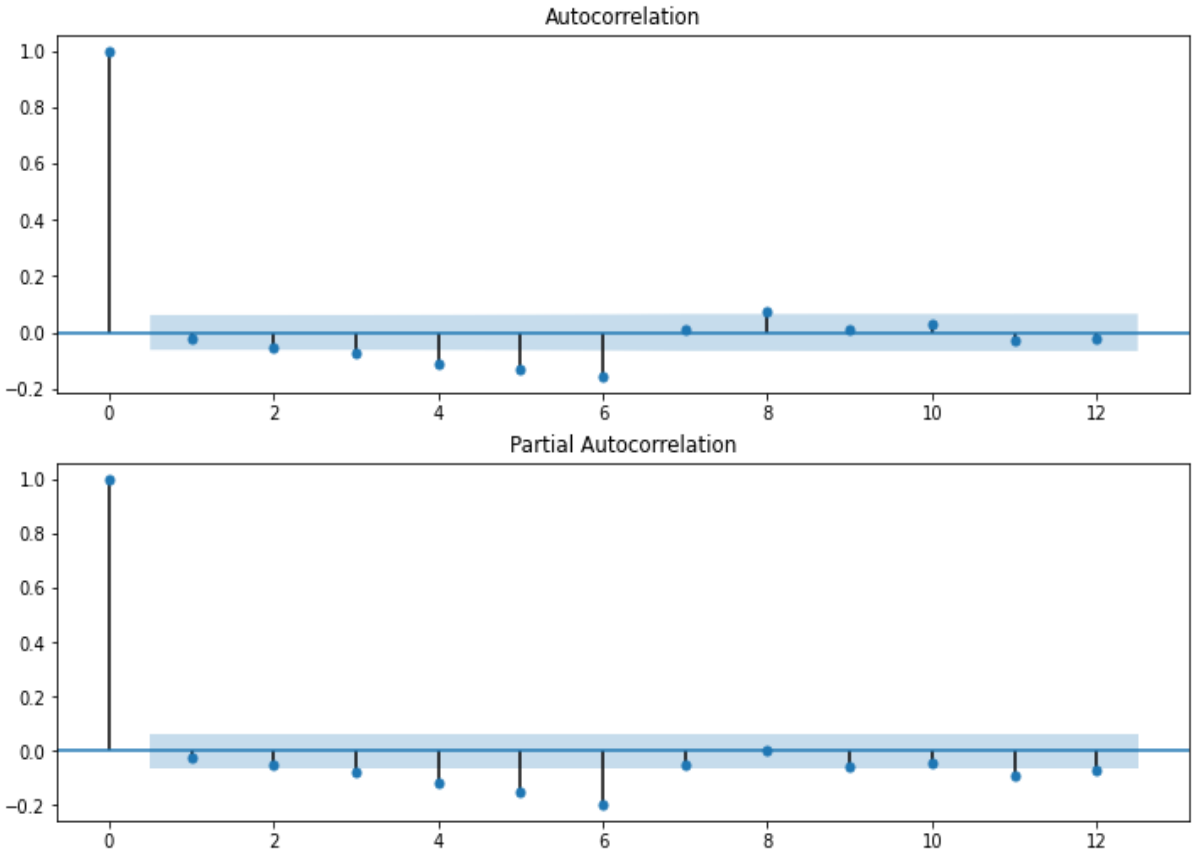

Рисунок 3.10 – графік АКФ та ЧАКФ для другого порядку

З графіку видно, що необхідно взяти p = 7, q = 6.

Значення моделі АРІКС (7, 1, 6) зображено на рисунку 3.11, а результати роботи зображено на рисунку 3.12

ARIMA Model Results

Dep. Variable:	D2.	**No. Observations:**	1003
Model:	ARIMA(7, 2, 6)	**Log Likelihood**	2935.351
Method:	css-mle	**S.D. of innovations**	0.013
Date:	Wed, 20 May 2020	**AIC**	-5840.702
Time:	15:50:48	**BIC**	-5767.041
Sample:	2	**HQIC**	-5812.710

	coef	std err	z	P>|z|	[0.025	0.975]
const	-9.518e-07	1.61e-06	-0.592	0.554	-4.1e-06	2.2e-06
ar.L1.D2.	-1.2141	nan	nan	nan	nan	nan
ar.L2.D2.	-0.6830	nan	nan	nan	nan	nan
ar.L3.D2.	-0.8530	nan	nan	nan	nan	nan
ar.L4.D2.	-1.1267	nan	nan	nan	nan	nan
ar.L5.D2.	-0.5829	nan	nan	nan	nan	nan
ar.L6.D2.	-0.0182	0.050	-0.364	0.716	-0.117	0.080
ar.L7.D2.	-0.0308	0.032	-0.963	0.336	-0.093	0.032
ma.L1.D2.	0.2250	nan	nan	nan	nan	nan
ma.L2.D2.	-0.5325	0.068	-7.790	0.000	-0.666	-0.398
ma.L3.D2.	0.1974	nan	nan	nan	nan	nan
ma.L4.D2.	0.2774	nan	nan	nan	nan	nan
ma.L5.D2.	-0.5704	0.060	-9.569	0.000	-0.687	-0.454
ma.L6.D2.	-0.5878	nan	nan	nan	nan	nan

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.5267	-0.9034j	1.0458	-0.1660
AR.2	0.5267	+0.9034j	1.0458	0.1660
AR.3	-1.1286	-0.0000j	1.1286	-0.5000
AR.4	-0.8940	-0.7214j	1.1488	-0.3919
AR.5	-0.8940	+0.7214j	1.1488	0.3919
AR.6	0.6353	-4.4217j	4.4671	-0.2273
AR.7	0.6353	+4.4217j	4.4671	0.2273
MA.1	1.0016	-0.0000j	1.0016	-0.0000
MA.2	0.5126	-0.8919j	1.0287	-0.1670
MA.3	0.5126	+0.8919j	1.0287	0.1670
MA.4	-0.9218	-0.7358j	1.1795	-0.3928
MA.5	-0.9218	+0.7358j	1.1795	0.3928
MA.6	-1.1536	-0.0000j	1.1536	-0.5000

Рисунок 3.11 – значення моделі АРІКС

Рисунок 3.12 – результати роботи моделі APIKC (7, 1, 6)

Результуючий графік можна побачити на рисунку 3.13, де зображено фактичне порівняння роботи моделі та фактичних значень.

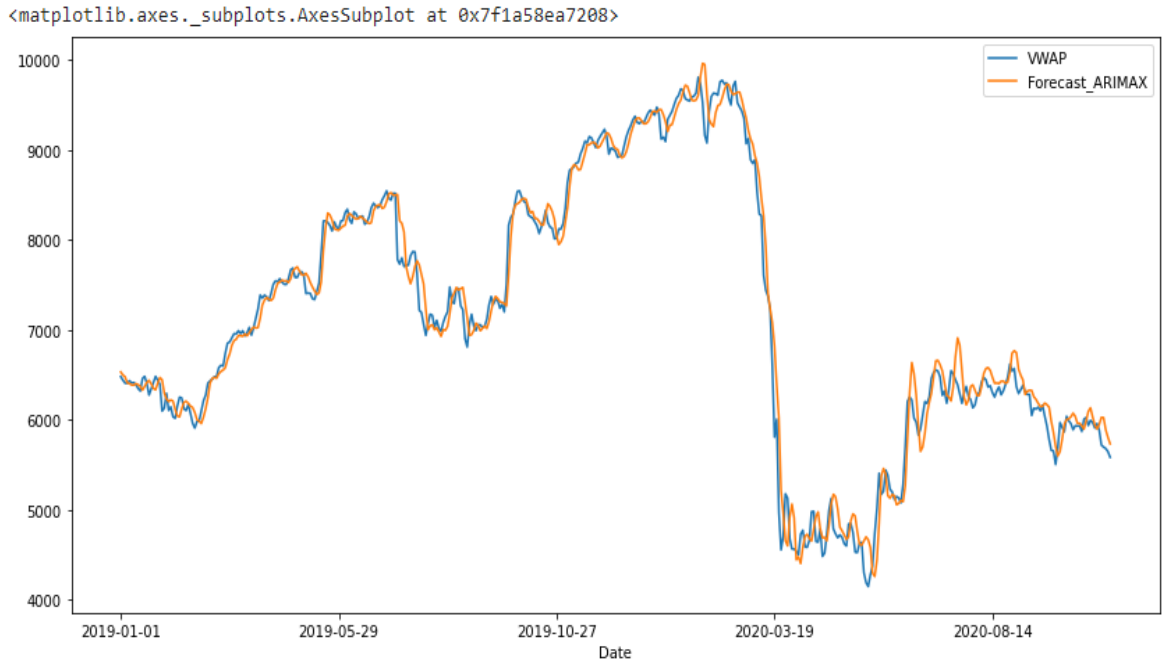

Рисунок 3.13 – порівняння реальних значень до значень моделі.

3.5 Прогнозування за допомогою Prophet

параметри за замовчуванням використовуються для Prophet. Графіки компонент які вдалося отримати за допомогою даної моделі зображені на рисунках 3.14-3.16. Порівняння прогнозу АРІКС та Prophet зображені на рисунку 3.17.

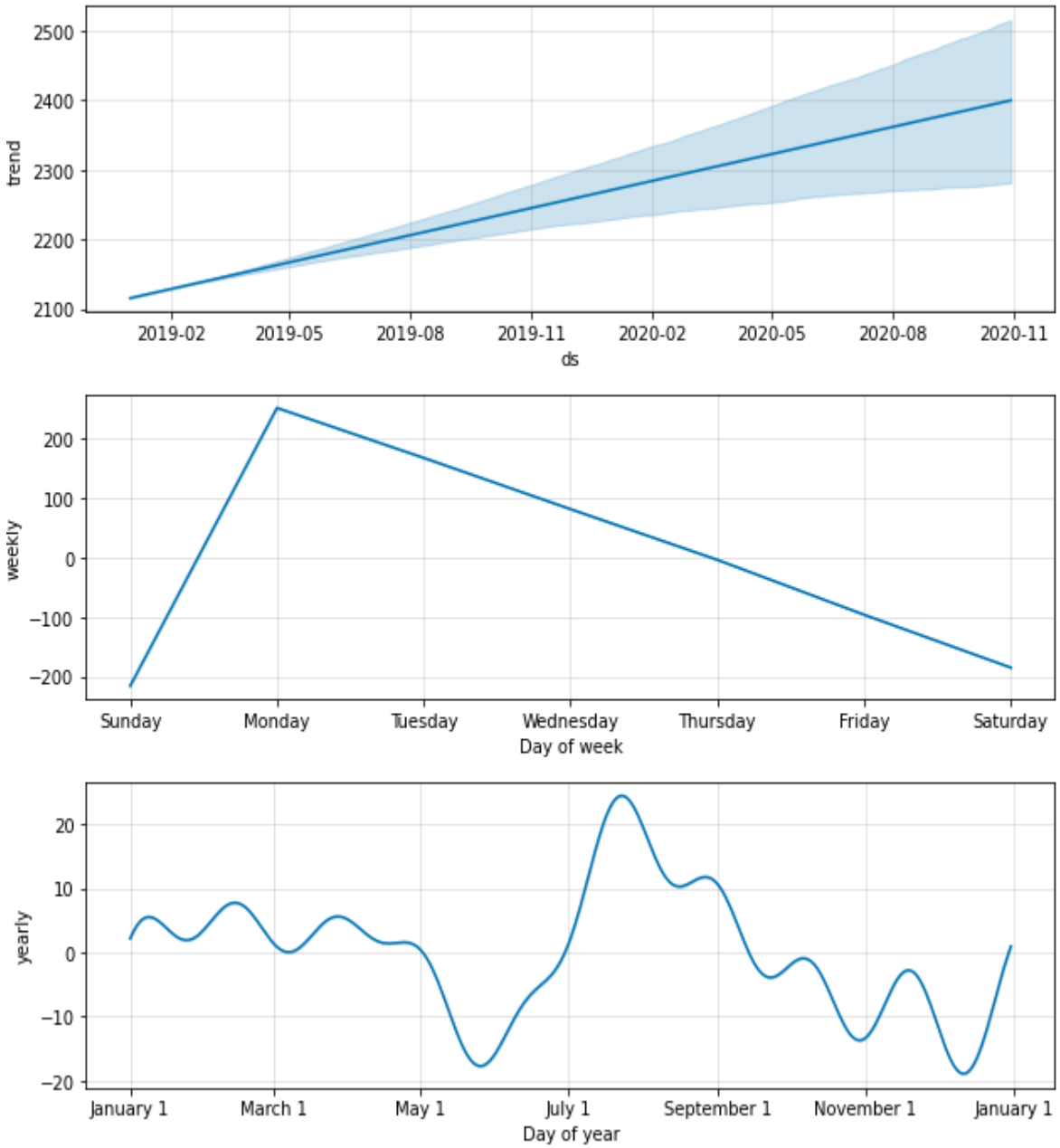

Рисунок 3.14 – графік компонент видубутий за допомогою Prophet

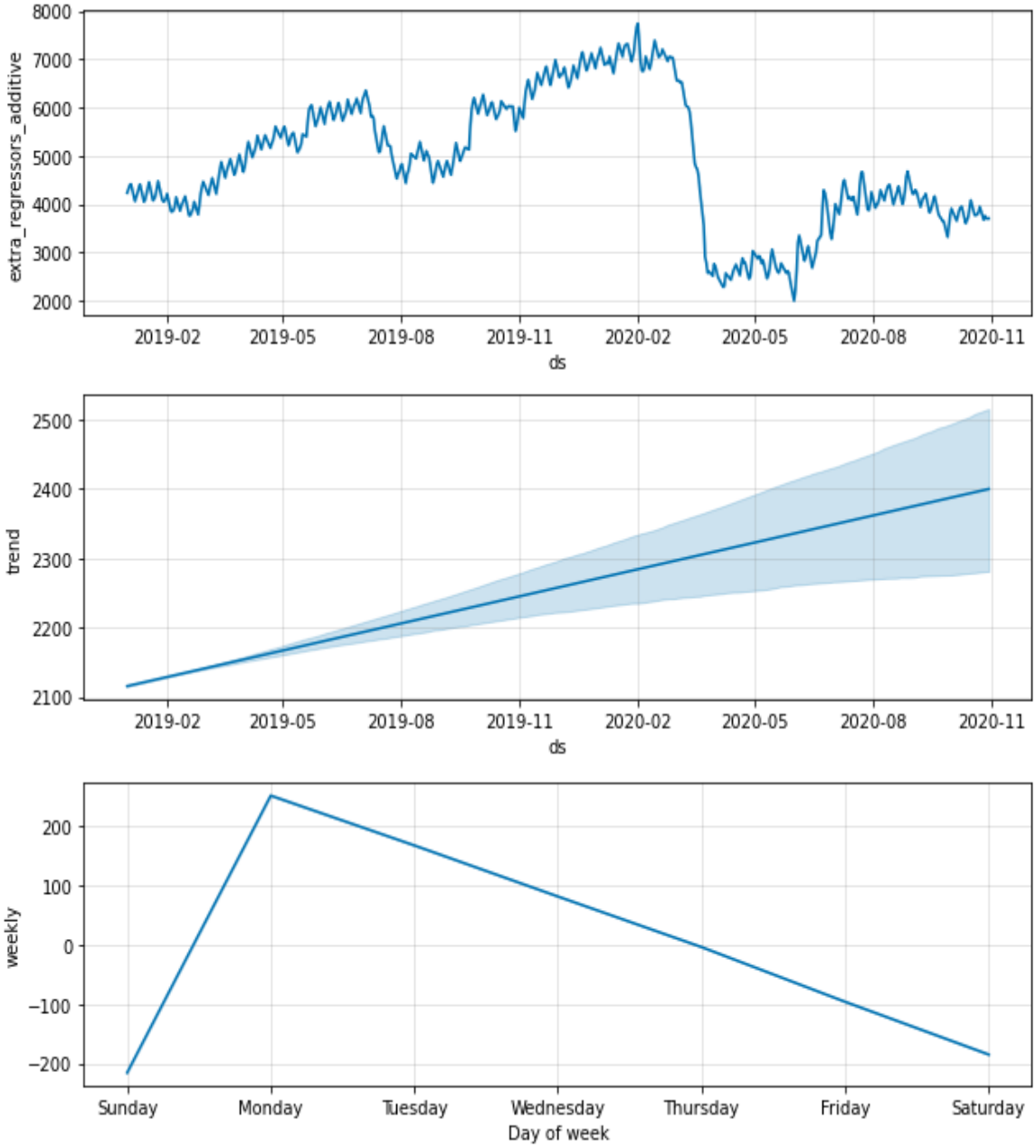

Рисунок 3.15 – графік компонент видубутий за допомогою Prophet частина 2

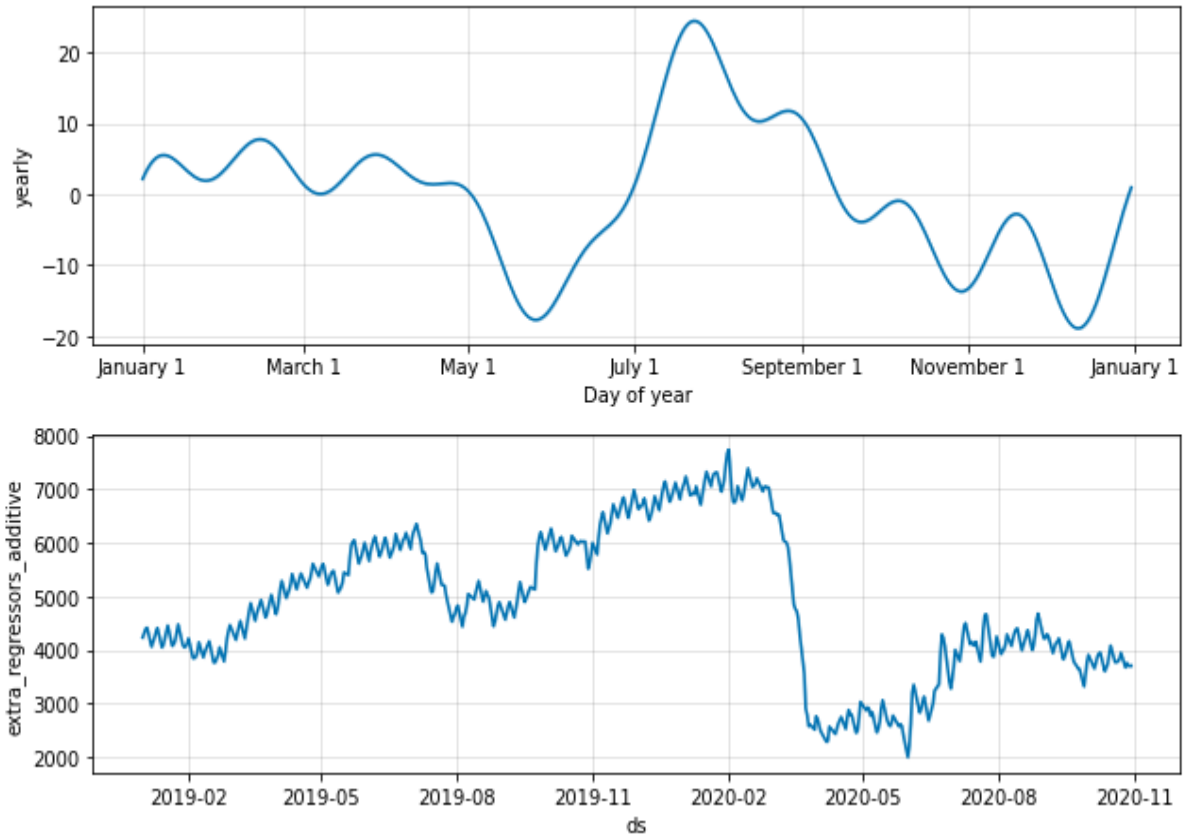

Рисунок 3.16 – графік компонент видубутий за допомогою Prophet частина 3

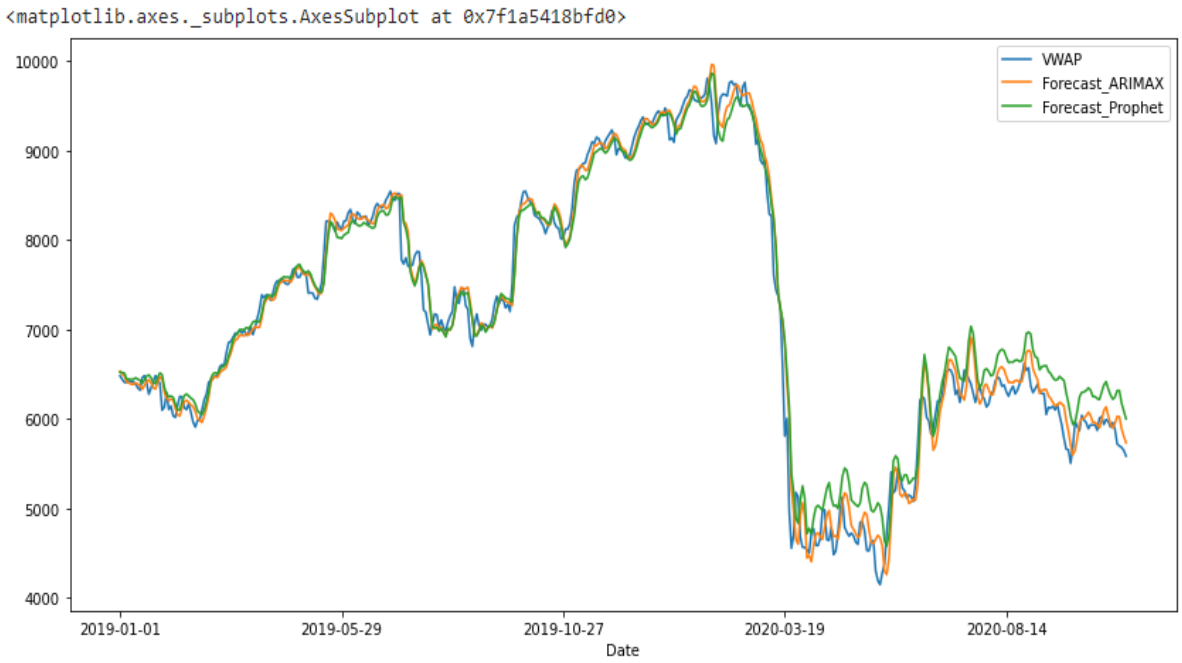

Рисунок 3.17 – прогноз за допомогою APIKC та Prophet на одному графіку

3.6 Прогнозування за допомогою LightGBM

Проблеми з часовими рядами часто перетворюються в табличні структури і передаються в моделі, такі як LightGBM і XGBoost. Існує втрата інформації з точки зору знання порядку проходження точок даних в часовому ряду, але її можна обійти функціями дати, щоб в якійсь мірі захопити цю інформацію. Зверніть увагу, що параметри за замовчуванням використовуються для LightGBM. На рисунку 3.18 зображено прогноз Light GBM в порівнянні з Prophet та APIKC.

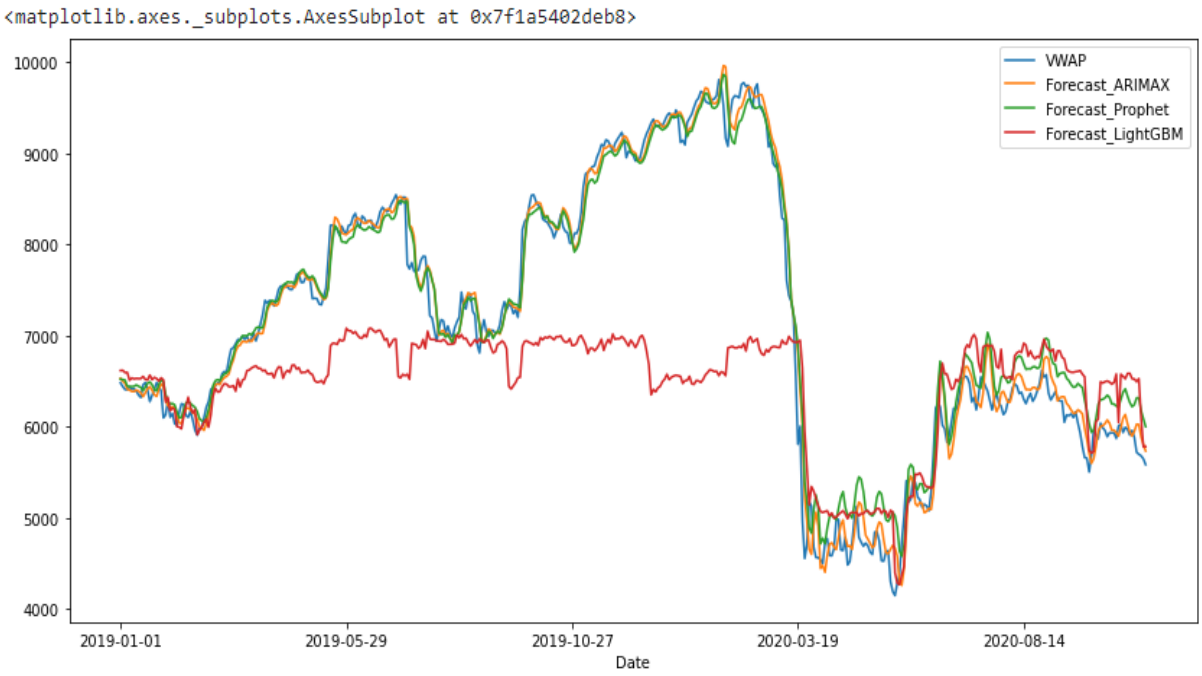

Рисунок 3.18 – прогноз за допомогою APIKC та Prophet в порівнянні з LightGBM на одному графіку

3.7 Прогнозування за допомогою LSTM

LSTM представлена у вигляді нейронної мережі з чотирма нейронними шарами, з дропаутом рівним 0.2 на кожному шарі. Функцією втрат в побудованій мережі є середня абсолютна похибка (MAE). Мережа має 100 епох з розміром батчів 64. Оптимізація відбувається за допомогою алгоритма ADAM.На рисунку 3.18 зображено прогноз LSTM в порівнянні з Prophet, APIKC та Light GBM

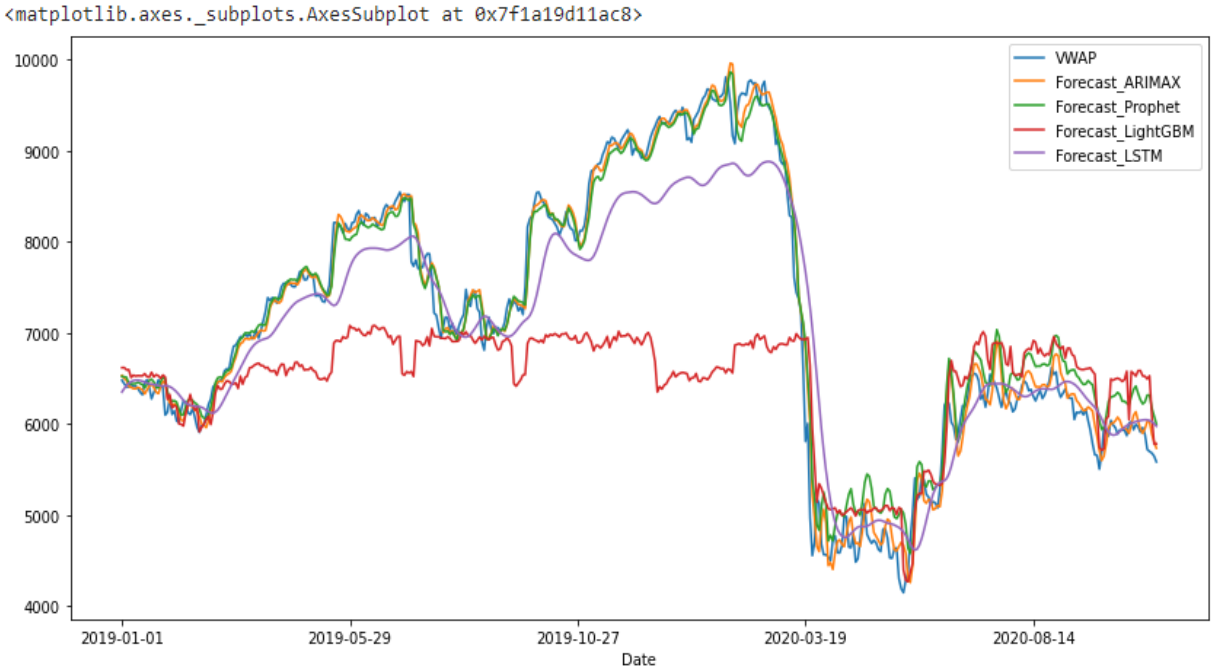

Рисунок 3.18 – порівняння прогнозу на графіку APIKC, Prophet, LightGBM, LSTM

Висновки до розділу 3

Як висновок можна представити таблицю 3.1 згідно до якої класичний підхід у вигляді АРІКС є дещо ліпшим ніж Prophet і значно ліпшим ніж Light GBM, разом з тим LSTM показала себе достатньо добре, проте гірше від Prophet та LSTM.

Таблиця 3.1 Порівняльна характеристика методів

	АРІКС	Prophet	LightGBM	LSTM
RMSE	147.086	151.2678	1233.324	510.939
MSE	104.019	109.829	959.139	379.124
R2	0.979	0.963	0.196	0.877

У даному розділі було продемонстровано побудовані моделі на основі регресійного підходу та інтелектуального аналізу даних за допомогою авторегресійної моделі та рекурентної нейронної мережі разом з виконаним прогнозом на крок вперед. Також було практично реалізовано програмний продукт, що синтезує систему для моделювання та прогнозування нелінійних нестаціонарних часових рядів на реальних статистичних даних. У роботі моделі були апробовані на економічних даних котирування акцій.

Було реалізовано регресійні моделі типу ARIMA, нейронну мережу LSTM, адитивну модель Prophet та дерева рішень Light GBM. Результати моделювання моделювання перевірені відповідними критеріями адекватності моделей та в результаті чого було обрано найкращу модель. Коли найкращу модель вибрали та вибрали найкращу структуру нейронної мережі, виконується прогнозування навчений моделей. Результати прогнозування перевірялися за допомогою критеріїв оцінювання якості прогнозу.

Результати дослідження показують, що:

- APIKC - відмінна базова модель, але більш нові алгоритми, такі як Facebook Prophet, надзвичайно потужні і з кожним днем стають все розумнішими.
- LightGBM, обмежені в прогнозуванні в межах значень цільової змінної в навчальних даних і не екстраполюються при наявності сильного тренда;
- Модель Facebook Prophet робить хорошу роботу по моделюванню, а також по виявленню і відображенню сезонності.
- LSTM є хорошим методом для прогнозування котирування акцій, проте дещо не дотягує до добре підготовленої APIKC.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

На сьогоднішній день все більше стартапів перетворюють ідеї пов’язані з технологіями машинного навчання у працюючі бізнес моделі. Інтерес інвесторів до стартапів, що працюють із машинним навчанням зростає з кожним роком. Все більше і більше з’являється прикладів успішного застосування штучного інтелекту у найрізноманітніших галузях діяльності людини, значно зростає кількість корпорацій що впроваджують системи прогнозування із застосуванням нейромереж у свою операційну діяльність.

Окрім цього, нейронні мережі знайшли широке застосування в прогнозуванні фінансових ринків. Нова і перспективна технологія швидко привернула увагу венчурних інвесторів. У цьому розділі розглянута розробка стартап проекту системи прогнозування фінансових ринків.

4.1. Опис ідеї проекту.

Результативність застосування традиційних методів прогнозування акцій які вільно продаються і купуються на біржах, можна назвати недостатньою для потреб сучасного ринку. Це пов’язано з тим, що інвестиції на фондовому ринку тісно пов’язані з Інтернетом і залежні від інформаційного середовища. Для підвищення точності прогнозування доцільно застосувати таку модель, що не тільки базується на кореляціях факторів та особливостях часового ряду, а й тісно пов’язана з декількома джерелами даних.

Сучасні системи прогнозування не враховують комплексно кількісні та якісні фактори, що впливають на зміну курсів акцій або враховують у векторному вигляді, який обмежує можливості представлення вхідних даних для нейромережі. Такі системи прогнозування зазвичай мають точність 53-58% вірного прогнозу. Цієї точності замало для того, щоб використовувати такі

системи як повноцінний інструмент для економічного аналізу та прогнозування. Отже, ідеєю стартап проекту є створення і розповсюдження системи прогнозування фінансових ринків із більшою ніж у конкурентів точністю прогнозування тренду (табл. 4.1).

Таблиця 4.1 Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1.Фінансова аналітика	Дешевша, порівняно з аналогами, караща точність прогнозування тренду
	2.Торгівля на біржах	Можливість отримувати безпосередній прибуток за рахунок точності прогнозування

Застосування зрозумілої математичної моделі спрощує реалізацію, а доведення кращих показників прогнозування може сприяти розповсюдженню системи. Безумовно, на ринку є аналоги. Для порівняння розробленої системи проведемо аналіз потенційних техніко-економічних переваг ідеї.

Метою аналізу техніко-економічних переваг є чітке виокремлення технічних і маркетингових особливостей розробленого продукту:

1) дослідження характеристик і властивостей розробленої системи;
2) дослідження конкурентів, товарів-аналогів, товарів-замінників та загальної ситуації на ринку де буде комерціалізуватис стартам;
3) проведення порівняльного аналізу слабких, нейтральних та сильних характеристик розробленої истеми.

Визначимо сильні, слабкі та нейтральних характеристик ідеї проекту (табл. 4.2).

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик деї проекту.

№п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Neural Builder 2015	Neural Shell	Neurall Trader			
1.	Вартість ПЗ	Низь	Вис	Вис	Вис			+
2.	Доступність	Низь	Вис	Вис	Сер		+	
3.	Кроссплатформа	Так	Так	Ні	Ні			+
4.	Підтримка	+	-	+	+		+	

З таблиці можна зробити висновок, що розроблена система є конкурентоспроможною.

4.2. Технологічний аудит проекту.

Метою технічного аудиту є визначення переліку технологій, за допомогою який реалізована система і їх аналіз. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

1) за допомогою якої технологією розроблена система згідно ідеї проекту;

2) чи існують у відкритому доступі ці технології, чи їх потрібно додатково розробляти або купувати;
3) чи має доступ розробник до описаних технологій.

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
		Рекурентні нейронні мережі Визначення курсу акцій		
Обраною мовою програмування є Python, використовуються нейронні мережі із довгою короткостроковою пам’яттю				

За результатами аналізу можна зробити висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано мову програмування Python через її доступність та безкоштовність.

4.3. Аналіз ринкових можливостей запуску стартап-проекту.

Метою аналізу ринкових можливостей є виявлення та дослідження обмежень, можливостей та розрахунок конкретних показників реалізації розробленої системи прогнозування фінансових ринків як стартап-проекту.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

.

Проаналізувавши результати, можна зробити висновок що проект є придатним до інвестицій, оскільки його рентабельність перевищує відсоток депозиту.

За результатами порівняння що було наведено у таблиці 4.4 було зроблено висновок, що ринок є придатним для розповсюдження системи як стартап-проекту.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап проекту

№ п/п	Показники стану ринку(найменування)	Характеристика
1	Кількість систем-конкурентів на ринку, од	4
2	Загальний обсяг продаж, грн/ум.од	234000
3	Динаміка ринку	Стагнує
4	Наявність обмежень для входу (вказати характер обмежень)	Нема
5	Специфічні вимоги до стандартизації та сертифікації	Нема
6	Середнє значення рентабельності в галузі(або по ринку), %	18% (відповідає середній річний ставці депозиту у гривні)

Після цього були досліджені потенційні категорії клієнтів, їх особливості та затверджено перелік вимог до системи для кожної категорії клієнтів (табл. 4.5)

Таблиця 4.5 — Характеристика потенційних клієнтів стартап-проект

№ п/п	Потреба, що формує ринок систем прогнозування	Цільова аудиторія та потенційні клієнти	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

Після дослідження потенційних категорій клієнтів було проведено дослідження ринкового середовища: складено таблиці факторів, що сприяють реалізації розробленої системи як стартап-проекту, та факторів, що йому заважають (табл. 4.6, 4.7).

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок систем з кращими Характеристиками та моделлю надання послуг	Вийти на ринок зосередивши увагу на власної переваги системи. Покращення якісних характеристик системи прогнозування. Вибрати цільову аудиторію
2	Зміна потреб користувачів	Клієнту потрібна буде система з більшою точністю прогнозування і новими функціми	Передбачити можливість розширення системи та підвищення точності прогнозування

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогів на українському ринку для вітч. корис-ча	Адаптація системи до особливостей потреб на укр. ринку
2	Поява альтернативних методів моделювання	Нові методи моделювання, більш легкі в освоєні праці	Розширення можливостей, максимальне спрощення

Надалі було проведено дослідження пропозиції: визначили загальні харантеристик конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому виражена дана характеристика	Вплив на діяльність компанії (можливі дії для підвищення конкурентоспроможності)
1.Вказати тип конкуренції - монополія	На ринку присутні декілька постачальників-конкурентів, але їх товар дещо відрізняється від нашої сситеми.	Підтримка якості системи, безперервний розвиток, покращення, вдосконалення, оновлення та підтримка.

Продовження таблиці 4.8

2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу системи Прогнозування таким чином, щоб можна було легко локалізувати її для використання у інших країнах.
3.За галузевою Ознакою - внутрішньогалузева	Система може застосовуватися в одній галузі, але її різних сферах.	Постійне вдосконалення Системи прогнозування, що не має прив'язки до сфери, але має до галузі
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами систем прогнозування, їх особливостями.	Створити систему прогнозування, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - цінова	Покращення процесу створення рограмного продукту, мінімізація витрат на оновлення, застосування безперервної інтеграції	Використання відкритих та дешевших технологій для побудови системи в порівнянні з системами-конкурентами, але тільки якщо ці технології відповідають необхідним критеріям якості.
6. За інтенсивністю - не марочна	Бренд присутній, Але його роль незначна	Реклама, участь у конференціях, семінарах, виставках.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік безпосередніх конкурентів Neural Builder	Визначити бар'єри входження в ринок Наявність вже існуючих рішень	Визначити фактори сили постачальників -	Визначити фактори сили клієнтів Якість системи та її підтримка оновлення	Фактори загроз з боку товарів-замінників Більш відомий розробник, що підтримує свою систему
Висновки:	На даний момент немає конкурентів на українському ринку	Вихід на український ринок буде легшим через відсутність конкуренції	-	Вимоги клієнтів такі, як зручний інтерфейс, якість програмного продукту	Випустити систему, що буду не гірше, ніж у конкурента, але мати кращу точність прогнозування.

За результатами порівняльного дослідження що наведене у табл. 4.9 було зроблено висновок про доцільність виходу на український та міжнародні ринки.

На основі аналізу ситуації на ринках, проведеного в табл. 4.9, а також із урахуванням характеристик розробленої системи (табл. 4.2), вимог потенційних клієнтів до товару (табл. 4.5) та особливостей ринкового середовища (таблиці 4.6, 4.7) досліджується та визначається перелік факторів конкурентоспроможності розробленої системи прогнозування фінансових ринків.

проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.10).

Таблиця 4.10 — Порівняльний аналіз сильних та слабких сторін проекту

п/п	Фактор конкурентоспроможності	али -20	Рейтинг систем-конкурентів у порівнянні з розробленою системою прогнозування						
			3	2	1		1	2	3
	Ціна	5							
	Кросплатформність	0							

Кінцевим кроком маркетингового дослідження можливостей реалізації системи прогнозування фінансових ринків як стартап-проекту є побудова SWOT-аналізу (матриці сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 4.11) на основі описаних конкурентних та маркетингових загроз та можливостей, та сильних і слабких сторін (таблиця 4.10). Список маркетингових загроз та можливостей було

складено на основі дослідження факторів загроз та факторів можливостей ринкової ситуації. Маркетингові загрози та можливості є наслідками (прогнозованими результатами) впливу ринкових факторів.

Таблиця 4.11 — SWOT-аналіз стартап-проекту

Сильні сторони: Ціна, орієнтованість на кінцевого споживача, кросплатформеність	Слабкі сторони: Складність розповсюджувати продукцію закордоном.
Можливості: Відсутність конкуренції на українському ринку	Загрози: Зміна основних потреб клієнтів, при відсутності конкуренції необхідно підтримувати інтерес аудиторії до продукту

За результатами SWOT-аналізу було сформовано альтернативи ринкової стратегії (перелік заходів) для виведення розробленої системи як стартап-проекту на український та міжнародні ринки та розрахований оптимальний час їх ринкової реалізації з урахуванням потенційних розробок конкурентів, що можуть бути виведені на ринок (див. таблицю 4.9, аналіз потенційних конкурентів). Запропоновані альтернативи були проаналізовані з точки зору часу реалізації та ймовірності отримання ресурсів (таблиця 4.12).

Таблиця 4.12 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний Комплекс заходів) ринкової Поведінки	Приблизна ймовірність отримання ресурсів	Приблизні строки реалізації
1	Безкоштовне розповсюдження Обмеженої версії створеного програмного продукту	85%	12 місяців
2	Створення Програмної системи з подальшим платним розповсюдженням (продаж платної ліцензії)	45%	12 місяців

Після аналізу було обрано альтернативу №2.

4.4. Розроблення ринкової стратегії проекту.

Розроблення ринкової стратегії першим етапом передбачає визначення стратегії охоплення ринку: дослідження цільових груп потенційних клієнтів, які приведенні в таблиці 4.13.

Таблиця 4.13. Вибір цільових груп потенційних споживачів

№п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Компанії що займаються фінансовою аналітикою	Готові	Необхідно	Висока	Середня

Визначена цільова група клієнтів: Компанії що займаються фінансовою аналітикою

Дослідивши потенційні групи клієнтів було визначено цільові категорію, для яких буде пропонуватися розроблена система, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу.

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.14).

Таблиця 4.14 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку стартап-проекту	Обрана альтернатива розвитку стартап-проекту	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1		Визначити потреби сучасного ринку для кожної з груп.	Цінова політика, універсальність продукту.	Стратегія Диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.15).

Таблиця 4.15 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект першою подібною системою на ринку?	Чи буде розробник системи шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде розробник копіювати властивості товару конкурента	Стратегія конкурентної поведінки*
1	Ні	Шукати Нових	Ні	Заняття конкурентної ніші

За результатами аналізу вимог клієнтів визначених категорій до розробника стартап-проекту та до продукту (див. таблицю 4.5), а також в залежності від обраної базової стратегії розвитку (таблиця 4.14) та стратегії конкурентної поведінки (таблиця 4.6) розроблено стратегію позиціонування (таблиця 4.16), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торговельну марку/проект.

Таблиця 4.16 — Визначення стратегії позиціонування

№ п/п	Вимоги до системи з боку потенційних клієнтів	Основна стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір основних асоціацій
1	Проста побудова моделі прогнозування, довгий час навчання нейромережі, проте вища точність прогнозу та інтуїтивно зрозумілий інтерфейс, докладне керівництво користувача	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність

За результатами дослідження була сформована система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії українському та міжнародному ринках.

4.5. Розроблення маркетингової програми стартап-проекту

Визначено маркетингову концепцію системи прогнозування фінансових ринків з використанням рекурентних нейромереж, яку буде купувати споживач. Основна концепція продукту — письмовий опис фізичних та прогнозних характеристик системи, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

За результатами дослідження було сформовано трирівневу маркетингову модель системи: ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання:

- 1-й рівень вирішує питання щодо того, засобом вирішення якої потреби і / або проблеми буде система, яка її основна вигода;
- 2-й рівень являє рішення того, як буде реалізована система в реальному ринковому середовищі. Рівень включає в себе якість, властивості, дизайн, упаковку, ціну;
- 3-й рівень визначає додаткові послуги та переваги для клієнта системи, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Після цього визначимо цінові межі, якими необхідно керуватись при встановленні ціни на систему, яке включає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової категорії клієнтів (таблиця 4.17). Аналіз проведено експертним методом.

Таблиця 4.17 — Визначення меж встановлення ціни

№ п/п	Приблизна вилка вартості товарів-замінників	Приблизна вилка вартості товарів-аналогів	Приблизний рівень доходів цільової групи клієнтів	Верхня та нижня межі вартості системи
1	800-3000$	800-3000$	3000$+	200-700$

Після цього визначимо оптимальну систему збуту, за допомогою якої буде розповсюджуватися система як сатрап-проект.(таблиця 4.18)

Таблиця 4.18 — Формування системи збуту

№ п/п	Особливості ринкової поведінки потенційних клієнтів	Функції збуту, які має виконувати постачальник системи	Глибина каналу збуту	Оптимальний канал збуту
1	Цільові клієнти – компанії, які бажають запровадити у своїй роботі сучасні засоби допоможуть прогнозувати фінансові ринки із кращою точністю	Побудова прямих контактів із потенційними клієнтами і їх підтримка. формування попиту і стимулювання продажів.	Один (від виробника одразу споживачу)	Прямий канал збуту до клієнта, мінімізувати збутові витрати. Розвиток нових маркетингових концепцій.

Фінальною складовою маркетингової програми є визначення концепції маркетингових комунікацій.

Результатом дослідження стала робоча ринкова програма, що включає в себе концепції системи, збуту, просування та попереднє дослідження ціноутворення на продукт, базується на цінностях та потребах категорій покупців, конкурентні переваги системи, стан та динаміку маркетингового середовища, в межах якого впроваджено системи прогнозування фінансових ринків яз стартап-проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 4

В цьому розділі було проведено аналіз розробленої системи прогнозування фінансових ринків у якості стартап-проекту. Варто зауважити, що проект має можливість ринкової комерціалізації, через те, що ринок систем прогнозування потребує якісного та інноваційного продукту для прогнозування фінансові ринки.

Результатом роботи є розроблений стартап-проект, план виходу на ринки програмного забезпечення та маркетингова стратегія. Розроблений стартап-проект доцільно застосувати при комерціалізації розробки.

Висока конкуренція зумовлює необхідність виваженого підходу до просування продукту. Проте кількість учасників фінансових ринків зростає з кожним днем, що зумовлює хороші перспективи для комерціалізації системи. Для впровадження ринкової реалізації проекту була обрана стратегія, яка включає розробку програмної системи із класичною моделлю ліцензування за певну плату.

Можна сказати, що подальший розвиток проекту є доцільним, оскільки кількість потенційних користувачів системи зростає кожного року.

ВИСНОВКИ ПО РОБОТІ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Представлена дисертація є результатом дослідження аналізу, моделювання та прогнозування часових рядів, для яких характерні нелінійність та нестаціонарність, а також розробленого програмного продукту за допомогою якого можна отримати емпіричні результати на основі реальних статистичних даних. У даній роботі було продемонстровані моделі, що побудовані на економічних даних – котирування акцій NIFTY-50 в проміжок часу між 2000 роком та початком осені 2020р.

У ході виконаня дисертаційної роботи було створено авторегресійну модель АРІКС, побудовано рекурентну нейронну мережу, використано адитивну модель та розроблено модель на базі деревей прийняття рішень. Виконано аналіз тестових даних на наявність нелінійності, нестаціонарності, сезонності за допомогою відомих тестів. Зроблено графічну візуалізацію побудованих моделей на тестовій та валідаційній вибірках та в порівняльному режимі. Перевірено адекватність побудованих моделей на основі критеріїв адекватності та зроблено висновки щодо якості оцінювання прогнозу. Отримані результати та безпосередньо прогнозування на декілька кроків вперед імплементовано у графіки, що демонструють майбутню тенденцію характеру поведінки розглянутих часових рядів.

Кожен метод має деякі обмеження і недоліки. Обмеження можуть бути подолані шляхом вибору відповідних методів прогнозування для конкретних областей. У майбутньому можна комбінувати ці методи і отримувати правильний результат і висновок. Мною було оцінено різні методи прогнозуваггя котирування акцій, за допомогою яких будь-який інвестор може знайти кращий метод, за допомогою якого можна передбачити котирування акцій набагато точніше, ніж за допомогою раніше використовувалися методів.

Отримані експериментальні результати продемонстрували потенціал моделі АРІКС для прогнозування цінових індексів акцій на короткостроковій

основі. Це може допомогти інвесторам на фондовому ринку приймати вигідні інвестиційні рішення про купівлю / продаж / зберіганні акцій. З отриманими результатами модель АРІКС може досить добре конкурувати з новими методами короткострокового прогнозування.

ей аналіз може бути використаний для зниження похибки прогнозу майбутніх цін акцій у відсотках. Це збільшує шанси інвесторів більш точно прогнозувати ціни, знижуючи відсоток помилки і, отже, збільшуючи свій прибуток на фондових ринках. Використання нейромережевих моделей разом з іншими інструментами і методиками прогнозування може вважатися ще одним цінним досягненням в століття технологій.

Для покращення майбутніх дослідженнь економічних процесів необхідно спробувати застосувати інші моделі регресійного аналізу та їх модифіковані типи, а також спробувати ансамбль цих методів. Для отримання більш цілісної картини можна застосувати пояснювальну змінну, що буде характеризувати вплив різних індексів на ціну акцій. Також доцільно буде розглянути інші методи моделювання та прогнозування такі як GAN, eGAN та інші методи машинного навчання. Для покращення вибору моделі необхідно збільшити кількість рівнів перевірки адекватності моделювання та якості прогнозу, врахувати максимальну та мінімальну абсолютну похибку та метод максимальної правдоподібності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Wilson O. Sharda B. Neural networks for stock predictions. *PLoS ONE* №28. 2013. P. 13.
2. Wong, F. S., Wang, P. Z., Goh, T. H., & Quek, B. K. Fuzzy neural systems forstock selection. *Financial Analysts Journal*. №4. 1992 P.47-52.
3. Kim, K. & Han, I. Extracting trading rules from the multiple classifiers and technical indicators in stock market. *In Proceedings of KMIS'98 International Conference*. 1998. P. 20-31
4. Global Burden of Disease Collaborative Network. Global Burden of Disease Study 2017 (GBD 2017) Results. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2018. URL: http://ghdx.healthdata.org/gbd-results-tool (Last accessed: 14.11.2020)
5. Касперович С. А. Прогнозирование и планирование экономики : курс лекций для студентов специальностей 1-25 01 07 «Экономика и управление предприятием», 1-25 01 08 «Бухгалтерский учет, анализ и аудит», 1-26 02 02 «Менеджмент», 1-26 02 03 «Маркетинг». Минск. БГТУ. Минск: 2007. 172 с.
6. Антохонова И.В. Методы прогнозирования социально-экономических процессов: Учебное пособие. Улан-Удэ: Изд-во ВСГТУ. Улан-Удэ: 2004. 212 с.
7. Львович, М. И. Вода и жизнь. Водные ресурсы, их преобразование и охрана. Москва : Мысль, 1986. 254 с.
8. Athanasios Loukas, Exploring the Non-Stationary Effects of Forests and Developed Land within Watersheds on Biological Indicators of Streams Using Geographically-Weighted Regression; Seoul: Department of Environmental Planning, Konkuk University, 2016. 310 p.
9. AboutEViews: Part 2: Powerful Analytical Tools, Presentation Quality Output. URL: https://www.eviews.com/EViews10/ev10analytics.html (Last accessed: 15.11.2020)

10. Andrej Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, Blog by Andrej Karpathy – 2015 URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (Last accessed: 15.11.2020)
11. Бідюк П.І., Коршевнюк Л.О., Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібни. Київ: ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 340 с.
12. Бідюк П.І., Половцев О.В. Аналіз та моделювання економічних процесів перехідного періоду. Київ: ПЛАБ-75, 1999. 230 с.
13. Бидюк П.И., Баклан И.В. Системный подход к построению регрессионной модели по временным рядам. *Системні дослідження та інформаційні технології*. 2002. № 3. С. 114-131.
14. Бідюк П.І. Часові ряди: моделювання та прогнозування. Київ: ЕКМО, 2004.144 с.
15. Пашин В.П. Функционально-стоимостный анализ конструкторско-технологических решений. Київ: РДЭНТП «Знание» УССР, 1989. 222с.
16. Пашін В.П. Оцінка конкурентоспроможності електронних пристроїв на стадії проектування. Київ: Економічний вісник НТУУ „КПІ", 2006. 255с.
17. Пашин В.П. Управление качеством изделий на основе функционально-стоимостного анализа. Київ: «Технология и организация производства», 1989. 177с.
18. Пашін В. П. Методичні вказівки до виконання економіко-організаційного розділу дипломних проектів (робіт) освітньо-кваліфікаційних рівнів «бакалавр» і «спеціаліст» для студентів інституту прикладного системного аналізу. URL: https://ela.kpi.ua/handle/123456789/1819 (Останній доступ: 18.11.2020р.)

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
!pip install pyramid

!pip install pmdarima

import lightgbm as lgb

import numpy as np

import pandas as pd

from fbprophet import Prophet

from matplotlib import pyplot as plt

from pmdarima import auto_arima

from sklearn.metrics import mean_absolute_error, mean_squared_error

myfavouritenumber = 13

seed = myfavouritenumber

np.random.seed(seed)

df = pd.read_csv("/content/BAJAJFINSV.csv")

df.set_index("Date", drop=False, inplace=True)

df.head()

df

df.VWAP.plot(figsize=(14, 7))

df.reset_index(drop=True, inplace=True)

lag_features = ["High", "Low", "Volume", "Turnover", "Trades"]

window1 = 3

window2 = 7

window3 = 30

df_rolled_3d = df[lag_features].rolling(window=window1, min_periods=0)

df_rolled_7d = df[lag_features].rolling(window=window2, min_periods=0)

df_rolled_30d = df[lag_features].rolling(window=window3, min_periods=0)

df_mean_3d = df_rolled_3d.mean().shift(1).reset_index().astype(np.float32)

df_mean_7d = df_rolled_7d.mean().shift(1).reset_index().astype(np.float32)

df_mean_30d = df_rolled_30d.mean().shift(1).reset_index().astype(np.float32)

df_std_3d = df_rolled_3d.std().shift(1).reset_index().astype(np.float32)

df_std_7d = df_rolled_7d.std().shift(1).reset_index().astype(np.float32)
```

```
    df_std_30d = df_rolled_30d.std().shift(1).reset_index().astype(np.float32)

    for feature in lag_features:

        df[f"{feature}_mean_lag{window1}"] = df_mean_3d[feature]

        df[f"{feature}_mean_lag{window2}"] = df_mean_7d[feature]

        df[f"{feature}_mean_lag{window3}"] = df_mean_30d[feature]

        df[f"{feature}_std_lag{window1}"] = df_std_3d[feature]

        df[f"{feature}_std_lag{window2}"] = df_std_7d[feature]

        df[f"{feature}_std_lag{window3}"] = df_std_30d[feature]

    df.fillna(df.mean(), inplace=True)

    df.set_index("Date", drop=False, inplace=True)

    df.head()

    print()

    df.Date = pd.to_datetime(df.Date, format="%Y-%m-%d")

    df["month"] = df.Date.dt.month

    df["week"] = df.Date.dt.week

    df["day"] = df.Date.dt.day

    df["day_of_week"] = df.Date.dt.dayofweek

    df.head()

    df_train = df[df.Date < "2019"]

    df_valid = df[df.Date >= "2019"]

    exogenous_features = ["High_mean_lag3", "High_std_lag3", "Low_mean_lag3", "Low_std_lag3",
                          "Volume_mean_lag3", "Volume_std_lag3", "Turnover_mean_lag3",
                          "Turnover_std_lag3", "Trades_mean_lag3", "Trades_std_lag3",
                          "High_mean_lag7", "High_std_lag7", "Low_mean_lag7", "Low_std_lag7",
                          "Volume_mean_lag7", "Volume_std_lag7", "Turnover_mean_lag7",
                          "Turnover_std_lag7", "Trades_mean_lag7", "Trades_std_lag7",
                          "High_mean_lag30", "High_std_lag30", "Low_mean_lag30", "Low_std_lag30",
                          "Volume_mean_lag30", "Volume_std_lag30", "Turnover_mean_lag30",
                          "Turnover_std_lag30", "Trades_mean_lag30", "Trades_std_lag30",
                          "month", "week", "day", "day_of_week"]

    model = auto_arima(df_train.VWAP, exogenous=df_train[exogenous_features], trace=True, error_action="ignore",
suppress_warnings=True)

    model.fit(df_train.VWAP, exogenous=df_train[exogenous_features])
```

```
forecast = model.predict(n_periods=len(df_valid), exogenous=df_valid[exogenous_features])
df_valid["Forecast_ARIMAX"] = forecast
df_valid[["VWAP", "Forecast_ARIMAX"]].plot(figsize=(14, 7))
print("RMSE of Auto ARIMAX:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_ARIMAX)))
print("\nMAE of Auto ARIMAX:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_ARIMAX))
 model_fbp = Prophet()
 for feature in exogenous_features:
    model_fbp.add_regressor(feature)
 model_fbp.fit(df_train[["Date", "VWAP"] + exogenous_features].rename(columns={"Date": "ds", "VWAP": "y"}))
 forecast = model_fbp.predict(df_valid[["Date", "VWAP"] + exogenous_features].rename(columns={"Date": "ds"}))
 df_valid["Forecast_Prophet"] = forecast.yhat.values
model_fbp.plot_components(forecast)
df_valid[["VWAP", "Forecast_ARIMAX", "Forecast_Prophet"]].plot(figsize=(14, 7))
print("RMSE of Auto ARIMAX:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_ARIMAX)))
print("RMSE of Prophet:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_Prophet)))
print("\nMAE of Auto ARIMAX:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_ARIMAX))
print("MAE of Prophet:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_Prophet))
params = {"objective": "regression"}
dtrain = lgb.Dataset(df_train[exogenous_features], label=df_train.VWAP.values)
dvalid = lgb.Dataset(df_valid[exogenous_features])
model_lgb = lgb.train(params, train_set=dtrain)
forecast = model_lgb.predict(df_valid[exogenous_features])
df_valid["Forecast_LightGBM"] = forecast
df_valid[["VWAP", "Forecast_ARIMAX", "Forecast_Prophet", "Forecast_LightGBM"]].plot(figsize=(14, 7))
from sklearn.metrics import r2_score
print("RMSE of Auto ARIMAX:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_ARIMAX)))
print("RMSE of Prophet:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_Prophet)))
print("RMSE of LightGBM:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_LightGBM)))
print("\nMAE of Auto ARIMAX:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_ARIMAX))
print("MAE of Prophet:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_Prophet))
print("MAE of LightGBM:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_LightGBM))
print("\nR2 of ARIMAX:", r2_score(df_valid.VWAP, df_valid.Forecast_ARIMAX))
print("R2 of Prophet:", r2_score(df_valid.VWAP, df_valid.Forecast_Prophet))
```

```
print("R2 of LightGBM:", r2_score(df_valid.VWAP, df_valid.Forecast_LightGBM))

### Importing Required Libraries ###

import pandas as pd

import numpy as np

import datetime as dt

from datetime import datetime

import matplotlib.pyplot as plt

import numpy as np

from sklearn.preprocessing import MinMaxScaler

### Create the Stacked LSTM model

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import LSTM

from keras.layers import Dropout

features = ["Date", "VWAP"]

all_data = df[features]

df_valid

#creating training and validation sets

dataset = all_data.values

train = dataset[:2624,:]

valid = dataset[2624:,:]

scaler = MinMaxScaler(feature_range=(0, 1))

scaled_data = scaler.fit_transform(dataset)

LSTM_window = 24

x_train, y_train = [], []

for i in range(LSTM_window,len(train)):

    x_train.append(scaled_data[i-LSTM_window:i,0])

    y_train.append(scaled_data[i,0])

x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))

# create and fit the LSTM network

model = Sequential()

model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1],1)))

model.add(Dropout(rate = 0.2))
```

```
model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(rate = 0.2))
model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(rate = 0.2))
model.add(LSTM(units=50, return_sequences = False))
model.add(Dropout(rate = 0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=50, batch_size=64, verbose=1)
#predicting test data values, using past LSTM_window from the train data
inputs = all_data[len(all_data) - len(valid)-LSTM_window:].values
inputs = inputs.reshape(-1,1)
inputs  = scaler.transform(inputs)
X_test = []
for i in range(LSTM_window,inputs.shape[0]):
    X_test.append(inputs[i-LSTM_window:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
preds = model.predict(X_test)
preds = scaler.inverse_transform(preds)
rms=np.sqrt(np.mean(np.power((valid-preds),2)))
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
r2_score(preds,valid)
mean_absolute_error(preds,valid)/579
#for plotting
train = all_data[:2624]
valid = all_data[2624:]
valid['Predictions'] = preds
plt.figure(figsize=(20,8))
plt.plot(train['VWAP'])
plt.plot(valid['VWAP'], color = 'blue', label = 'Real Price')
plt.plot(valid['Predictions'], color = 'red', label = 'Predicted Price')
```

```
plt.title('SBIN price prediction')

plt.legend()

plt.show()

r2_score(preds,valid['VWAP'])

forecast

df_valid["Forecast_LSTM"] = preds

df_valid

df_valid[["VWAP", "Forecast_ARIMAX", "Forecast_Prophet", "Forecast_LightGBM","Forecast_LSTM"]].plot(figsize=(14, 7))

print("RMSE of Auto ARIMAX:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_ARIMAX)))

print("RMSE of Prophet:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_Prophet)))

print("RMSE of LightGBM:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_LightGBM)))

print("RMSE of LSTM:", np.sqrt(mean_squared_error(df_valid.VWAP, df_valid.Forecast_LSTM)))

print("\nMAE of Auto ARIMAX:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_ARIMAX))

print("MAE of Prophet:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_Prophet))

print("MAE of LightGBM:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_LightGBM))

print("MAE of FLSTM:", mean_absolute_error(df_valid.VWAP, df_valid.Forecast_LSTM))


print("\nR2 of ARIMAX:", r2_score(df_valid.VWAP, df_valid.Forecast_ARIMAX))

print("R2 of Prophet:", r2_score(df_valid.VWAP, df_valid.Forecast_Prophet))

print("R2 of LightGBM:", r2_score(df_valid.VWAP, df_valid.Forecast_LightGBM))

print("R2 of LSTM:", r2_score(df_valid.VWAP, df_valid.Forecast_LSTM))

forecast.shape
```