

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики факультету інформатики



Кількісні інвестиційні стратегії
Курсова робота
за спеціальністю „Прикладна математика ” 6.040301

Керівник курсової роботи

Канд. ф.-м. наук, доц. Чорней Р. К.

(підпис)

“ ____ ” _____ 2019 р.

Виконав

студент 3 курсу

факультету інформатики

Іванюк-Скульський Б. В.

Зміст

Календарний план виконання роботи.....	1
Вступ	2
Теоретичні відомості	4
1. Технічний аналіз	5
1.1 RSI (Індекс Відносної Сили, Relative Strength Index)	5
1.2 Bollinger Bands (Лінії Боллінжера)	5
1.3 Aroon Oscillator (Індикатор Арун)	7
1.4 Stochastic Oscillator (Стохастичний індикатор)	8
1.5 ROC (Швидкість змін, Rate of Changes)	9
1.6 VWAP (Обсяг середньозваженої ціни, Volume Weighted Average Price)	10
1.7 Momentum	11
1.8 CCI (Індекс товарного каналу, Commodity Channel Index)	11
2. Прогнозування	13
2.1 Модель авторегресії (AR)	13
2.2 Ковзаюче середнє (Moving Average, MA)	15
2.3 Авторегресивне ковзаюче середнє (ARMA)	15
2.4 ARIMA	16
2.5 Рекурентні нейронні мережі (RNN) та LSTM (Long-Short Term Memory)	18
Розділ 3	22
3.1 Лінійна регресія	22
3.2 Метод прямого поширення помилки (forward propagation)	24
3.3 Метод зворотного поширення помилки (backpropagation)	25
Висновки	26
Список використаної літератури	27

Календарний план виконання роботи

Номер	Назва етапу курсової роботи	Термін виконання етапу
1	Отримання теми курсової роботи	10.10.18
2	Огляд літератури за темою роботи	20.10.18
3	Нарешті зрозумів тему курсової роботи	20.04.19
4	Написання програмного продукту	21.04.19
5	Написання вступу та плану роботи	30.05.19
6	Написання основних розділів роботи	02.05.19
7	Розумію, що все написане до цього не по темі роботи	04.05.19
8	Пересування роботи	05.05.19
9	Написання висновку	07.05.19

Вступ

Мета дослідження: розглянути основні методи прогнозування ціни акцій та реалізувати деякі з них.

Предмет дослідження: дослідження алгоритмів прогнозування

Об'єкт та методи дослідження: кількісні інвестиційні стратегії методами тестування та порівняння.

Що було зроблено: реалізовано основні методи технічного аналізу та прогнозування (модель авторегресії, ARIMA та рекурентна нейронна мережа)

Прогнозування вартості цінних паперів одна з найскладніших задач інвестиційних фінансів, оскільки є дуже багато факторів, які впливають на результат передбачень: фізичні фактори, психологічні, раціональна та нерациональна поведінки. Всі ці аспекти впливають на коливання ціни акцій, що спричиняє складність передбачень.

“Мавпа з зав'язаними очима, кидаючи дротики на фінансові сторінки газети, може вибрати портфель, який буде так само добре заробляти гроші, як і ретельно відібраний експертами”

Бертон Малкіл,

Американський економіст та письменник

Враховуючи *гіпотезу ефективного ринку* (вся суттєва інформація негайно і повною мірою впливає на ринкову ціну акцій), ринок цінних паперів є непередбачуваним та незалежним від попередніх змін ціни. З цієї гіпотези слідує, що ціни не можуть бути прогнозованими, однак багато учасників ринку не бачать ситуацію таким чином.

В цій роботі я досліджую найбільш популярні методи передбачення вартості цінних паперів.

Широко використовуваний математичний метод прогнозування полягає у виявленні закономірностей в історичних рухах ціни акцій та перевірці

існування цієї закономірності у майбутньому. Інший метод полягає у виявленні провідного фактору, який буде використовуватися для передбачення ціни до її зміни. Вищезгаданий фактор суперечить гіпотезі ефективного ринку, але це не зупиняє людей застосовувати цей метод.

Теоретичні відомості

Прогнозування ціни акцій дуже популярна задача кількісних фінансів. Багато алгоритмів машинного навчання та статистики були запропоновані для передбачення ціни акцій: нейронні мережі, дерева рішень, генетичні алгоритми та інші.

Огляд попередніх досліджень щодо прогнозування ціни акцій демонструє переважне використання технічних індикаторів разом з штучними нейронними мережами для прогнозування ринку цінних паперів протягом останніх двох десятиліть. Дослідження Кун Хуан та Тіффані Хуей-Куанг показало, що штучні нейронні мережі мають кращу спроможність прогнозування за моделі часових рядів.

Дерева рішень найкраще підходять для прийняття фінансових рішень в умовах, коли потрібно брати до уваги багато даних. Вони надають ефективну структуру, в якій можуть бути встановлені та оцінені альтернативні рішення та наслідки прийняття цих рішень. Також дерева рішень формують схему ризиків та нагород, які можуть виникнути в результаті конкретного вибору.

В цьому дослідженні розглядається порівняння методів штучних нейронних мереж та моделей часових рядів.

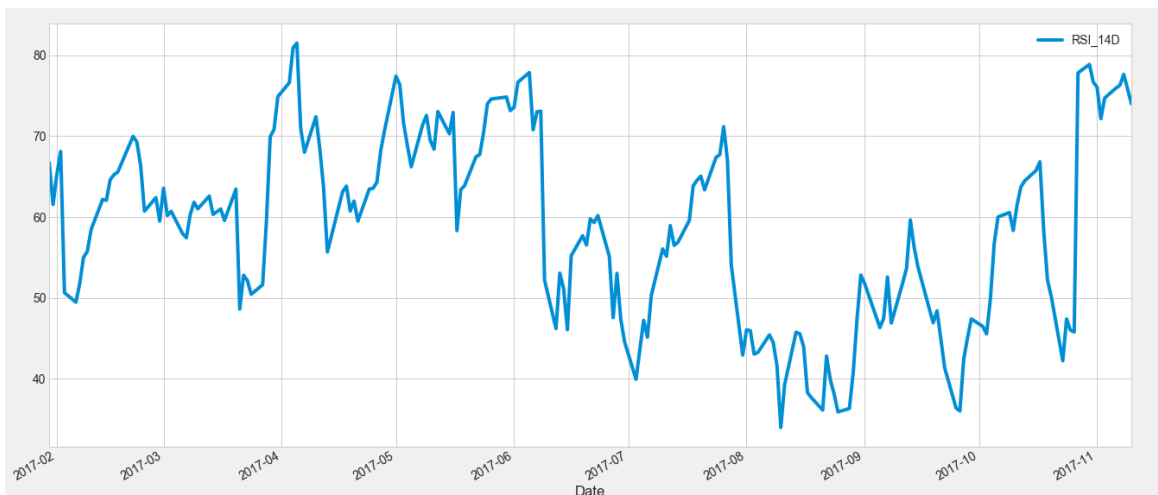
1. Технічний аналіз

Технічний аналіз включає в себе аналіз коливань ціни та обсягу торгів з метою прогнозування ціни та встановленню вдалих часових проміжків для продажу або купівлі активу.

Технічний аналіз часто піддається критиці, оскільки більшість індикаторів не має чіткого обґрунтування в статистиці.

1.1 RSI (Індекс Відносної Сили, Relative Strength Index)

Імпульсний показник, що вимірює величину нещодавніх змін цін акцій чи інших активів для оцінки умов переоціненості або недооцінений.



Традиційна інтерпретація та використання RSI полягає в тому, що значення 70 одиниць або вище вказує на те, що цінні папери стають переоціненими з чого випливає ймовірне падіння ціни. З іншого боку, значення 30 одиниць або менше вказує на недооцінений стан і вважається, що ціна, ймовірно, зросте.

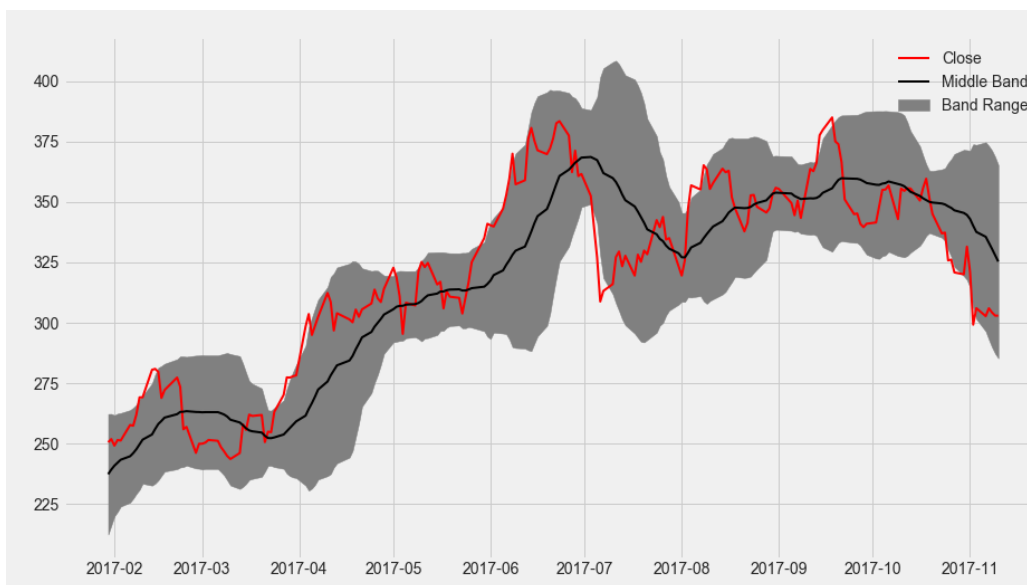
$$RSI = 100 - \frac{100}{1 + \frac{\text{середній приріст}}{\text{середня втрата}}}$$

1.2 Bollinger Bands (Лінії Боллінжера)

Інструмент фінансового аналізу, що визначається набором ліній, побудованих на двох стандартних відхиленнях від звичайного

ковзаючого середнього. Оскільки стандартне відхилення є мірою коливань ціни, при збільшенні мінливості ринку, смуги розширюються. Під час менш мінливих періодів смуги стискаються.

$$Upper\ Band = Moving\ Average + 2 * Std$$

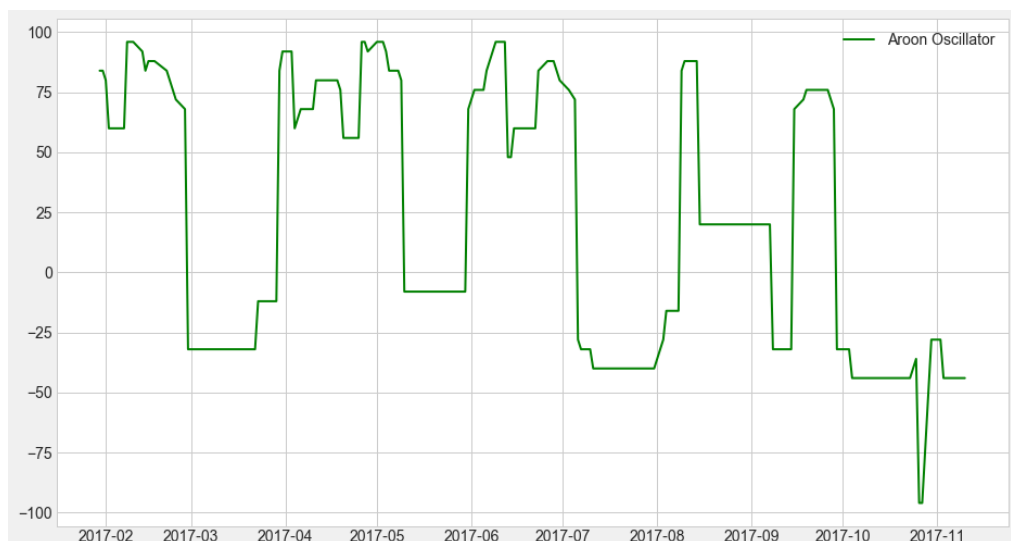


$$Lower\ Band = Moving\ Average - 2 * Std$$

- *Squeeze* - головна концепція ліній Боллінжера. Зближення граничних ліній свідчить про період низької мінливості ціни, і торговці вважають його ймовірною ознакою підвищення коливань та потенційних торговельних можливостей. І навпаки, чим ширше розходяться смуги, тим більша ймовірність зменшення мінливості ціни та більшої можливості виходу з торгівлі. Смуги не зазначають точного часу, коли зміни матимуть місце та в яку сторону ціна може піти.
- *Breakouts*. Близько 90% коливань ціни лежать в межах ліній Боллінжера. Будь-який викид за ці лінії – серйозна подія, яка не є сигналом для продажу або купівлі, але вимагаюча досліджень. Викиди не підказують напрямки та масштаби майбутніх змін ціни.

1.3 Aroon Oscillator (Індикатор Арун)

Тренд, який використовується для оцінки силу поточної тенденції та ймовірності її продовження. Значення вище нуля вказують на наявність висхідного тренду, показник нижче нуля вказує на спадний тренд.



$$Aroon\ Oscillator = Aroon\ Up - Aroon\ Down$$

$$Aroon\ Up = 100 * \frac{25 - \text{кіл. періодів з } max \text{ знач. за 25 пер.}}{25}$$

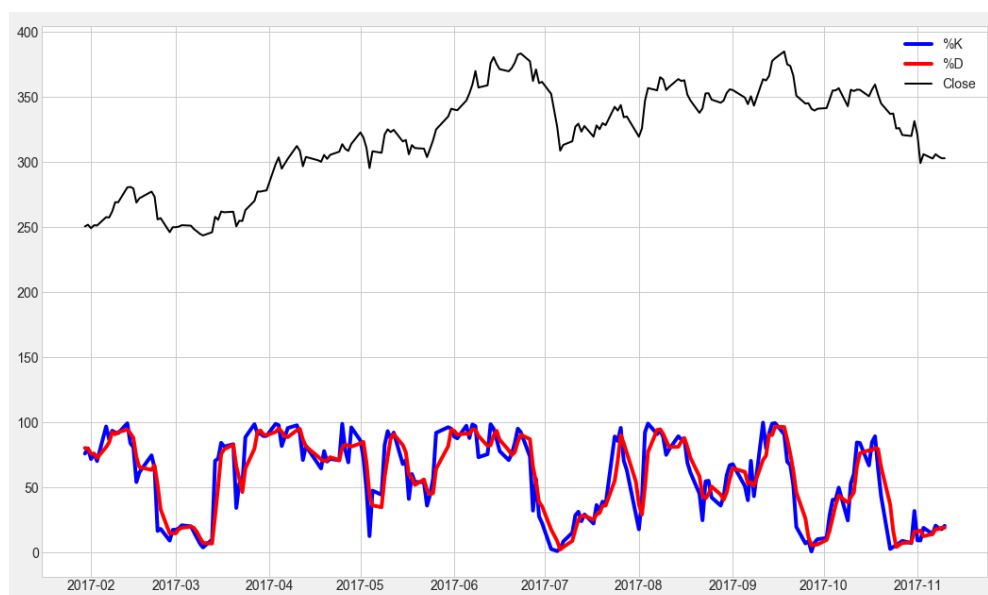
$$Aroon\ Down = 100 * \frac{25 - \text{кіл. періодів з } min \text{ знач. за 25 пер.}}{25}$$

Індикатор зазвичай використовує суб'єктивні часові проміжки у 25-ть періодів. При використанні більшої кількості періодів, отримаємо більш гладкий індикатор. Значення індикатора коливаються у діапазоні між 0 та 100. Чим більше значення - тим сильніше тренд. Наприклад, ціна що досягла нового максимуму 1 день тому, матиме значення Aroon Up 96 $(=(25-1)/25)*100$). Аналогічно і для Aroon Down.

Максимуми та мінімуми, що використовуються в розрахунках Aroon Up та Aroon Down, допомагають створити зворотний зв'язок між двома показниками. Коли значення Aroon Up збільшується, значення Aroon Down, як правило, зменшується і навпаки.

1.4 Stochastic Oscillator (Стохастичний індикатор)

Індикатор, що порівнює конкретну ціну активу на час закриття біржі з діапазоном його цін з певного періоду. Основою для цього показника є те, що на ринку, який рухається вгору, ціни наближатимуться до вищого рівня, а в умовах тенденції на зниження – ціни наближатимуться до нижчого рівня.



$$\%K = \frac{\text{остання ціна закриття} - \text{найнижча ціна 14 днів}}{\text{найвища ціна 14 днів} - \text{найнижча ціна 14 днів}} * 100$$

$$\%D = \text{ковзаюче середнє } \%K$$

Значення цього індексу завжди знаходяться в діапазоні від 0 до 100.

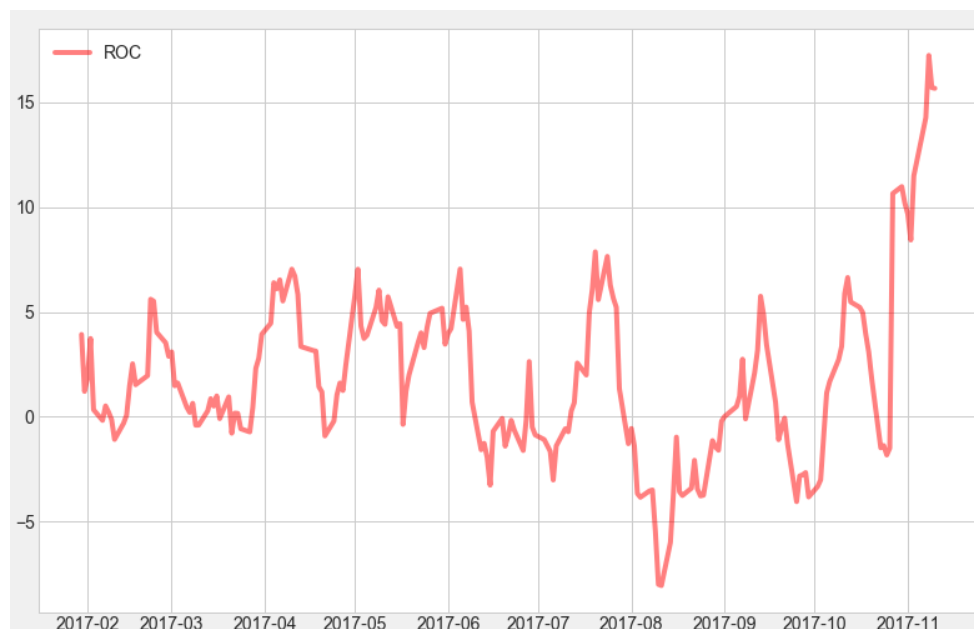
Це робить його корисним для визначення рівня переоціненості та недооціненості. Традиційно значення понад 80 одиниць символізує переоціненість, а значення до 20 одиниць - недооціненість. Проте, вони не завжди вказують на майбутню зміну тенденції. RSI та Stochastic Oscillator дуже часто використовують в тандемі, хоча вони мають різні теоретичні основи. Stochastic Oscillator спирається на припущення, що ціна закриття прямує відповідно поточної тенденції. В той час як RSI розроблений для визначення переоціненості та недооціненості, вимірюючи швидкості руху цін.

1.5 ROC (Швидкість змін, Rate of Changes)

Фінансовий показник, що визначає відсоткову зміну ціни поточного періоду в порівнянні з 'n' попередніх періодів. Вимірюється відносно нульової лінії, тобто розрізняє позитивні та негативні значення.

Позитивні значення вказують на висхідний тренд та схильність до купівлі, в той час як негативні - на схильність до продажу, та низхідний тренд. Зазвичай, цей індекс не використовується для торговельних цілей, а радше, це попередження про те, що може наблизитися зміна тенденції ціни.

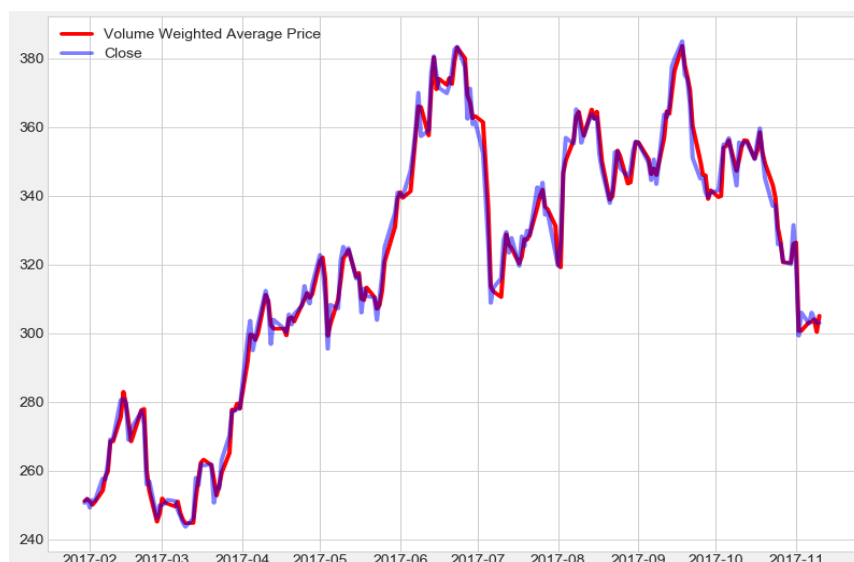
$$ROC = \frac{\text{ціна закриття} - \text{ціна закриття } n \text{ минулих періодів}}{\text{ціна закриття } n \text{ минулих періодів}} * 100$$



1.6 VWAP (Обсяг середньозваженої ціни, Volume Weighted Average Price)

Індекс, що розраховує середню ціну, за якою торгуються цінні папери протягом дня, на основі обсягу та ціни. Великі інвестиційні компанії використовують цей індекс, для зайняття позицій чи їх скидання. Отже, варто купляти акції, коли ціна менша за VWAP, та продавати, коли більша.

Індекс розраховується шляхом сумування доларів, що торгуються за кожну операцію (ціна помножена на кількість акцій, що торгуються), а потім діленням на загальну частку, що торгується.



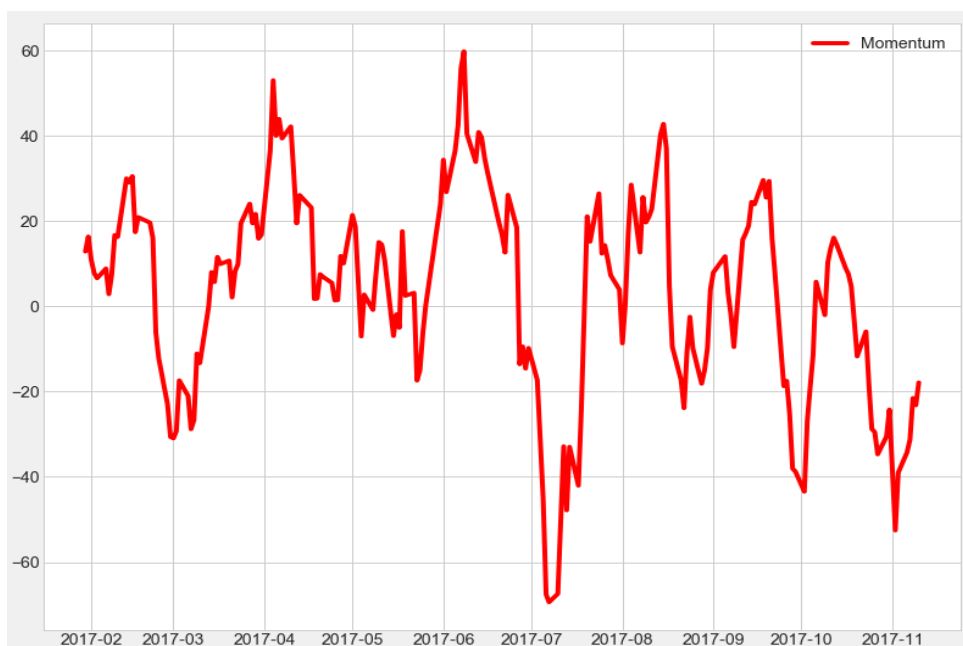
$$VWAP = \frac{\text{ціна} * \text{кількість акцій, що торгується в цій угоді}}{\text{загальна кількість акцій, що торгується}}$$

1.7 Momentum

Індекс, який вимірює темп зростання або спадання ціни акції. Для аналізу тенденцій Momentum є корисним при визначенні сили або слабкості активу.

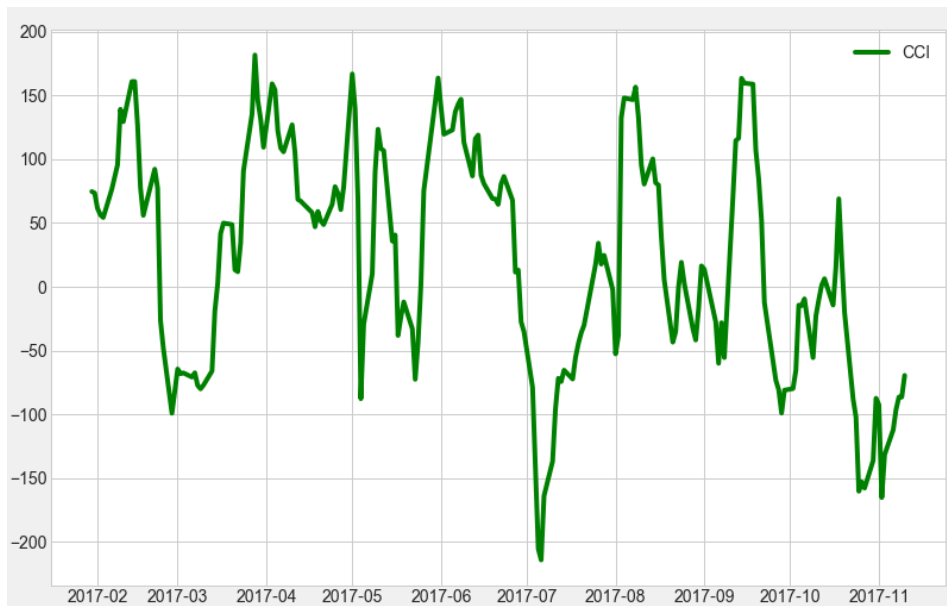
$Momentum = \text{Остання ціна} - \text{Ціна закриття } n \text{ минулих періодів}$

Розглянемо використання Momentum на прикладі потягу. Потяг починає розганятися, але рухається ще повільно. Набирає швидкість. Потяг перестає прискорюватися, але по інерції їде з великою швидкістю. Зрештою, починаючи гальмувати, він уповільнюється. Отже, найкращий час для інвесторів це середина, коли ціна рухається з великою швидкістю.



1.8 CCI (Індекс товарного каналу, Commodity Channel Index)

Індекс, що використовують для виявлення нових тенденцій, спостерігаючи за рівнем переоціненості та недооціненості, а також для виявлення слабкості тенденцій, коли показник розходиться з цінами. Перехід CCI від негативного, або майже нульового значення до 100 одиниць, може свідчити про новий висхідний тренд.



Ця ж концепція застосовується до нових трендів у спаді. Коли індикатор переходить від позитивного, чи близького нулю значенню до -100 одиниць, то може початися низхідний тренд, що є сигналом для того, щоб вийти з довгих позицій, або почати шукати можливості у короткострокових позиціях. Рівні переоціненості/недооціненості не фіксовані, оскільки індикатор не обмежений. Отже, варто дивитися на минулі значення індексу для конкретного активу, щоб мати уявлення про те, де ціна починає змінюватися. Для одного активу індекс може мати тенденцію до зворотного руху біля +200 та -150 одиниць, для іншого біля +325 та -350 одиниць.

$$CCI = \frac{\text{Типова ціна} - \text{Ковзаюче середнє}}{0.15 * \text{Середнє відхилення}}$$

$$\text{Типова ціна} = \frac{\text{Ден. максимум} + \text{Ден. мінімум} + \text{Ціна закриття}}{3}$$

$$\text{Ковзаюче середнє} = \frac{\text{Типові ціна}}{\text{кількість періодів}}$$

$$\text{Середнє відхилення} = \frac{|\text{Типова ціна} - \text{Ковзаюче середнє}|}{\text{кількість періодів}}$$

2.Прогнозування

2.1 Модель авторегресії (AR)

Модель часових рядів, що використовує спостереження з попередніх часових кроків(як вхідні дані рівняння регресії) для передбачення значень на наступному кроці.

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_n X_{t-p} + \varepsilon_t$$

Щоб отримати більш загальний вигляд, зручно записати модель з ненульовим середнім

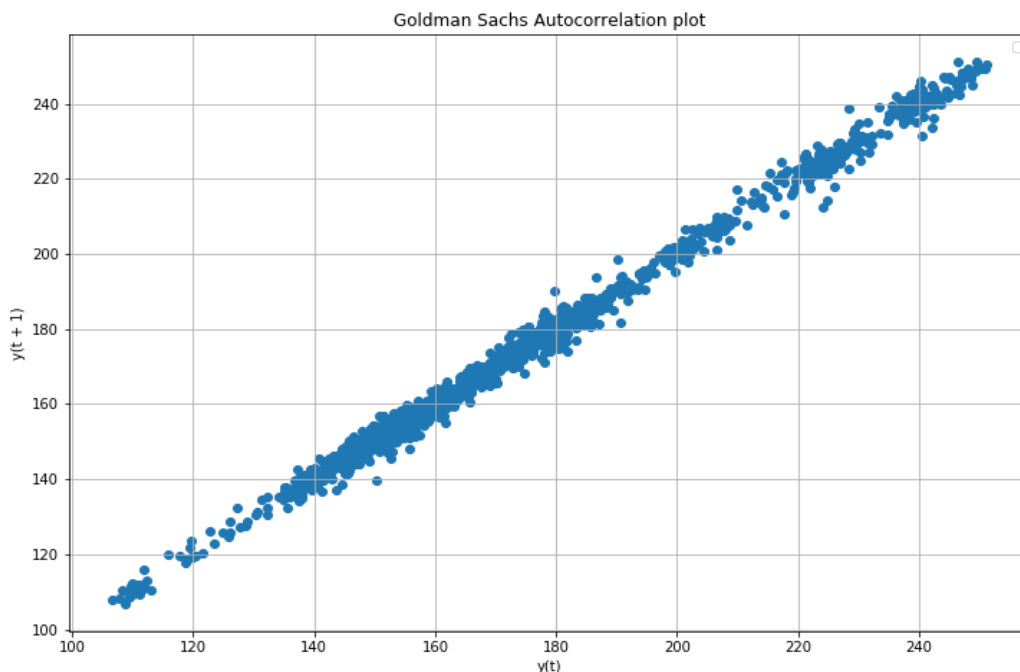
$$(X_t - \mu) = \alpha_1 (X_{t-1} - \mu) + \dots + \alpha_n (X_{t-p} - \mu) + \varepsilon_t$$

Коли $\mu = 0$, $E[X_t] = 0$ для всіх t . Можна уникнути цього обмеження, взявши $\mu \neq 0$. Новий результат $Z_t = X_t - \mu$ має 0 середнє та задовольняє початкове рівняння:

$$Z_t = \alpha_1 Z_{t-1} + \dots + \alpha_n Z_{t-p} + \varepsilon_t$$

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_n X_{t-p} + \varepsilon_t + (\mu - \alpha_1 \mu - \dots - \alpha_p \mu) = \alpha_1 X_{t-1} + \dots + \alpha_n X_{t-p} + \varepsilon_t + \tilde{\mu}$$

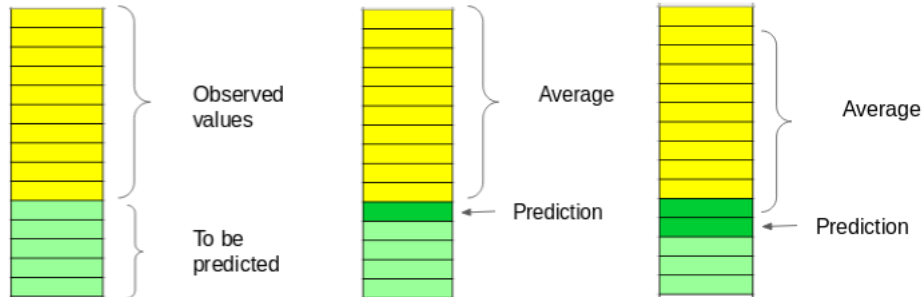
Першим чином перевіримо данні на автокореляцію, що визначить якою мірою попередні значення ціни відносяться до поточних.



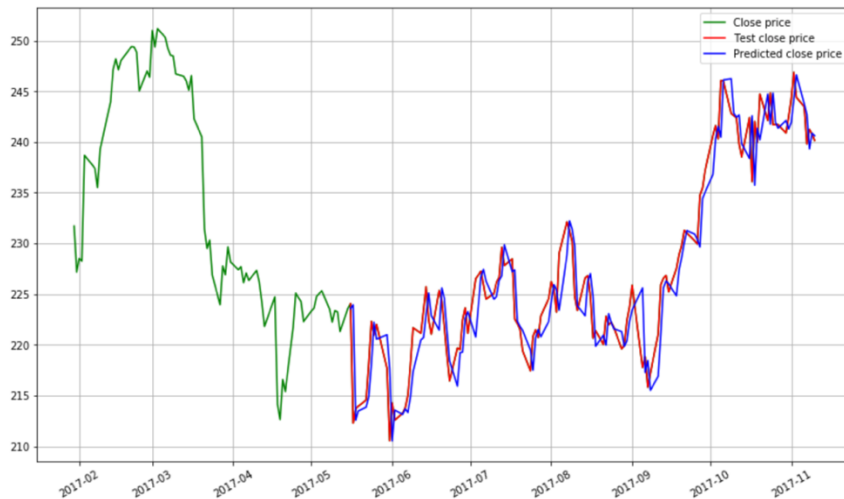
Як бачимо, значення доволі корельовані між собою. Перевіливши гіпотезу про залежність, використовуючи коефіцієнт кореляції Пірсона, отримали значення 0.997243, що свідчить про сильну лінійну залежність.

$$\rho_{X_t X_{t-1}} = \frac{Cov(X_t, X_{t-1})}{\sigma_{X_t} * \sigma_{X_{t-1}}}$$

Побудуємо модель, яка буде використовувати передбачені значення минулого кроку для прогнозування наступного.



Авторегресія це лінійна модель, що використовує зсунуті значення як вхідні. В кінці роботи розписано як працює лінійна регресія та знаходяться невідомі коефіцієнти.



Перевіримо якість передбачених результатів, використовуючи середню абсолютну та корінь середньоквадратичної помилки.

$$Mean Absolute Error = \frac{1}{m} \sum_{1}^m |Y_{true} - Y_{pred.}|$$

$$Root Mean Squared Error = \sqrt{\frac{1}{m} \sum_{1}^m (Y_{true} - Y_{pred.})^2}$$

Побудована модель має такі значення якості: $MAE = 2.17$; $RMSE = 2.86$

Оператор відставання часу

Нехай S – простір усіх послідовностей $(x_t)_{t \in \mathbb{Z}}$ дійсних чисел. Визначимо відображення $L: S \rightarrow S$, яке переводить послідовність у послідовність.

$$Lx_t = x_{t-1}$$

Варто писати $(Lx)_t = x_{t-1}$, якщо отримали послідовність $x = (x_t)_{t \in \mathbb{Z}}$ ми вводимо нову послідовність $Lx \in S$, що в момент часу t значення дорівнює початковій послідовності в момент часу $t-1$.

З цим означенням формулу авторегресії можна записати наступним чином

$$\left(1 - \sum_{k=1}^p \alpha_k L^k\right) X_t = \varepsilon_t$$

2.2 Ковзаюче середнє (Moving Average, MA)

Модель часових рядів, яка використовує середній шум, але не середнє з нульового відрізка часу до теперішнього. Замість того, використовується лише $q+1$ проміжки для передбачення t .

$$X_t = \beta_1 \varepsilon_{t-1} + \dots + \beta_n \varepsilon_{t-q} + \varepsilon_t$$

Використовуючи оператор відставання часу, можна записати так

$$X_t = \left(1 + \sum_{k=1}^q \beta_k L^k\right) * \varepsilon_t$$

2.3 Авторегресивне ковзаюче середнє (ARMA)

Це поєднання моделі авторегресії та ковзаючого середнього. У авторегресії $AR(p)$ майбутнє значення змінної вважається лінійною комбінацією p минулих спостережень та випадкової помилки.

Ковзаюче середнє $MA(q)$ використовує минулі помилки як незалежні змінні. Поєднання цих двох моделей утворює $ARMA$ модель, що визначається наступною формулою

$$\left(1 - \sum_{k=1}^p \alpha_k L^k\right) * X_t = \left(1 + \sum_{k=1}^q \beta_k L^k\right) * \varepsilon_t$$

або в повному вигляді

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_n X_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \dots + \beta_n \varepsilon_{t-q}$$

Можемо включити ненульове середнє значення в цій моделі. Якщо хочемо, щоб X_t мав середнє μ , треба мати середньо-нульове рішення Z_t

$$Z_t = \alpha_1 Z_{t-1} + \dots + \alpha_n Z_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \dots + \beta_n \varepsilon_{t-q}$$

Та взяти $X_t = Z_t + \mu$

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_n X_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \dots + \beta_n \varepsilon_{t-q} + \tilde{\mu}$$

Оператор різниці. Інтегрування

Перший оператор різниці Δ визначається за наступною формулою

$$\Delta X_t = X_t - X_{t-1} = (1 - L)X_t$$

Якщо ми візьмемо $Y_t = (1 - L)X_t$, ми можемо відновити значення X_t з Y_t інтегруванням, маючи початкову умову X_0 .

$$X_t = Y_t + X_{t-1} = Y_t + Y_{t-1} + X_{t-2} = \dots = Y_t + \dots + Y_1 + X_0$$

Другий оператор різниці Δ^2 визначається як

$$\Delta^2 X_t = (1 - L)^2 X_t$$

Припускаємо, що $Y_t = (1 - L)^2 X_t$. Тоді

$$Y_t = (1 - L)Z_t; \quad Z_t = (1 - L)X_t$$

Першим чином відновлюємо Z_t

$$Z_t = Y_t + \dots + Y_2 + Z_1, \quad \text{де } Z_1 = (1 - L)X_1 = X_1 - X_0$$

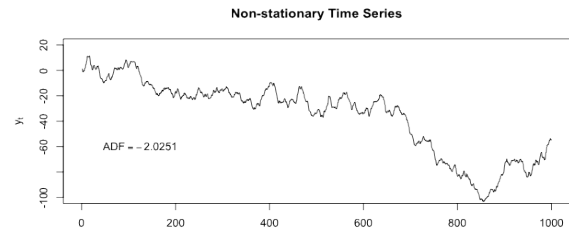
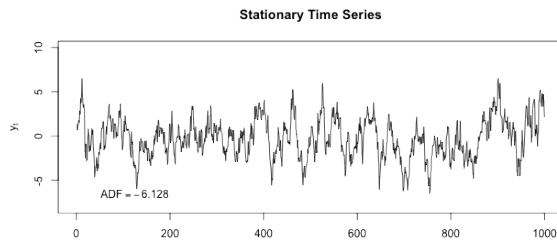
Потім відновлюємо X_t з Z_t

$$X_t = Z_t + \dots + Z_1 + X_0$$

Ці операції можуть бути узагальнені до Δ^d , де d – будь-яке натуральне число.

2.4 ARIMA

ARMA модель, описану вище, ефективно використовувати лише для стаціонарних часових рядів. Але, на практиці, більшість часових рядів демонструють нестаціонарну поведінку. Ряди, які містять тренд та сезонність також нестаціонарні. В цьому випадку недоцільно використовувати ARMA моделі, а краще використовувати узагальнену модель ARIMA.

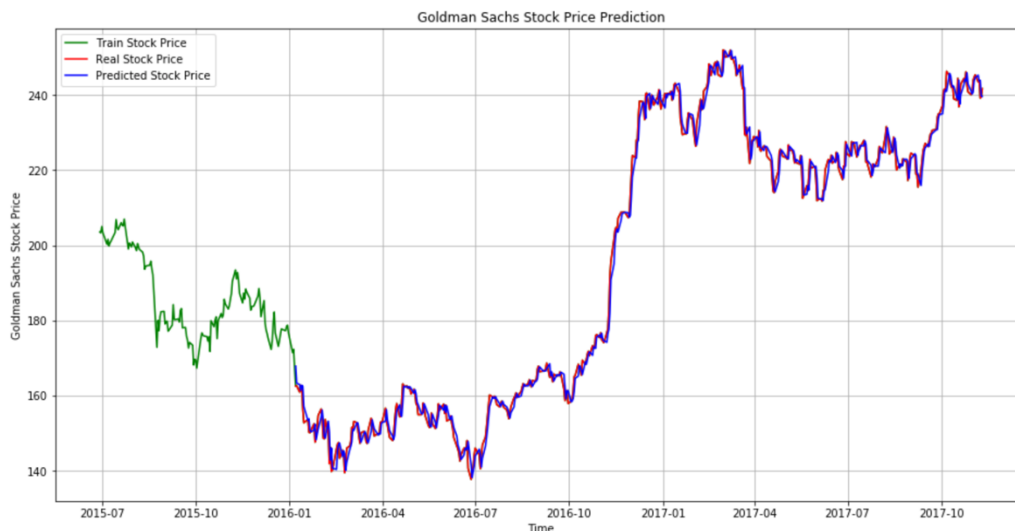


$$\left(1 - \sum_{k=1}^p \alpha_k L^k\right) * (1 - L)^d X_t = \left(1 + \sum_{k=1}^q \beta_k L^k\right) * \varepsilon_t$$

Якщо представити $Y_t = (1 - L)^d X_t$, тоді Y_t – це ARMA модель

$$\left(1 - \sum_{k=1}^p \alpha_k L^k\right) * Y_t = \left(1 + \sum_{k=1}^q \beta_k L^k\right) * \varepsilon_t$$

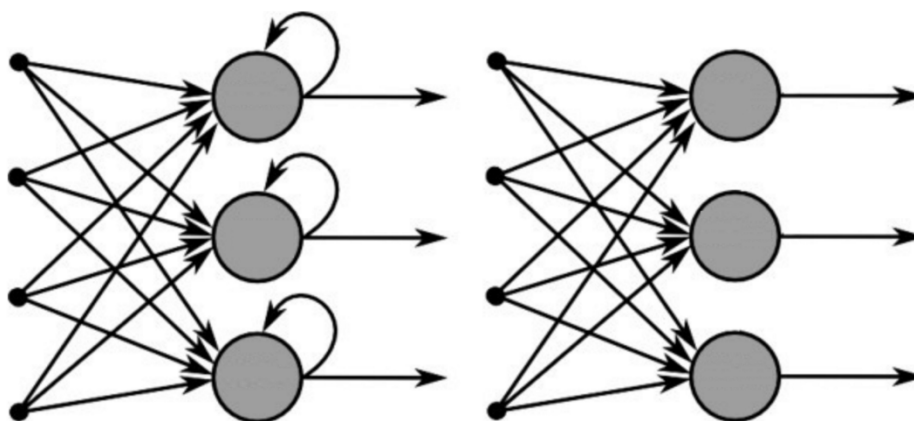
Та X_t обчислюється з Y_t за d послідовних інтегрувань. Таким чином, d – це степінь інтегрування.



Перевіримо якість передбачених результатів, використовуючи середню абсолютну та корінь середньоквадратичної помилки. Отримали наступні значення: $MAE = 2.02$; $RMSE = 2.69$. Що краще, за модель авторегресії.

2.5 Рекурентні нейронні мережі(RNN) та LSTM(Long-Short Term Memory)

Рекурентні нейронні мережі один з найперспективніших типів нейронних мереж, оскільки вони є єдиними з внутрішньою пам'яттю. Завдяки цій ознаці, RNN здатні зберігати важливу інформацію щодо вхідних даних, які вони отримали. Це є причиною, чому вони є кращим алгоритмом для послідовних даних, таких як часові ряди, мова, текст, фінансові дані, аудіо, відео та інші, оскільки вони можуть сформувати набагато глибше розуміння послідовності, порівняно з іншими алгоритмами.



Recurrent Neural Network

Feed-Forward Neural Network

Щоб описати як працюють рекурентні нейронні мережі, варто спершу описати важливі деталі про “звичайні” систему нейронна мережа прямого поширення (Feed Forward Neural Networks).

У нейронних мереж прямого поширення вхідна інформація рухається лише в одному напрямку, від вхідного шару, через декілька прихованих рівнів, до вихідного шару. Через це, вона ніколи не торкається вузлів мережі двічі.

Оскільки нейронні мережі прямого поширення розглядають лише поточний вхід, вони не мають поняття порядку в часі. Вони просто не пам'ятають нічого про те, що сталося в минулому, крім їх тренування.

У рекурентних нейронних мережах інформація проходить циклами. Приймаючи рішення, вони приймають до уваги поточний вхід, а також набуту інформацію з даних отриманих раніше. Таким чином, RNN мають два входи: минулі та теперішні дані. Це важливо, оскільки послідовність даних містить важливу інформацію про майбутнє.

Feed Forward нейронна мережа призначає, як і всі інші алгоритми глибинного навчання, вагову матрицю для вхідних даних, а потім

продукує вихід. Натомість, RNN привласнює ваги до поточного, а також попереднього входів.

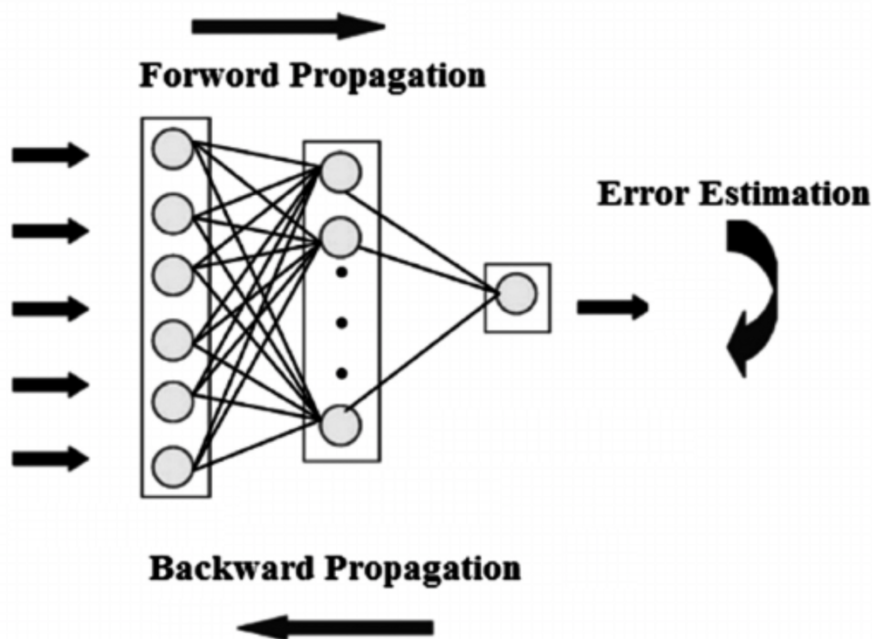
Зворотний метод поширення помилки через час (Backpropagation through time)

Щоб зрозуміти концепцію зворотного методу поширення помилки крізь час, варто спершу розібрати концепції *прямого(forward)* та *зворотного(back-propagation)* методів поширення помилки, які описані в кінці роботи.

Метод прямого поширення помилки використовується для того отримання результат моделі та перевірки результатів на правильність, з метою отримання значення помилки.

Наступним кроком, буде використання зворотного методу поширення помилки. Який є не чим іншим, як повернення назад, крізь нейронну мережу, для знаходження часткових похідних помилки відносно ваг, що дозволяє відняти ці значення від ваг.

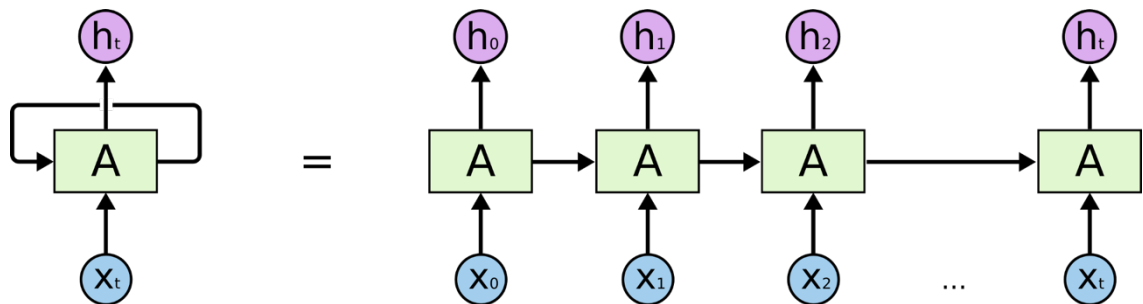
Ці похідні використовуються у методі градієнтного спуску, який описаний нижче. Таким чином регулюються ваги, в залежності від зменшення помилки. Саме так нейронні мережі навчаються під час процесу тренування. Тобто, методом зворотного поширення помилки ми намагаємося налаштувати ваги моделі під час тренування.



Зворотний метод поширення помилки скрізь час – це метод використання зворотного поширення помилки на розгорнутій нейронній мережі. Розгортання – інструмент візуалізації, який

допомагає зрозуміти, що відбувається в нейронній мережі. У більшості випадків, під час розробки нейронних мереж в популярних фрейм ворках вони автоматично піклуються про зворотне поширення помилки. Але постає проблема розуміння, що відбувається в середині мережу, з метою подальшого вирішення проблем, які виникли під час розробки.

Можна представити рекурентну нейронну мережу як послідовність мереж, які тренуються одна за одною з зворотним методом поширення помилки.



Картинка вище ілюструє представлення рекурентної нейронної мережі.

У випадку використання зворотного методу поширення помилки крізь час, необхідним є представлення рекурентної мережі як послідовності, оскільки помилка на кроці t впливає на помилку кроку $t+1$.

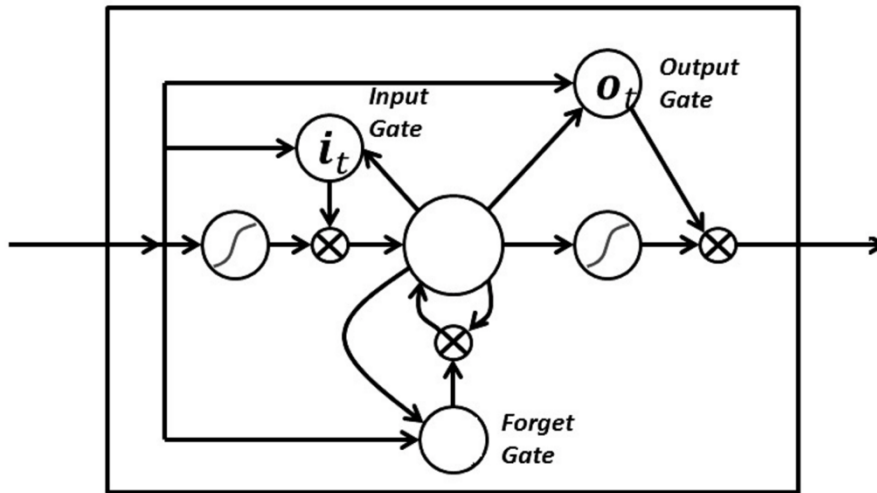
LSTM – розширення рекурентних нейронних мереж, які в основному розширюють пам'ять. Тому вони добре підходять для вивчення як короткострокових закономірностей, так і довгострокових.

Нейрони LSTM використовуються як будівельні одиниці шару рекурентної нейронної мережі, яку часто називають мережею LSTM.

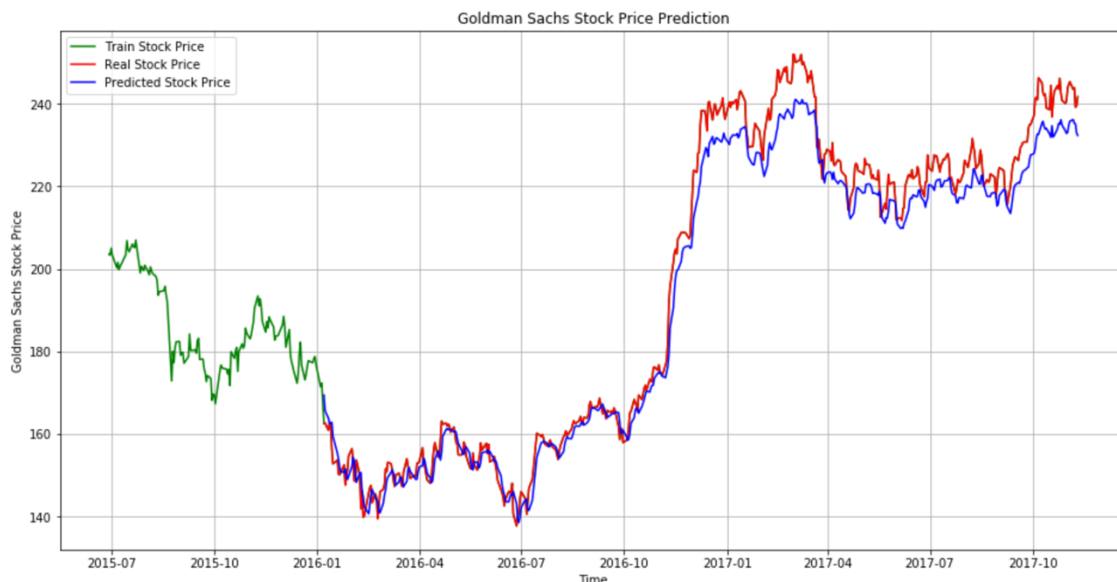
LSTM нейрони дозволяють рекурентній нейронній мережі запам'ятовувати вхідні данні на більш довгий період, оскільки вони зберігають інформацію в пам'яті, що нагадує пам'ять комп'ютера, оскільки LSTM може читати, записувати та видаляти інформацію з пам'яті.

Цю пам'ять можна розглядати як закриту комірку, що означає клітина сама вирішує зберігати чи видаляти інформацію, спираючись на приписану раніше важливість. Присвоєння важливості інформації відбувається за допомогою ваг, які також вивчаються за допомогою алгоритму. Загалом, це означає, що з часом модель дізнається, яка інформація є важливою, а яка ні.

У нейроні LSTM є три ‘вентилі’: вхід, забути, вихід. Ці ‘вентилі’ визначають, чи дозволити вхід інформації (вхід ‘вентиль’), чи видалити інформацію (забути ‘вентиль’), чи дозволити вихід (вихід ‘вентиль’).



Перевіряючи якість передбачених результатів, використовуючи вищезгадані функції якості, отримали наступні результати: : $MAE = 4.46$; $RMSE = 5.56$. Це гірші результати, за попередні моделі. Але їх можна пояснити тим, що глибокі рекурентні нейронні мережі зазвичай застосовують з значно більшими об’ємами даних. Тобто мережа не змогла достатньо навчитися.



Розділ 3

3.1 Лінійна регресія

Лінійні моделі показують взаємозв'язок між незалежними змінними та залежною змінною. ε – значення шуму, оскільки не існує точної залежності між змінними X та Y (у реальному світі принаймні).

$$Y = \theta_0 + \theta_1 X_1 + \dots + \theta_n X_n + \varepsilon$$

Ціль коефіцієнтів $\theta_0, \dots, \theta_n$ полягає в тому, щоб дозволити створити якомога краще відношення між X та Y з низьким значенням ε для ефективною для прогнозування Y для нових значень X .

Розрахунок параметрів $\theta_0, \dots, \theta_n$ зазвичай використовує метод найменших квадратів, в якому ми додаємо квадрати помилок ε . Це залишкова сума квадратів помилок, яку часто називають дельтою, оскільки вона вимірює різницю між дійсним значенням Y та прогнозованим Y_{pred} . Використовуючи $\theta_1 \dots \theta_n$.

$$\Delta = \sum_1^m (\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n - Y_{\text{true}})^2$$

Отже, маємо мінімізувати дельту. Вводимо поняття функцію втрат, яку будемо використовувати для вимірювання похибки моделі.

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_1^m (\theta_0 + \theta_1 X_1^i + \dots + \theta_n X_n^i - Y_{\text{true}}^i)^2$$

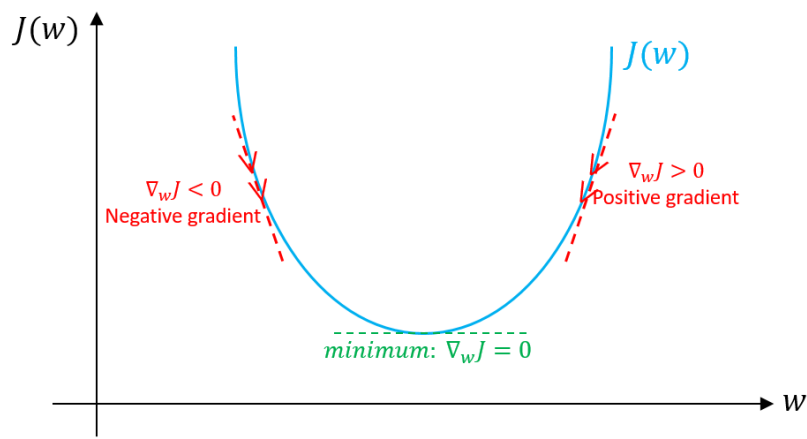
Для покращення передбачень необхідно мінімізувати функцію втрат. Використаємо алгоритм градієнтного спуску.

Спершу оберемо якісь значення параметрів $\theta_0, \dots, \theta_n$, а потім повторюємо до збіжності:

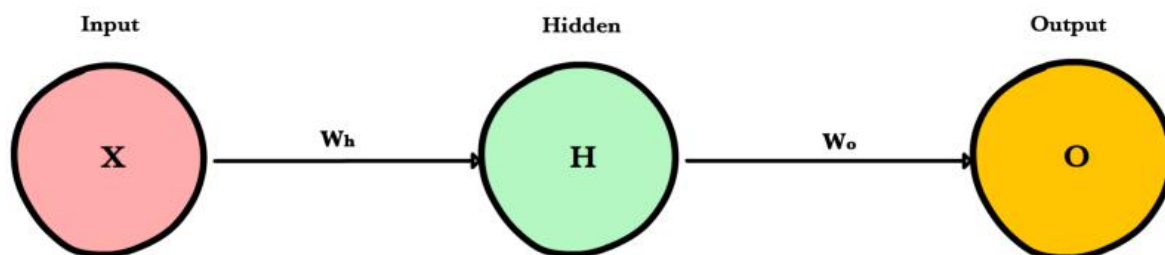
$$\begin{cases} \theta_0 = \theta_0 - \alpha * \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1, \dots, \theta_n) \\ \theta_1 = \theta_1 - \alpha * \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1, \dots, \theta_n) \\ \dots \\ \theta_n = \theta_n - \alpha * \frac{\partial}{\partial \theta_n} J(\theta_0, \theta_1, \dots, \theta_n) \end{cases}$$

Після математичних операцій диференціювання отримаємо

$$\begin{cases} \theta_0 = \theta_0 - \alpha * \frac{1}{m} \sum_1^m (\theta_0 + \theta_1 X_1^i + \dots + \theta_n X_n^i - Y_{true}^i) \\ \theta_1 = \theta_1 - \alpha * \frac{1}{m} \sum_1^m (\theta_0 + \theta_1 X_1^i + \dots + \theta_n X_n^i - Y_{true}^i) * X_1^i \\ \dots \end{cases}$$



3.2 Метод прямого поширення помилки (forward propagation)



Вхідні дані проходять шар за шаром до кінцевого шару, який виводить передбачення. Просту нейронну мережу можна розглядати як послідовність вкладених функцій. Для нейронної мережі зображеної вище, пряме поширення помилки виглядає наступним чином:

$$A(A(X * W_h) * W_o)$$

A - функція активації (залежність вихідного сигналу штучного нейрону від вхідних даних).

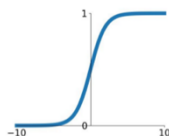
X – вхідні дані

W_h та W_o – ваги для прихованого та вихідного рівнів, відповідно.

Функції активації

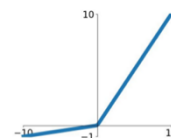
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



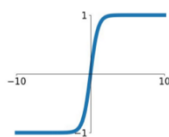
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

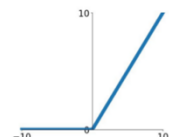


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

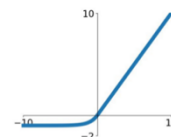
ReLU

$$\max(0, x)$$



ELU

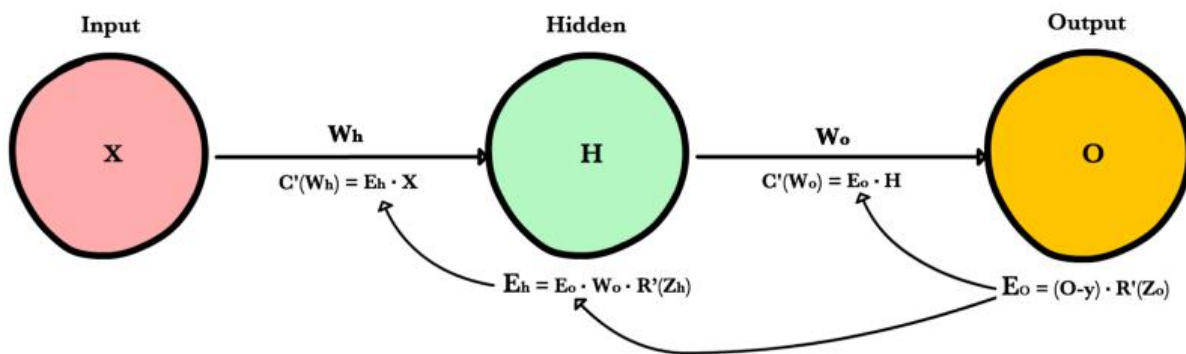
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



3.3 Метод зворотного поширення помилки (backpropagation)

Метод полягає в ітеративному регулюванні вагів пропорційно до того, наскільки вони впливають на загальну помилку моделі. Зрештою, коли помилка зведена до мінімуму, маємо ряд ваг, які дають хороші прогнози.

Як було показано вище, метод прямого поширення помилки – послідовність вкладених функцій. Таким чином, метод зворотного поширення помилки можна розглядати, як застосування правила ланцюга, щоб знайти похідну втрат відносно будь-якої ваги в мережі. Це відображує наскільки кожна вага сприяє загальній помилці, і напрям до оновлення ваг з метою зменшення помилки. Потім помилка мінімізується методом градієнтного спуску.



Для прикладу визначимо необхідні функції

$$\text{Зважені вхідні дані: } Z = X * W \Rightarrow Z'(X) = W$$

$$\text{ReLU функція активації: } R = \max(0, Z) \Rightarrow R'(Z) = \begin{cases} 0, & Z < 0 \\ 1, & Z > 0 \end{cases}$$

$$\text{Функція втрат: } C = \frac{1}{2}(\hat{y} - y)^2 \Rightarrow C'(\hat{y}) = (\hat{y} - y)$$

Першим чином рахуємо помилку вихідного рівня. Потім передаємо цю помилку на попередній рівень, де знову рахуємо помилку та передаємо на попередній рівень і так далі. На кожному рівні рахуємо похідну функції втрат, враховуючи ваги цього рівня. Ця похідна вказує на напрям в якому покращувати ваги з метою мінімізації загальної помилки.

Висновки

В цій роботі розглянуто основні методи передбачення ціни цінних паперів, а саме: авторегресії, ARIMA, а також рекурентну нейронну мережу та оцінено передбачення цих методів за допомогою функцій якості: середньої абсолютної та кореня середньоквадратичної помилки.

Отримані результати досліджень у цій роботі можна використовувати у побудові торгівельних ботів, іншому напрямку кількісних інвестиційних стратегій.

Список використаної літератури

1. Quantitative Finance for dummies by Steve Bell
2. Quantitative Equity Investing: Techniques and Strategies by Frank J.Fabozzi, Sergio M.Focardi, Petter N.Kolm
3. Big Data and Machine Learning in Quantitative Investment by Tony Guida
4. An Introductory Study on Time Series Modeling and Forecasting by Ratnadip Adhikari R. K. Agrawal
5. Deep Learning with Python by François Chollet
6. <https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/>
7. <https://www.datacamp.com/community/tutorials/lstm-python-stock-market>
8. <https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5>
9. <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
10. <https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/>
11. <http://users.dma.unipi.it/~flandoli/AUTCap4.pdf>
12. Deep Time Series Forecasting with Python by Dr. N.D Lewis
13. Time Series for Macroeconomics and Finance by John H. Cochrane
14. <https://www.investopedia.com/terms/r/rsi.asp>
15. <https://www.investopedia.com/terms/b/bollingerbands.asp>
16. <https://www.investopedia.com/terms/a/aroonoscillator.asp>
17. <https://www.investopedia.com/terms/s/stochasticoscillator.asp>
18. <https://www.investopedia.com/terms/p/pricerateofchange.asp>
19. <https://www.investopedia.com/terms/v/vwap.asp>
20. <https://www.investopedia.com/articles/technical/081501.asp>
21. <https://www.investopedia.com/terms/c/commoditychannelindex.asp>
22. <http://growingscience.com/beta/dsl/1868-forecasting-returns-on-a-stock-market-using-artificial-neural-networks-and-garch-family-models-evidence-of-stock-market-s-p-500.html>