КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра технологій управління

Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

"Аналіз і прогнозування трендів валютних ринків методами машинного навчання"

Студента 2-го курсу групи ІАВ-21

Борисова Віталія Євгеновича
(прізвище, ім'я, по батькові)

(підпис студента)

Науковий керівник:

Доктор технічних наук
(науковий ступінь, вчене звання)

Осауленко Ігор Анатолійович
(прізвище, ім'я, по батькові)

(дата) (підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри технологій управління

(підпис) (прізвище, ініціали) (дата)

Київ – 2023

ЗМІСТ

АНОТАЦІЯ **3**
ВСТУП **5**
РОЗДІЛ 1 **8**
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ **8**
1.1 Поняття валютного ринку 8
1.2 Основні характеристики валютного ринку 18
РОЗДІЛ 2 **22**
ПРОГНОЗУВАННЯ І ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ **22**
2.1 Поняття прогнозування 22
2.2 Існуючі інструменти прогнозування 29
РОЗДІЛ 3 **32**
МОДЕЛІ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ВАЛЮТНИХ РИНКІВ **32**
3.1 Аналіз даних 32
3.2 Нейронні мережі 44
3.3 Інструментарій прогнозування валютних ринків 62
РОЗДІЛ 4 **69**
РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ ВАЛЮТНИХ РИНКІВ **69**
4.1 Аналіз варіантів використання системи 69
4.2 Діаграма IDEF0 72
4.3 Діаграма компонентів 73
4.4 Розробка основних механізмів 75
4.5 Створення навчальних вибірок 78
4.6 Тестування 80
ВИСНОВКИ **83**
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ **84**

АНОТАЦІЯ

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 – Комп’ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

Дипломна робота магістранта Борисова Віталія Євгеновича.

Тема роботи – «Аналіз і прогнозування трендів валютних ринків методами машинного навчання».

Мета дипломної роботи магістра – проаналізувати валютні ринки та розробити модель прогнозування трендів за допомогою методів машинного навчання.

Об’єкт дослідження: курси валют.

Предмет дослідження. Математичні методи моделювання та прогнозування трендів валютних ринків, їх аналіз та оцінка.

Наукова новизна полягає у визначенні важливих аспектів функціонування валютних ринків та основних чинників впливу на сучасні тренди, застосуванні алгоритмів машинного навчання для аналізу великої кількості даних, розробці ефективної моделі машинного навчання, що допомагає виявляти складні зв'язки та тренди.

Були використані статистичні методи для збору та обробки даних, а також системні та комплексні способи аналізу та прогнозування цін. Для прогнозування було використано метод дерева рішень. Робота складається зі вступу, чотирьох розділів, висновків та списку використаних джерел. Робота налічує 93 сторінок, перелік з 40 джерел.

Ключові слова: моделювання, аналіз даних, аналіз валютного ринку, машинне навчання.

ВСТУП

Актуальність роботи. У сучасному світі валютні ринки є ключовими складовими глобальної економіки. Їх динаміка та тренди мають велике значення для багатьох галузей, включаючи фінансові установи, міжнародну торгівлю та інвестиції. Аналіз та прогнозування трендів валютних ринків є складним завданням, оскільки вони підпорядковуються впливу різних факторів, таких як політична ситуація, макроекономічні показники та світові події.

З метою вдосконалення процесу аналізу та прогнозування трендів валютних ринків, методи машинного навчання стали все більш популярними. Ці методи дозволяють автоматизувати процес обробки та аналізу великої кількості даних, виявляти складні залежності та шаблони, які можуть бути важкі для виявлення людським розумом.

Метою роботи є дослідження та розробка ефективних моделей, заснованих на методах машинного навчання, для аналізу та прогнозування трендів на валютних ринках.

Основні цілі роботи включають:

1. Аналіз валютних ринків: Проведення комплексного аналізу валютних ринків, їх динаміки та трендів. Вивчення впливу різних факторів, таких як економічні показники, політична ситуація та глобальні події, на коливання валютних курсів.
2. Застосування методів машинного навчання: Використання різноманітних методів машинного навчання, таких як нейронні мережі, регресійні моделі та алгоритми кластеризації, для аналізу великих обсягів даних з валютних ринків. Розробка моделей, які здатні виявляти складні залежності та шаблони в даних та прогнозувати майбутні тренди.

3. Ефективність та точність прогнозів: Оцінка ефективності розроблених моделей на основі реальних даних валютних ринків. Визначення точності прогнозів та порівняння їх з існуючими підходами до прогнозування валютних трендів.
4. Практичні застосування: Дослідження можливості використання отриманих результатів для прийняття рішень на валютних ринках. Розгляд можливих сценаріїв застосування прогнозів для трейдерів, інвесторів та фінансових аналітиків з метою покращення прибутковості та зниження ризиків.

Об’єктом дослідження коливання курсів валют.

Предметом дослідження є математичні методи моделювання і прогнозування трендів валютних ринків, їх аналіз та оцінка.

Методологія та методи дослідження. Методологічною основою дослідження стали теоретичні напрацювання українських та зарубіжних експертів із питань, пов'язаних з тенденціями розвитку валютного ринку. Для дослідження теоретичних питань використовувалися методи дедукції та індукції, аналізу та синтезу. Для аналізу та прогнозування використовувалися статистичні методи збору та обробки даних. Для прогнозування використовувався метод дерева рішень.

Теоретична, методична та практична значущість роботи полягає у визначенні важливих аспектів функціонування валютних ринків. Із отриманих результатів, фахівці зможуть краще планувати фінанси, інвестиції, проводити економічний аналіз ринку. Практична значущість відображається у можливості використання моделі машинного навчання з метою впровадження її на підприємстві як допоміжного інструменту аналізу даних.

Наукова новизна полягає в аналізі великої кількості даних про тренди валютних ринків за допомогою алгоритмів машинного навчання,

розробці ефективної моделі машинного навчання, що допомагає виявляти складні зв'язки та тренди. Зокрема, було побудовано модель на основі дерева прийняття рішень.

Інформаційну базу дослідження склали статті вітчизняних та зарубіжних науковців і практиків, статистичні дані зі спеціалізованих інтернет-журналів та наукових періодичних видань.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття валютного ринку

Валютний ринок (FX, forex або валютний ринок) — це глобальний позабіржовий або децентралізований (OTC) ринок для торгівлі валютами. Цей ринок визначає обмінні курси валют. Він включає всі аспекти операцій із валютами за поточними або визначеними цінами. З точки зору обсягу торгів, це, безперечно, найбільший ринок у світі, за яким слід кредитний ринок.[1]

Основні учасники такого ринку — великі міжнародні банки. Фінансові центри в усьому світі цілодобово, за винятком вихідних, функціонують як якорі торгівлі між різними типами покупців і продавців. Валютний ринок не визначає абсолютну вартість валюти, оскільки валюти завжди торгуються парами, а скоріше визначає її відносну вартість, встановлюючи ринкову ціну однієї валюти за умови оплати іншою. Наприклад, 1 долар США коштує m UAH, k JPY, або t CAD тощо, де m, k, t — курси відповідних валют до долара.

Валютний ринок функціонує на різних рівнях через фінансові установи. За лаштунками банки спілкуються з обраною групою фінансових установ, відомих як "дилери", які беруть участь у торгівлі іноземною валютою у великих обсягах. Більшість торговців іноземною валютою є банками, тому цей закулісний ринок іноді називають «міжбанківським ринком» (хоча в ньому задіяно декілька страхових

компаній та інших фінансових компаній). Обсяг торгів між валютними дилерами може бути величезним, включаючи сотні мільйонів доларів. Через проблему суверенітету, що стосується двох валют, у Forex мало (якщо взагалі є) наглядового органу, який регулює його дії.

Дозволяючи конвертацію валют, валютний ринок підтримує міжнародну торгівлю та інвестиції. Наприклад, він дозволяє американським компаніям імпортувати продукцію з країн-членів ЄС, особливо тих, що входять до єврозони. Американські компанії можуть імпортувати продукцію з країн-членів ЄС, особливо тих, що входять до єврозони, і платити в євро, навіть якщо їхній дохід виражений у доларах США. Крім того, це заохочує прямі спекуляції та оцінки щодо вартості валют і спекуляцій керрі-трейду на основі різниці процентних ставок між двома валютами.[2]

У типовій валютній операції сторона купує певну кількість однієї валюти, сплачуючи деякою кількістю іншої валюти.

У 1970-х роках почав розвиватися сучасний валютний ринок, де грошові одиниці обмінюються за допомогою інших валют. Цей розвиток став можливим після тривалого періоду, коли уряди здійснювали обмеження на операції з іноземною валютою в рамках Бреттон-Вудської системи монетарного управління. Ця система встановлювала правила для комерційних і фінансових відносин між найбільшими промисловими державами після Другої світової війни. Поступово країни переходили до плаваючих обмінних курсів, відміняючи попередній режим фіксованого обмінного курсу, який діяв за Бреттон-Вудською системою.

Валютний ринок є унікальним завдяки таким характеристикам:

- обсяг торгів, який є найбільшим класом активів у світі та забезпечує високу ліквідність;

- географічна розпорошеність;

- безперервна робота: 24 години на добу, крім вихідних, тобто торгівля з 22:00 UTC у неділю (Сідней) до 22:00 UTC п’ятниці (Нью-Йорк);

- різноманітність факторів, що впливають на валютні курси;

- низькі норми відносного прибутку порівняно з іншими ринками фіксованого доходу; і

- використання кредитного плеча для збільшення маржі прибутків і збитків, а також щодо розміру рахунку.

Розмір і ліквідність ринку

Валютний ринок є найбільш ліквідним фінансовим ринком у світі. Трейдерами є уряди та центральні банки, комерційні банки, інші інституційні інвестори та фінансові установи, валютні спекулянти, інші комерційні корпорації та окремі особи. Відповідно до Трирічного огляду центральних банків за 2019 рік, координованого Банком міжнародних розрахунків, середній щоденний оборот становив 7,5 трильйона доларів у квітні 2022 року (порівняно з 1,9 трильйона доларів у 2004 році).[3] З цих 6,6 трильйонів доларів США 2,1 трильйона припадали на спотові операції, а 5,4 трильйона доларів – на прямі форварди, свопи та інші похідні інструменти.

Операції з іноземною валютою проводяться на ринку, який не є офіційною біржею, але відомий як позабіржовий ринок. На цьому ринку брокери і дилери взаємодіють безпосередньо між собою, без участі централізованої біржі або розрахункової палати. Головним географічним центром цієї торгівлі є Велика Британія, зокрема місто Лондон. У квітні 2022 року торгівля у Сполученому Королівстві становила 38,1% від загального обсягу, що зробило це місто найважливішим центром торгівлі іноземною валютою у світі. У зв'язку з домінуванням Лондона на ринку, котирувальна ціна певної валюти зазвичай є ціною Лондонського ринку. Наприклад, коли Міжнародний валютний фонд щодня розраховує вартість своїх спеціальних прав запозичення, він використовує ціни Лондонського ринку опівдні цього дня. Торгівля в Сполучених Штатах склала 19,4%, на Сінгапур і Гонконг припадає 9,4% і 7,1% відповідно, а на Японію припадає 4,4% [3].

За період від 2004 до 2013 року, обсяг торгівлі біржовими валютними ф'ючерсами і опціонами швидко збільшувався, досягнувши рекордних 145 мільярдів доларів у квітні 2013 року. (удвічі більше, ніж у квітні 2007 року).[57] Станом на квітень 2022 року біржові валютні деривативи становлять 2% від загального обсягу позабіржової торгівлі валютами. Ф'ючерсні контракти на іноземну валюту були введені в 1972 році на Чиказькій товарній біржі і на сьогоднішній день є одними з найбільш популярних ф'ючерсних контрактів, які торгуються.

Більшість розвинених країн дозволяють торгувати похідними продуктами, такими як ф'ючерси та опціони на ф'ючерси, на своїх біржах. У цих країнах капітальні рахунки повністю конвертовані. Уряди деяких країн з ринками, що розвиваються, не дозволяють валютні деривативи на своїх біржах, оскільки контролюють рух капіталу. Однак використання деривативів зростає в багатьох країнах з економікою, що розвивається.[58]

Наприклад, Південна Корея, Південна Африка та Індія створили валютні ф'ючерсні біржі, незважаючи на обмеження на рух капіталу.

Обсяг торгівлі іноземною валютою зросла на 20% у період з квітня 2007 року по квітень 2010 року, а з 2004 року подвоїлася.[59] Збільшення товарообігу зумовлене низкою факторів: зростанням значення іноземної валюти як класу активів, збільшенням торгової активності високочастотних трейдерів і появою роздрібних інвесторів як важливого сегмента ринку. Зростання електронного виконання та різноманітний вибір місць виконання знизили транзакційні витрати, підвищили ліквідність ринку та залучили більшу участь багатьох типів клієнтів. Зокрема, електронні торгівлі через онлайн-портали полегшили роздрібним торговцям торгівлю на валютному ринку. До 2010 року роздрібна торгівля, за оцінками, становила до 10% спотового обороту, або 150 мільярдів доларів США на день.

Учасники ринку

Більшість розвинених країн мають можливість торгувати похідними фінансовими інструментами, такими як ф'ючерси та опціони на ф'ючерси, на своїх валютних біржах. У цих країнах капітальні рахунки вже повністю конвертовані. Однак, деякі країни з ринками, які знаходяться у стані розвитку, не дозволяють торгувати валютними похідними інструментами на своїх біржах, оскільки уряди контролюють рух капіталу. Незважаючи на це, використання таких похідних інструментів зростає в багатьох країнах з економікою, що знаходиться у стані розвитку. Наприклад, Південна Корея, Південна Африка та Індія створили свої власні валютні ф'ючерсні біржі, незважаючи на обмеження щодо руху капіталу. Торгівля іноземною валютою значно зросла на 20% в період від квітня 2007 року до квітня 2010 року і подвоїлася з 2004 року.

За словами Галаті та Мелвіна, «з початку 2000-х років пенсійні фонди, страхові компанії, взаємні фонди та інші інституційні інвестори відіграють дедалі важливішу роль на фінансових ринках загалом і на валютних ринках зокрема». (2004) Крім того, він зазначає, що «за період 2001–2004 років хедж-фонди помітно зросли як з точки зору кількості, так і загального розміру».[62] Центральні банки також беруть участь у валютному ринку, щоб узгодити валюти зі своїми економічними потребами.

Комерційні компанії

Фінансова діяльність компаній, які шукають іноземну валюту для оплати товарів або послуг, є важливою складовою валютного ринку. Торговельні операції комерційних компаній зазвичай мають незначний обсяг порівняно з операціями банків або спекулянтів, і їхні вплив на ринкові ставки зазвичай є короткостроковим. Однак, торговельні потоки відіграють важливу роль у визначенні довгострокового курсу валют. Деякі транснаціональні корпорації можуть мати непередбачуваний вплив, коли здійснюють великі операції, пов'язані з ризиками, про які інші учасники ринку не мають повної інформації.

Центральні банки

Центральні банки країн мають значний вплив на валютні ринки і виконують важливу роль. Вони прагнуть контролювати грошову масу, інфляцію та/або процентні ставки, і часто встановлюють офіційні або неофіційні цільові ставки для своїх валют. Центральні банки можуть використовувати свої значні резерви валюти для стабілізації ринку. Проте, ефективність так званих "стабілізуючих спекуляцій" центрального банку є питанням у сумніві, оскільки центральні банки не можуть збанкрутувати, як це може статися з іншими трейдерами. Крім того, немає переконливих

доказів того, що центральні банки дійсно отримують прибуток від своїх торгівельних операцій.

Валютний фіксінг

Валютний фіксінг — це денний грошовий обмінний курс, що фіксується національним банком кожної країни. Ідея полягає в тому, що центральні банки використовують час фіксування та обмінний курс для оцінки поведінки своєї валюти. Фіксовані валютні курси відображають реальну вартість рівноваги на ринку. Банки, дилери та трейдери використовують фіксовані курси як індикатори ринкових тенденцій.

Просте очікування або слухи про можливу валютну інтервенцію з боку центрального банку можуть впливати на стабілізацію валюти. У країнах з режимом брудної плаваючої валюти можуть вживатися активні заходи для втручання на ринок, і такі втручання можуть мати місце кілька разів на рік. Проте, центральні банки не завжди досягають своїх цілей. Об’єднані ресурси ринку можуть легко перевищити можливості будь-якого центрального банку. В минулому такі сценарії спостерігалися під час злету і падіння Європейського механізму валютних курсів у 1992–1993 роках, а також під час кризи в Азії.

Фірми з управління інвестиціями

Компанії, що займаються управлінням інвестиціями, активно використовують валютний ринок для здійснення операцій з іноземними цінними паперами. Ці компанії діють від імені своїх клієнтів, таких як пенсійні фонди та благодійні фонди, і мають значні обсяги ресурсів під управлінням. Вони використовують валютний ринок для полегшення обміну та купівлі-продажу іноземних цінних паперів, що дозволяє їм розширити можливості інвестування та управління портфелем активів. Це

дозволяє їм диверсифікувати ризики та забезпечити максимальну ефективність управління інвестиціями своїх клієнтів. Наприклад, інвестиційний менеджер, який має міжнародний портфель акцій, повинен придбати та продати кілька пар іноземних валют, щоб оплатити придбання іноземних цінних паперів.

Деякі фірми з управління інвестиціями також мають більш спекулятивні спеціалізовані валютні операції, які керують валютними ризиками клієнтів з метою отримання прибутку, а також обмеження ризику. Хоча кількість спеціалізованих фірм цього типу досить мала, багато з них мають велику вартість активів під управлінням і, отже, можуть генерувати великі угоди.

Роздрібні торговці іноземною валютою

Індивідуальні роздрібні спекулятивні трейдери становлять зростаючий сегмент цього ринку. Зараз вони беруть участь опосередковано через брокерів або банки. Роздрібні брокери, які в основному контролюються та регулюються в США Комісією з торгівлі товарними ф'ючерсами та Національною асоціацією ф'ючерсів, раніше періодично піддавалися шахрайству на валютних ринках.[64][65] Щоб вирішити цю проблему, у 2010 році NFA вимагала від своїх членів, які працюють на ринках Forex, зареєструватися як такі (тобто Forex CTA замість CTA). Ті учасники NFA, які традиційно підпадають під вимоги до мінімального чистого капіталу, FCM та IB, підпадають під більші вимоги до мінімального чистого капіталу, якщо вони працюють на Форекс. Декілька валютних брокерів працюють у Великій Британії відповідно до правил Управління фінансовими послугами включає торгівлю іноземною валютою з використанням маржі і є частиною широкої позабіржової індустрії

торгівлі деривативами. Цей сегмент ринку включає контракти на різницю та ставки на фінансовий спред.

На роздрібному ринку валютних брокерів можна виділити два основних типи: брокери і дилери або маркет-мейкери. Брокери діють як посередники між клієнтами та загальним валютним ринком. Вони виконують роздрібні замовлення валютних операцій та намагаються знайти найкращі ціни на ринку для своїх клієнтів. Брокери отримують комісію або додаткову винагороду за свої послуги, яка додається до ціни, отриманої на ринку.

Дилери або маркет-мейкери, навпаки, зазвичай діють як принципали в угоді проти роздрібного клієнта та пропонують ціну, за якою вони готові мати справу.

Небанківські валютні компанії

Небанківські обмінні компанії надають послуги обміну валюти та міжнародних платежів як приватним особам, так і компаніям. Їх також називають "валютними брокерами", проте вони відрізняються від традиційних брокерів тим, що їх основна діяльність - це обмін валюти з проведенням платежів. Це означає, що зазвичай відбувається фізична доставка валюти на банківський рахунок, а не спекулятивна торгівля.

За оцінками, у Великій Британії 14% валютних переказів/платежів здійснюються через компанії з обміну іноземної валюти.[66] Перевага цих компаній зазвичай полягає в тому, що вони пропонують кращі обмінні курси або дешевші платежі, ніж банк клієнта.[67] Ці компанії відрізняються від компаній грошових переказів/переказів тим, що вони зазвичай пропонують послуги вищої вартості. Щодня обсяг транзакцій, які проводяться через валютні компанії в Індії, становить приблизно 2

мільярди доларів США [68]. Хоча цей обсяг не конкурує з великими та добре відомими міжнародними валютними ринками, з появою онлайн-валютних компаній ринок постійно зростає. Небанківські валютні компанії здійснюють приблизно 25% валютних переказів та платежів в Індії.[69] Більшість із цих компаній використовують USP кращих обмінних курсів, ніж банки. Вони регулюються FEDAI, і будь-які транзакції в іноземній валюті регулюються Законом про управління валютними операціями 1999 року (FEMA).

Компанії з грошових переказів і обмінні пункти

Основні статті: Платіжна система та обмінні пункти

Компанії з грошових переказів/компанії, що здійснюють грошові перекази, здійснюють великі перекази на невеликі суми, як правило, економічними мігрантами до своєї країни. У 2007 році Aite Group оцінила грошові перекази на суму 369 мільярдів доларів США (зростання на 8% порівняно з попереднім роком). Чотири найбільші зовнішні ринки (Індія, Китай, Мексика та Філіппіни) отримують 95 мільярдів доларів. Найбільшим і найвідомішим постачальником є Western Union з 345 000 агентами по всьому світу, за яким слід UAE Exchange. Обмінні пункти або компанії з переказу валюти надають мандрівникам послуги з обміну невеликої іноземної валюти. Зазвичай вони розташовані в аеропортах і на вокзалах або в туристичних місцях і дозволяють обмінювати фізичні купюри з однієї валюти на іншу. Вони отримують доступ до валютних ринків через банки або небанківські валютні компанії.

1.2 Основні характеристики валютного ринку

У режимі фіксованого обмінного курсу валютні курси встановлюються урядом, а для пояснення (і прогнозування) коливань обмінних курсів у режимі плаваючого обмінного курсу запропоновано ряд теорій, зокрема:

- Умови міжнародного паритету включають паритет відносної купівельної спроможності, паритет процентних ставок, внутрішній ефект Фішера та міжнародний ефект Фішера. Ці теорії певною мірою дають логічне пояснення коливань обмінного курсу, але вони є хибними, оскільки ґрунтуються на сумнівних припущеннях (наприклад, необмежений рух товарів, послуг та капіталу), які рідко відповідають дійсності в реальному світі.
- Модель платіжного балансу зосереджується насамперед на ринкових товарах і послугах, ігноруючи зростаючу важливість глобальних потоків капіталу. Вона не змогла пояснити постійне зростання курсу долара США протягом 1980-х і більшої частини 1990-х років, незважаючи на значний дефіцит поточного рахунку платіжного балансу США.

- Валюта розглядається як ключовий клас активів у моделі ринку активів, який використовується для побудови інвестиційних портфелів. Бажання людей володіти певною кількістю активів значною мірою залежить від їхніх очікувань щодо потенційної майбутньої вартості активів. Згідно з концепцією обмінного курсу на ринку активів, "обмінний курс між двома валютами - це просто ціна, яка врівноважує відносний попит і пропозицію на активи, деноміновані в цих валютах".

Жодна з розроблених на даний момент моделей не може пояснити обмінні курси та волатильність у більш тривалих часових рамках. Для більш коротких часових рамок (менше кількох днів) можна розробити алгоритми для прогнозування цін. З наведених вище моделей зрозуміло, що багато макроекономічних факторів впливають на обмінні курси і, зрештою, ціни на валюту є результатом подвійних сил попиту та пропозиції. Світові валютні ринки можна розглядати як величезний плавильний котел: у великій і постійно мінливій суміші поточних подій фактори попиту та пропозиції постійно змінюються, і відповідно змінюється ціна однієї валюти по відношенню до іншої. Жоден інший ринок не охоплює (і дистилює) стільки того, що відбувається у світі в будь-який момент часу, як іноземна валюта.[71]

На попит і пропозицію будь-якої валюти, а отже, і на її вартість, впливає не один елемент, а декілька. Ці елементи зазвичай діляться на три категорії: економічні фактори, політичні умови та ринкова психологія.

Економічні фактори

Економічні фактори включають (а) економічну політику уряду та центральних банків, (б) економічні умови, що розкриваються в економічних звітах, та інші економічні показники.

- Фіскальна політика (бюджет/видатки) та монетарна політика (механізм, за допомогою якого центральний банк впливає на пропозицію та "вартість" грошей, виражену у відсоткових ставках) є складовими економічної стратегії уряду.

- Ринок часто негативно реагує на збільшення дефіциту державного бюджету і позитивно - на його зменшення. Вартість національної

валюти відображає цей ефект.

- Рівень і тенденції торговельного балансу показують попит на товари та послуги, що, в свою чергу, виявляє потребу у валюті певної країни для здійснення торгівлі. Конкурентоспроможність економіки країни відображається в профіциті та дефіциті торгівлі товарами та послугами. Наприклад, дисбаланс торгівлі може негативно вплинути на вартість національної валюти.

- Темпи та характер інфляції: Як правило, валюта знецінюється, коли спостерігається значна інфляція або коли є підстави вважати, що інфляція зростає. Це пов'язано з тим, що інфляція знижує купівельну спроможність і, як наслідок, попит на конкретну валюту. Очікування того, що центральний банк підвищить короткострокові процентні ставки, щоб протидіяти зростаючій інфляції, може призвести до підвищення курсу валюти під час зростання інфляції.

- Економічне зростання та здоров'я: Звіти про ВВП, зайнятість, роздрібні продажі, завантаження виробничих потужностей та інші показники надають інформацію про те, наскільки добре розвивається економіка країни. Загалом, валюта країни матиме кращий курс і більший попит, коли її економіка буде здоровішою і сильнішою.

- Економічна продуктивність: Валюта країни має зростати в ціні, коли її економіка стає більш продуктивною. Якщо зростання відбувається в торговельному секторі, його наслідки будуть більш

чевидними.[72]

Політичні умови

Внутрішні, регіональні та міжнародні політичні умови та події можуть мати глибокий вплив на валютні ринки.

Усі обмінні курси чутливі до політичної нестабільності та очікувань щодо нової правлячої партії. Політичні потрясіння та нестабільність можуть мати негативний вплив на економіку країни. Наприклад, політичні турбуленції та нестабільність урядів у Пакистані та Таїланді можуть мати негативний вплив на курс їхніх валют. Подібним чином у країні, яка переживає фінансові труднощі, зростання політичної фракції, яка вважається фінансово відповідальною, може мати протилежний ефект. Крім того, події в одній країні в регіоні можуть підштовхнути позитивний/негативний інтерес до сусідньої країни та, у процесі, вплинути на її валюту.

РОЗДІЛ 2

ПРОГНОЗУВАННЯ І ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ

2.1 Поняття прогнозування

Прогнозування включає розробку прогнозу, яке у вузькому сенсі передбачає проведення спеціальних наукових досліджень для прогнозування подальшого розвитку певного процесу. Потреба в прогнозуванні пояснюється бажанням отримати інформацію про майбутні події, хоча це неможливо здійснити абсолютно точно через статистичні (помилки в поточних оцінках), ймовірнісні (багатоваріантність наслідків), емпіричні (помилки в методології моделей) та філософські (обмеження наших знань) принципи. Точність будь-якого прогнозу залежить від таких факторів:

- Об'єму "істинних" (перевірених з відомою похибкою) вихідних даних та тривалості їх збору.
- Об'єму неперевірених вихідних даних та тривалості їх збору.
- Властивостей об'єкта прогнозування та системи його взаємодії з особою, що прогнозує.
- Методик і моделей прогнозування. Зі зростанням кількості факторів, що впливають на точність прогнозу, він практично стає рутинним розрахунком з певною постійною похибкою. Прогнози можуть бути класифіковані (умовно) за такими критеріями:

- За термінами: короткострокові, середньострокові, довгострокові, далекострокові.
- За масштабом: приватні, місцеві, регіональні, галузеві, країнові, світові (глобальні).
- За відповідальністю (авторством): особисті, на рівні підприємства (організації), на рівні державних органів.

На сьогоднішній день широко застосовується комплексний метод прогнозування, відомий як форсайт (або майбутні студії), особливо в Європі. Форсайт є системою методів, що включають експертну оцінку стратегічних напрямків соціально-економічного та інноваційного розвитку, виявлення технологічних проривів, які можуть вплинути на економіку та суспільство у середньо- та довгостроковій перспективі. За допомогою цього методу створюються різні сценарії майбутнього світу, включаючи важливі аспекти та критичні ситуації, які виникають внаслідок різних варіантів розвитку і визначають заплановані дії.

Прогнозування входить до підготовчих етапів різного виду (економіко-соціального, науково-технічного, територіального тощо) та рівня (поточного, середньострокового, довгострокового, стратегічного) планування, при цьому форсайт застосовується при підготовці стратегічних планів (технологічного, наукового, науково -технічного та інших) розвитку країн, систем загальнодержавного рівня (охорони здоров'я, освіти тощо) та організацій (корпорацій, підприємств тощо).

Статистичні методи прогнозування є галуззю науки та навчання, що займається розробкою, вивченням і застосуванням сучасних математико-статистичних методів прогнозування на підставі об'єктивних даних. Основні завдання цієї дисципліни включають розвиток теорії та

практики імовірнісно-статистичного моделювання, експертних методів прогнозування, методів прогнозування в умовах ризику та комбінованих підходів до прогнозування, які використовують спільні економіко-математичні та економетричні (математико-статистичні та експертні) моделі. Ці методи ґрунтуються на прикладній статистиці та теорії прийняття рішень.

У контексті прогнозування, тимчасовий ряд часто аналізується з використанням ймовірнісних моделей, де враховуються додаткові фактори (незалежні змінні) окрім часу, наприклад, обсяг грошової маси. Тимчасовий ряд може бути багатовимірним. Основні завдання полягають у проведенні інтерполяції та екстраполяції. Метод найменших квадратів, запропонований К. Гауссом у 1794-1795 роках, є одним з найпростіших методів, особливо для лінійних функцій залежності від одного фактора. Іноді використовуються попередні перетворення змінних, наприклад, логарифмування. Часто застосовується метод найменших квадратів з кількома факторами, але також існують інші методи, такі як метод найменших модулів, сплайни, які рідше використовуються, але мають кращі статистичні властивості.

Оцінювання точності прогнозу, зокрема за допомогою довірчих інтервалів, є важливою частиною процедури прогнозування. Зазвичай використовують імовірнісно-статистичні моделі для відновлення залежностей, наприклад, будується найкращий прогноз за методом максимальної правдоподібності. Для оцінки точності прогнозу та встановлення довірчих інтервалів для нього (за основу береться Центральна Гранична Теорема теорії ймовірностей) розроблено параметричні методи (зазвичай на основі моделі нормальних помилок) та непараметричні методи. Крім того, існують евристичні прийоми, які не базуються на ймовірнісно-статистичній теорії, наприклад, метод ковзних середніх.

На сьогоднішній день, основним статистичним інструментом прогнозування є багатомірна регресія, включаючи використання непараметричних оцінок густини розподілу. Необов'язково припускати, що похибки вимірів та відхилення від лінії (поверхні) регресії мають нормальний розподіл; однак, для відмови від цього припущення потрібно використовувати математичний апарат, який базується на багатовимірній Центральній Граничній Теоремі теорії ймовірностей, технології лінеаризації та спадкуванні збіжності[4]. Він дозволяє проводити точкове та інтервальне оцінювання параметрів, перевіряти значущість їхньої відмінності від 0 у непараметричній постановці, будувати довірчі межі для прогнозу.

Дуже важливими є дві проблеми - перевірка адекватності моделі та відбір факторів. Зазвичай існує великий початковий список факторів, які впливають на відгук, але їх бажано скоротити, і сучасні дослідження активно присвячені методам відбору "інформативного набору ознак". Проте ця проблема поки що не має остаточного рішення. Також виявляються незвичайні ефекти. Наприклад, було встановлено, що оцінки ступеня полінома, які зазвичай використовуються, мають геометричний розподіл в асимптотиці. Перспективними є непараметричні методи оцінювання щільності ймовірності та їх застосування для відновлення регресійних залежностей будь-якого вигляду. Найбільш загальні результати в цій галузі були отримані за допомогою підходів статистики для нечислових даних.

До сучасних методів статистичного прогнозування належать різні підходи, включаючи експоненційне згладжування, авторегресію з ковзною середньою, економетричні системи, які базуються як на параметричних, так і на непараметричних методах.

Для застосування асимптотичних результатів при невеликому обсязі вибірки корисними є комп'ютерні статистичні технології. Вони також дозволяють створювати різні імітаційні моделі. Зокрема, методи розмноження даних, такі як бутстреп-методи, є корисними.

Системи прогнозування, які активно використовують комп'ютери, об'єднують різні методи прогнозування в одній автоматизованій робочій обстановці прогнозиста.

Прогнозування якісних ознак на основі даних, що не мають числової природи, базується на використанні статистичних методів для аналізу нечислових даних. Важливими напрямками для прогнозування є регресійний аналіз на основі інтервальних даних, який включає визначення та розрахунок надійного рівня і оптимального обсягу вибірки, а також регресійний аналіз для нечітких даних, що був розроблений у [5]. Загальна постановка регресійного аналізу в рамках статистики нечислових даних, яка включає дисперсійний аналіз та дискримінантний аналіз (розпізнавання образів з учителем), надає єдиний підхід до формально різних методів, що є корисним при реалізації сучасних статистичних методів прогнозування.

Основними кроками у обробці прогностичних експертних оцінок є перевірка узгодженості, кластерний аналіз та виявлення групової думки.

Перевірка узгодженості думок експертів, виражених у вигляді ранжувань, виконується за допомогою коефіцієнтів рангової кореляції Кендалла і Спірмена, а також коефіцієнта рангової конкордації Кендалла і Бебінгтона Сміта. Для цього використовуються як параметричні моделі парних порівнянь, такі як Терстоуна, Бредлі-Террі-Льюса, так і непараметричні моделі теорії люсіанів.

Іншою корисною процедурою для узгодження ранжувань і класифікацій є побудова бінарних відносин, які забезпечують узгодженість. У випадку відсутності узгодженості в розподілі думок експертів, групи, які мають схожі думки, об'єднуються за допомогою методу найближчого сусіда або інших методів кластерного аналізу (автоматичної побудови класифікацій, розпізнавання образів без вчителя).

Класифікація люсіанів здійснюється на основі імовірнісно-статистичної моделі.

Для отримання підсумкової думки комісії експертів застосовуються різноманітні методи. Серед них особливо простими є методи середніх арифметичних і медіан рангів. Завдяки комп'ютерному моделюванню [3] було виявлено кілька властивостей медіани Кемені, яка часто рекомендується в якості підсумкової (узагальненої, середньої) думки комісії експертів. Інтерпретація закону великих чисел для нечислових даних в контексті експертного опитування полягає в тому, що підсумкова думка є стійкою, тобто мало змінюється при зміні складу експертної комісії, і зі збільшенням кількості експертів наближається до "істини". Згідно з підходом, запропонованим у [4], вважається, що відповіді експертів можна розглядати як результати вимірювань з помилками, причому всі вони є незалежними однаково розподіленими випадковими величинами. Ймовірність прийняття певного значення зменшується з віддаленням від певного центру - "істини", а загальна кількість експертів є достатньо великою.

Існують багато прикладів ситуацій, що включають соціальні, технологічні, економічні, політичні, екологічні та інші ризики. Зазвичай в таких ситуаціях потрібне прогнозування. У теорії прийняття рішень [2] використовуються різні види критеріїв для ситуацій невизначеності

(ризику). Оскільки рішення, отримані за допомогою різних критеріїв, можуть бути протилежними, виникає необхідність використовувати оцінки експертів.

У конкретних прогнозувальних завданнях потрібно класифікувати ризики, встановлювати завдання оцінювання конкретного ризику, структурувати ризик, включаючи побудову причинних дерев (або дерева відмов) та дерева наслідків (дерево подій). Центральним завданням є створення групових та узагальнених показників, наприклад, показників конкурентоспроможності та якості.

При прогнозуванні економічних наслідків прийнятих рішень, споживацької поведінки та конкурентного оточення, зовнішньоекономічних умов та макроекономічного розвитку, екологічного стану довкілля, безпеки технологій та екологічних небезпек промислових та інших об'єктів необхідно враховувати ризики.

Сучасні прогнозувальні комп'ютерні технології базуються на використанні інтерактивних статистичних методів прогнозування, що включають економетричні дані, імітаційні (включаючи метод статистичних випробувань) та економіко-математичні динамічні моделі. Ці технології поєднують експертні, математико-статистичні та моделюючі компоненти.

Для прогнозування часових рядів зазвичай використовують комп'ютерні програми. Це дозволяє автоматизувати багато операцій при побудові прогнозів і уникнути помилок, пов'язаних з введенням даних та моделюванням. Такі програми можуть бути локальними (встановлені на комп'ютері користувача) або інтернет-додатками (доступними через веб-сайт). До прикладів локальних програм відносяться R, SPSS, Statistica, Forecast Pro, Forecast Expert.

Нейронна мережа має здатність до прогнозування, оскільки вона може узагальнювати і розпізнавати приховані залежності між вхідними і вихідними даними. Після процесу навчання мережа здатна передбачати майбутні значення послідовності на основі попередніх значень та/або існуючих факторів на даний момент. Прогнозування можливе тільки в тих випадках, коли попередні зміни в дійсності впливають на майбутні результати. Наприклад, прогнозування акційних котирувань на основі минулих тижневих котирувань може бути успішним (але не завжди), тоді як спроба прогнозування результатів лотереї на основі даних за останні 50 років майже напевно не дасть результатів.

2.2 Існуючі інструменти прогнозування

R

Мова програмування R є безкоштовним середовищем для статистичних обчислень і графіки, яке підтримується командою R і R Foundation for Statistical Computing[7]. Вона широко використовується статистиками та майнерами даних для розробки статистичного програмного забезпечення та аналізу даних. Дослідження та аналіз наукової літератури свідчать про значний ріст популярності мови R. За даними з серпня 2021 року, R займає 14-е місце в індексі TIOBE, що відображає популярність мов програмування.[9]

Офіційне програмне середовище R є пакетом GNU. Він написаний переважно на C, Fortran і R (тому він частково розміщений самостійно) і є вільно доступним під загальною суспільною ліцензією GNU. Для різних операційних систем надаються попередньо скомпільовані виконувані

файли. Він має інтерфейс командного рядка, але доступні кілька графічних інтерфейсів сторонніх виробників, таких як RStudio, інтегроване середовище розробки; і Jupyter, інтерфейс ноутбука.

R та його бібліотеки надають широкий спектр статистичних та графічних методів, які включають лінійне і нелінійне моделювання, класичні статистичні тести, просторовий і часовий аналіз, класифікацію, кластеризацію та багато інших. R легко розширюється за допомогою функцій та розширень, а його спільнота відома активним внеском у створення пакетів. Багато стандартних функцій R написані саме на мові R, що дозволяє користувачам легко розуміти алгоритмічні вибори, які були зроблені. Для інтенсивних обчислень код C, C++ і Fortran можна зв’язувати та викликати під час виконання. Досвідчені користувачі можуть писати код C, C++, Java, .NET або Python, щоб безпосередньо маніпулювати об'єктами R. R є дуже розширюваним завдяки використанню пакетів, поданих користувачами, для конкретних функцій і конкретних областей дослідження. Завдяки своїй спадщині S, R має потужніші засоби об'єктно-орієнтованого програмування, ніж більшість мов статистичних обчислень. Розширенню його також сприяють його правила лексичного визначення області.

Ще одна сильна сторона R — статична графіка; він може створювати графіки якості публікації, включаючи математичні символи. Динамічна та інтерактивна графіка доступна через додаткові пакети.

Statistica

Statistica — це пакет програмного забезпечення для передової аналітики, спочатку розроблений компанією StatSoft і в даний час підтримується TIBCO Software Inc.[1] Statistica забезпечує аналіз даних,

керування даними, статистику, інтелектуальний аналіз даних, машинне навчання, аналіз тексту та процедури візуалізації даних.

Statistica є набором аналітичних програмних продуктів і рішень, що спочатку були розроблені компанією StatSoft і придбані компанією Dell у березні 2014 року. Це програмне забезпечення включає набір даних для аналізу, керування даними, візуалізації даних та процедур аналізу даних. Також воно надає різноманітні методи прогнозного моделювання, кластеризації, класифікації та дослідження. Завдяки інтеграції з безкоштовним середовищем програмування R з відкритим кодом[3][4], Statistica має додаткові методи і можливості. Різні пакети аналітичних методів доступні в шести лінійках продуктів.[5]

Statistica включає аналітичні та дослідницькі графіки на додаток до стандартних 2- і 3-вимірних графіків. Дії кисті (інтерактивне маркування, маркування та виключення даних) дозволяють досліджувати викиди та досліджувати дані.

Робота програмного забезпечення зазвичай включає завантаження таблиці даних і застосування статистичних функцій зі спадних меню або (у версіях, починаючи з 9.0) із стрічки. У меню потім буде запропоновано вказати змінні та тип необхідного аналізу. Немає необхідності вводити командні рядки. Кожен аналіз може включати графічний або табличний вихід і зберігається в окремій робочій книзі.

РОЗДІЛ 3

МОДЕЛІ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ВАЛЮТНИХ РИНКІВ

3.1 Аналіз даних

Аналіз даних - це процес обстеження, очищення, перетворення та моделювання даних з метою виявлення цінної інформації, формулювання висновків та підтримки процесу прийняття рішень[1]. Ця дисципліна має різноманітні аспекти і підходи, які охоплюють різні методи, що носять різні назви, і вона застосовується у різних галузях, таких як бізнес, наука та соціальні науки[2]. У сучасному корпоративному середовищі аналіз даних відіграє важливу роль у науковому підтвердженні рішень та допомагає підприємствам працювати більш ефективно. Цей процес заснований на обробці, вивченні та використанні великих обсягів даних для отримання нових інсайтів та розуміння реальних явищ та тенденцій.[3]

Інтелектуальний аналіз даних - це особлива методика аналізу, яка зосереджена на використанні статистичного моделювання та виявленні знань для прогнозування майбутніх подій, а не просто для описувальних цілей. У той час як бізнес-аналітика охоплює ширший спектр даних та в основному зосереджується на агрегації даних, з упором на бізнес-інформацію[4].

В рамках статистичного аналізу даних можна виділити три підходи: описову статистику, пошуковий аналіз даних (EDA) та підтверджуючий

аналіз даних (CDA)[5]. EDA спрямований на виявлення нових особливостей у даних, тоді як CDA фокусується на перевірці чи спростуванні існуючих гіпотез[7]. Прогностична аналітика ставить своєю метою застосування статистичних моделей для прогнозування або класифікації подій, тоді як текстова аналітика використовує статистичні, лінгвістичні та структурні методи для екстракції та класифікації інформації з текстових джерел та інших неструктурованих даних. Всі ці підходи є складовими аналізу даних.[8]

Процес інтеграції даних передує аналізу даних, і сам аналіз даних тісно пов'язаний з візуалізацією та поширенням результатів аналізу даних для їх подальшого використання та розуміння.[9]

Процес аналізу даних

Аналіз означає розбиття цілого на складові частини з метою окремого вивчення[10]. Аналіз даних - це процес обробки сирових даних з метою отримання корисної інформації, яка може бути використана для прийняття рішень користувачами[1]. Збирання та аналіз даних проводиться з метою відповіді на питання, перевірки гіпотез або спростування теорій[11].

Можна виділити кілька фаз, описаних нижче. Фази повторюються, тому зворотний зв'язок з наступних фаз може призвести до додаткової роботи на попередніх фазах.[13] Фреймворк CRISP, який використовується для інтелектуального аналізу даних, має подібні кроки.

Вимоги до даних

Аналітичні дані є вхідними даними, які використовуються для аналізу і вони залежать від вимог, встановлених аналітичними управліннями або клієнтами, які планують використовувати результати

аналізу[14][15]. Загальний об'єкт, щодо якого збираються дані, називається експериментальною одиницею, наприклад, людина або група людей. Можуть бути визначені і отримані конкретні змінні, що стосуються цільової аудиторії, такі як вік і дохід. Дані можуть бути представлені числами або категоріями (текстовими мітками для числових значень)[13].

Збір даних

Інформація збирається з різних джерел, які доступні для дослідження та аналізу[16][17]. Аналітики можуть надати вимоги зберігачам даних, наприклад, співробітникам відділу інформаційних технологій в організації[18]. Дані також можуть бути отримані з датчиків, розташованих у навколишньому середовищі, таких як дорожні камери, супутники, пристрої запису тощо. Інформацію також можна отримати шляхом проведення інтерв'ю, завантаження з онлайн-джерел або прочитання документації[13].

Обробка даних

Перед тим як проводити аналіз, отримані дані потрібно піддати обробці та організації.[19][20] Наприклад, це може включати структурування даних у вигляді таблиці з рядками і стовпцями (так звані структуровані дані) для подальшого аналізу, часто з використанням електронних таблиць або статистичного програмного забезпечення.[13]

Очищення даних

Після обробки та організації дані можуть бути неповними, містити дублікати або помилки.[21][22] Потреба у очищенні даних виникає через проблеми з введенням і збереженням даних.[21] Очищення даних - це процес виявлення та виправлення цих помилок. Загальні завдання включають порівняння записів, виявлення неточностей даних, оцінку

загальної якості даних, видалення дублікатів та групування стовпців.[23] Проблеми з даними також можна виявити за допомогою різних аналітичних методів. Наприклад, у випадку фінансової інформації можна порівняти загальні показники для окремих змінних з опублікованими надійними цифрами.[24][25] Також можуть бути перевірені незвичайні суми, які перевищують або знижуються нижче попередньо встановлених порогових значень. Існує кілька типів очищення даних, які залежать від типу даних у наборі, таких як номери телефонів, електронні адреси, роботодавці та інші значення[26][27]. Методи кількісних даних для виявлення викидів можуть бути використані, щоб позбутися даних, які, здається, мають більшу ймовірність бути неправильно введеними.[28] Щоб зменшити кількість слів з помилками, можна використовувати засоби перевірки орфографії текстових даних. Однак важче сказати, чи правильні самі слова.[29]

Пошуковий аналіз даних:

Після очищення набору даних, їх можна піддати аналізу. Аналітики використовують різноманітні методи, що відомі як дослідницький аналіз даних, для отримання розуміння повідомлень, що містяться в цих даних.[30] Процес дослідження даних може призвести до додаткового очищення даних або створення додаткових запитів до даних, що відображає початок ітераційного циклу, згаданого у першому параграфі цього розділу.[31] Для полегшення розуміння даних можна провести описову статистику, таку як середнє значення або медіана.[32][33] Також використовується візуалізація даних, за допомогою якої аналітики можуть вивчати дані у графічному форматі, отримуючи додаткову інформацію про повідомлення, що містяться в даних.[13]

Моделювання та алгоритми:

Математичні формули або моделі (також відомі як алгоритми) можуть бути застосовані до даних з метою виявлення зв'язків між змінними, наприклад, шляхом використання кореляції або причинно-наслідкового зв'язку.[34][35] Загалом, моделі можуть бути розроблені для оцінки конкретної змінної на основі інших змінних, що містяться в наборі даних, з деякою залишковою помилкою, залежно від точності моделі, що використовується (наприклад, Дані = Модель + Помилка)[36][11].

Інференційна статистика включає в себе використання методів, які вимірюють зв’язки між певними змінними.[37] Наприклад, регресійний аналіз може бути використаний для моделювання того, чи пояснює зміна в рекламі (незалежна змінна X) зміну продажів (залежна змінна Y).[38] З математичної точки зору, Y (продажі) є функцією X (реклама).[39] Його можна описати як (Y = aX + b + помилка), де модель розроблена таким чином, що (a) і (b) мінімізують помилку, коли модель передбачає Y для даного діапазону значень X.[40] Аналітики також можуть спробувати побудувати моделі, які описують дані, з метою спрощення аналізу та передачі результатів.[11]

Продукт даних

Продукт даних — це комп’ютерна програма, яка приймає вхідні дані та генерує вихідні дані, повертаючи їх у середовище.[41] Він може базуватися на моделі або алгоритмі. Наприклад, програма, яка аналізує дані про історію покупок клієнтів і використовує результати, щоб рекомендувати інші покупки, які можуть сподобатися клієнтам.[42][13]

Після того, як дані проаналізовано, вони можуть бути передані в багатьох форматах користувачам аналізу для підтримки їхніх вимог.[44]

Користувачі можуть мати відгуки, що призводить до додаткового аналізу. Таким чином, більша частина аналітичного циклу є ітеративною.[13]

Визначаючи, як повідомити результати, аналітик може використовувати різні методи візуалізації даних, щоб допомогти донести повідомлення до аудиторії більш зрозуміло та ефективно.[45] Візуалізація даних використовує інформаційні експонати (таблиці та графіки), щоб допомогти донести ключові повідомлення даних.[46] Таблиці є цінним інструментом, оскільки вони дають змогу користувачеві зробити запит і сконцентруватися на конкретних цифрах, тоді як графіки (наприклад, гістограми або лінійні діаграми) допомагають прояснити повідомлення.

Кількісні комунікації

Стівен Мало окреслив вісім категорій кількісних повідомлень, які користувачі можуть спробувати зрозуміти або передати на основі набору даних і графіків, що використовуються для передачі повідомлення.[48] Протягом усього процесу клієнти, які визначають вимоги, та аналітики, які виконують аналіз даних, можуть обмірковувати ці повідомлення.[49]

- Часовий ряд: одна змінна фіксується за певний період часу, наприклад, рівень безробіття за 10-річний період. Для демонстрації тенденції можна використовувати лінійну діаграму.[50]

- Рейтинг: Категориальні підрозділи ранжуються в порядку зростання або спадання, наприклад, рейтинг ефективності продажів (вимірювання) за продавцями (категорія, з кожним

продавцем категорійний підрозділ) протягом одного періоду.[51] Гістограму можна використовувати для порівняння між продавцями.[52]

- Частково до цілого: категоріальні підрозділи вимірюються як співвідношення до цілого (тобто у відсотках від 100%). Кругова діаграма або стовпчаста діаграма може показати порівняння співвідношень, таких як частка ринку, представлена конкурентами на ринку.[53]

- Відхилення: категоріальні підрозділи порівнюються з еталонними, наприклад порівняння фактичних і бюджетних витрат для кількох відділів підприємства за певний період часу. Гістограма може показати порівняння фактичної та контрольної суми.[54]

- Частотний розподіл: показує кількість спостережень певної змінної для заданого інтервалу, наприклад, кількість років, протягом яких прибутковість фондового ринку знаходиться між такими інтервалами, як 0–10%, 11–20% тощо. Гістограма, а типу гістограми, можна використовувати для цього аналізу.[55]

- Кореляція: порівняння між спостереженнями, представленими двома змінними (X, Y), щоб визначити, чи вони мають тенденцію рухатися в однакових чи протилежних напрямках. Наприклад, побудова графіка безробіття (X) та інфляції (Y) для вибірки місяців. Для цього повідомлення зазвичай використовується діаграма розсіювання.[56]

- Номінальне порівняння: порівняння категорійних підрозділів без певного порядку, наприклад, обсяг продажів за кодом продукту. Для цього порівняння можна використовувати гістограму.[57]

- Географічне або геопросторове: порівняння змінної на карті чи макеті, як-от рівень безробіття за штатом або кількість людей на різних поверхах будівлі. Картограма - це типова графіка, що використовується.[58][59]

Методи аналізу кількісних даних

Автор Джонатан Кумі рекомендував низку найкращих практик для розуміння кількісних даних.[60] До них належать:

- Перевірте необроблені дані на наявність аномалій перед виконанням аналізу;

- Повторне виконання важливих обчислень, таких як перевірка стовпців даних, які керуються формулою;

- Підтвердьте, що основні підсумки є сумою проміжних підсумків;

- Перевірте зв’язки між числами, які мають бути пов’язані передбачуваним чином, наприклад співвідношення в часі;

- Нормалізуйте числа, щоб полегшити порівняння, наприклад, аналізуючи суми на людину або відносно ВВП, або як

значення індексу відносно базового року;

- Розбийте проблеми на складові частини, проаналізувавши фактори, які призвели до результатів, наприклад аналіз рентабельності капіталу DuPont.[25]

Аналітики зазвичай отримують описову статистику, таку як середнє значення, медіана та стандартне відхилення для досліджуваних змінних. Крім того, вони можуть проаналізувати розподіл важливих змінних, щоб визначити, як значення агрегуються навколо середнього значення[62].

McKinsey & Company назвала метод декомпозиції кількісної проблеми на складові елементи принципом MECE.[63] Кожен рівень може бути поділений на підкомпоненти, кожен з яких повинен бути взаємовиключним по відношенню до підкомпонентів, що складають рівень вище нього.[64] Відношення відоме як "взаємовиключний і колективно вичерпний" (MECE). Наприклад, прибуток за визначенням можна розділити на загальний дохід і загальні витрати.[65] Загальний дохід можна розбити на складові частини, такі як дохід від підрозділів A, B і C (які є взаємовиключними), і вони повинні бути додані до загального доходу (взаємовиключні)[66].

Певні аналітичні проблеми можуть бути вирішені аналітиками за допомогою надійних статистичних вимірювань.[67] Коли аналітик висуває певну гіпотезу про справжній стан речей і накопичує дані, щоб визначити, чи є цей стан речей істинним чи хибним, проводиться перевірка гіпотези. [Наприклад, гіпотезою може бути "безробіття не впливає на інфляцію", що стосується кривої Філліпса, економічної концепції.[70] Під час перевірки гіпотези необхідно враховувати ймовірність помилок першого та другого

типу, які стосуються того, чи підтверджують дані прийняття або відкидання гіпотези, чи ні.

Аналітик може використовувати регресійний аналіз, намагаючись визначити ступінь впливу незалежної змінної X на залежну змінну Y (наприклад, "Якою мірою зміни в рівні безробіття (X) впливають на рівень інфляції (Y)?"). [73] Це спроба змоделювати або підігнати криву чи лінію до даних таким чином, щоб Y була функцією від X.[74][75].

Аналіз необхідних умов (NCA) використовується при визначенні ступеня залежності незалежної змінної X від залежної змінної Y (наприклад, "Якою мірою певний рівень безробіття (X) необхідний для певного рівня інфляції (Y)?"). [В той час як (множинний) регресійний аналіз використовує адитивну логіку, в якій кожна змінна X може спричинити результат, а X можуть компенсувати одна одну (вони є достатніми, але не обов'язковими),[76] аналіз необхідних умов (АНУ) використовує логіку необхідності, в якій одна або більше змінних X дозволяють результату існувати, але не обов'язково спричиняють його (вони є необхідними, але не достатніми). Кожна окрема умова має бути виконана, і немає ніякої можливості компенсації"[77].

Перешкоди для ефективного аналізу

Перешкоди для ефективного аналізу можуть існувати серед аналітиків, які виконують аналіз даних, або серед аудиторії. Відрізнити факти від думок, когнітивні упередження та неврахування рахунків – усе це складнощі для надійного аналізу даних.[82]

Плутання фактів і думок

Ефективний аналіз вимагає збору відповідних даних для того, щоб відповісти на питання, підтримати висновок чи офіційну думку або

перевірити гіпотези.[83][84] За визначенням, факти є незаперечними, тому всі учасники аналізу повинні бути здатні їх прийняти.[85] Наприклад, Бюджетне управління Конгресу США (CBO) підрахувало в серпні 2010 року, що продовження зниження податків Буша в 2001 і 2003 роках на період 2011-2020 років збільшить державний борг приблизно на 3,3 трильйона доларів[86]. Всі повинні погодитися, що це саме те, про що повідомляє CBO; всі можуть перевірити цю інформацію. Отже, це реальність. "Погоджуються чи не погоджуються люди з висновками РПЦ - це питання індивідуальної думки"[87].

Аудитор публічної компанії також повинен отримати офіційний висновок про те, чи є фінансова звітність "достовірною в усіх суттєвих аспектах"[88]. Це вимагає ретельного вивчення фактів і доказів, що підтверджують їхню позицію. При переході від фактів до висновків завжди існує ймовірність того, що висновок буде неправильним"[89].

Когнітивні упередження

Існують різноманітні когнітивні упередження, які можуть негативно вплинути на аналіз. Наприклад, упередження підтвердження – це схильність шукати або інтерпретувати інформацію таким чином, щоб підтверджувати власні упередження.[90] Крім того, особи можуть дискредитувати інформацію, яка не підтверджує їхні погляди.[91]

Аналітики можуть пройти спеціальну підготовку для усвідомлення цих упереджень і того, як їх подолати.[92] У своїй книзі «Психологія аналізу розвідки» аналітик ЦРУ у відставці Річардс Хойер писав, що аналітики повинні чітко окреслити свої припущення та ланцюжки висновків, а також визначити ступінь і джерело невизначеності, пов’язаної з висновками.[93] Він наголосив на процедурах, які допомагають висвітлювати та обговорювати альтернативні точки зору.[94]

Ненумерація

Типовими є аналітики, які володіють різноманітними числовими методами. З іншого боку, аудиторія може бути не настільки чисельною; її називають нечисельною.[95] Ті, хто передає дані, можуть також намагатися ввести в оману або дезінформувати, навмисно використовуючи неякісні числові методи.[96] Наприклад, може бути неважливо, зростає число чи зменшується.

Наприклад, те, зростає чи зменшується число, може не бути визначальним фактором. Більш важливим може бути співвідношення між двома числами, наприклад, розмір державних доходів чи видатків відносно розміру економіки (ВВП) або кількість витрат відносно доходів у корпоративній фінансовій звітності.[97] Цей числовий метод відомий як нормалізація [25] або стандартне масштабування. Незалежно від того, чи це поправка на інфляцію (тобто порівняння реальних і номінальних даних), чи врахування зростання населення, демографічних показників тощо, аналітики застосовують різноманітні методи[98]. Аналітики використовують різноманітні методи для вирішення різних кількісних проблем, описаних у попередньому розділі.

Аналітики також можуть аналізувати дані на основі різних гіпотез або сценаріїв. Наприклад, коли аналітики проводять аналіз фінансової звітності, вони часто перераховують фінансові звіти відповідно до різних припущень, щоб допомогти отримати оцінку майбутніх грошових потоків, які вони потім дисконтують до теперішньої вартості на основі певної відсоткової ставки, щоб визначити оцінку вартості компанії або акцій.[100][101] Аналогічно, ОУП аналізує вплив альтернативних варіантів політики на державні доходи, витрати і дефіцит, розробляючи альтернативні сценарії майбутнього для ключових показників.

3.2 Нейронні мережі

Штучні нейронні мережі (ANN), які зазвичай просто називають нейронними мережами (NN), - це обчислювальні системи, нечітко натхненні біологічними нейронними мережами, які складають мозок тварин. [1]

ANN складається з мережі взаємопов'язаних блоків або вузлів, відомих як штучні нейрони, які дуже схожі на нейрони біологічного мозку. Подібно до синапсів у біологічному мозку, кожне з'єднання може передавати сигнал іншим нейронам. Штучний нейрон, який отримує сигнал, потім обробляє його і може спілкуватися з нейронами, які до нього під'єднані. "Сигнал" у з'єднанні є дійсним числом, а вихід кожного нейрона визначається нелінійною функцією, що включає суму його входів. Термін для з'єднань - ребро. Під час навчання вага нейронів та ребер зазвичай змінюється. Вага або збільшує, або зменшує амплітуду сигналу на з'єднанні. Нейрони можуть мати поріг чутливості, так що сигнал передається лише тоді, коли сумарний сигнал перевищує його. Нейрони зазвичай розташовані шарами. На своїх входах різні шари можуть виконувати різні перетворення. Сигнали проходять від першого шару (вхідного) до останнього (вихідного), ймовірно, після багаторазового проходження шарів.

Уоррен Маккалох і Волтер Піттс [2] (1943) започаткували цю тему, розробивши обчислювальну модель нейронної мережі. [3] Наприкінці 1940-х років Д. О. Хебб [4] розробив гіпотезу навчання Хебба, яка ґрунтується на механізмі нейронної пластичності. Фарлі та Веслі А. Кларк

[5] (1954) спочатку використовували комп'ютери, згодом названі "калькуляторами", для моделювання мережі Гебба. Розенблат [6] розробив персептрон у 1958 році. [7] Івахненко та Лапа опублікували перші функціональні мережі з декількома шарами в 1965 році як метод групової обробки даних. [8] [9] [10] Келлі [14] та Брайсон [15] вивели основи неперервного зворотного поширення [8] [11] [12] [13] в контексті теорії управління в 1960 та 1961 роках відповідно, використовуючи принципи динамічного програмування.

У 1970р. Сеппо Ліннайнмаа опублікував метод автоматичного диференціювання (АД) дискретних зв'язаних мереж вкладених диференційовних функцій [16] [17]. Дрейфус використав зворотне поширення в 1973 році для регулювання параметрів регулятора пропорційно градієнтам помилки. Алгоритм зворотного розповсюдження, розроблений Вербосом у 1975 році, уможливив практичне навчання багаторівневих мереж. У 1982 році він адаптував метод АД Ліннайнмаа до нейронних мереж у спосіб, який отримав широке визнання [11] [19]. Після того, як Мінський і Паперт (1969) [20] виявили, що фундаментальні персептрони не здатні обробляти ексклюзивність або схеми, і що комп'ютери не мають можливості обробляти корисні нейронні мережі, дослідження зайшли в глухий кут.

У цифровій електроніці розвиток дуже широкомасштабної інтеграції метал-оксид-напівпровідник (МОН) у вигляді технології комплементарного метал-оксид-напівпровідник (КМОН) уможливив збільшення кількості МОН-транзисторів. У 1980-х роках це збільшило обчислювальну потужність для розробки практичних штучних нейронних мереж [21].

У 1992 році було запроваджено максимальне об'єднання, щоб допомогти з мінімальним зміщенням інваріантності та толерантності до деформації для полегшення розпізнавання 3D-об'єктів. [22] [23] [24] Шмідхубер (1992) прийняв багаторівневу ієрархію мережі, яка була попередньо навчена на кожному рівні за допомогою неконтрольованого навчання і вдосконалена шляхом зворотного поширення.

Джеффрі Хінтон та ін. (2006) запропонували навчання високорівневих репрезентацій з використанням послідовних шарів двійкових або реальних латентних змінних, змодельованих обмеженою машиною Больцмана [26]. У 2012 році Нг і Дінг створили мережу, яка навчилася розпізнавати високорівневі концепти, такі як коти, переглядаючи лише немарковані зображення [27]. Неконтрольоване попереднє навчання і збільшення обчислювальної потужності завдяки графічним процесорам і розподіленим обчисленням дозволили використовувати більші мережі, особливо для задач розпізнавання зображень і зору - явище, яке отримало назву "глибоке навчання". [28]

Ciresan з колегами (2010) [29] продемонстрували, що графічні процесори забезпечують зворотне поширення для багатошарових нейронних мереж прямого поширення, незважаючи на проблему зникнення градієнта. [30] ШНМ почали завойовувати нагороди на змаганнях ШНМ між 2009 та 2012 роками, наближаючись до людського рівня у виконанні різноманітних завдань, починаючи з розпізнавання образів та машинного навчання. [31] [32] Наприклад, двонаправлена і багатовимірна довга короткочасна пам'ять (LSTM) [33] [34] [35] [36] Грейвса та ін. виграла три конкурси з розпізнавання рукописного тексту в 2009 році без будь-яких попередніх знань трьох мов, які потрібно було вивчити [35] [34].

Ciresan and colleagues створили перші розпізнавачі шаблонів для досягнення конкурентоспроможності людини / надлюдської діяльності [37] на таких орієнтирах, як розпізнавання дорожніх знаків (IJCNN 2012).

Навчання нейронних мереж передбачає обробку прикладів, кожен з яких містить відомий "вхід" і "вихід", і зберігання зважених за ймовірністю асоціацій між ними в структурі даних мережі. Зазвичай при навчанні нейромережі на заданому прикладі визначають різницю між обробленим виходом мережі (часто передбаченням) і бажаним результатом. Це і є помилка. Використовуючи це значення помилки, мережа потім модифікує свої зважені асоціації відповідно до правила навчання. Послідовні модифікації призведуть до того, що вихід нейромережі стане все більш схожим на бажаний результат. Після достатньої кількості таких коригувань, на основі певних критеріїв, навчання можна припинити. Це називається керованим навчанням.

Такі системи "вчаться" виконувати завдання, спостерігаючи за прикладами, як правило, без програмування принципів, що відповідають конкретному завданню. Наприклад, у розпізнаванні зображень вони можуть навчитися розпізнавати зображення, що містять котів, аналізуючи приклади зображень, які були вручну позначені як "кіт" або "без кота", і застосовуючи результати для ідентифікації котів на інших зображеннях. Вони роблять це, не маючи жодних попередніх знань про котів, наприклад, про те, що вони мають хутро, хвости, вуса та котячі риси. Натомість вони генерують ідентифікаційні характеристики автономно на основі прикладів, які обробляють.

Існує дві парадигми навчання нейронних мереж - з учителем і без вчителя. У першому випадку, на вхідний вектор є готова відповідь, у другому випадку нейронна мережа самонавчається. У кожного виду

навчання є своя ніша завдань і за великим рахунком вони не перетинаються. На даний момент придумано і запатентовано велика кількість архітектур нейронних мереж і методів їх навчання. Але основними (вихідними) є - для навчання з учителем це «алгоритм зворотного поширення помилки», а для навчання без учителя це алгоритми Хебба і Кохонена. Ці парадигми сильно перетинаються з біологічної дійсністю, наприклад - дитина навчається з учителем або без? Також, останнім часом, сформувалася нова, третя парадигма - навчання з підкріпленням.

Навчання з учителем

Завданням машинного навчання (навчання з учителем) є вивчення функції, яка перетворює вхідні дані у вихідні на основі прикладів пар вхід-вихід.[1] Він виводить функцію з набору навчальних прикладів і маркованих навчальних даних.[2] Кожен приклад у навчанні з учителем складається з вхідного об'єкта (як правило, вектора) і бажаного вихідного значення (також відомого як керуючий сигнал). Алгоритм керованого навчання аналізує навчальні дані і генерує вихідну функцію, яка може бути використана для відображення нових прикладів. Оптимальний сценарій дозволить алгоритму визначити позначення класів для невидимих прикладів. Це вимагає від алгоритму навчання "розумного" узагальнення навчальних даних до неспостережуваних ситуацій (див. індуктивне зміщення). Так звана помилка узагальнення кількісно вимірює цю статистичну характеристику алгоритму.

Паралельне завдання в психології людини і тварин часто називають поняттям навчання.

Щоб вирішити задану проблему навчання з керівництвом, необхідно виконати наступні кроки:

- Визначте тип навчальних прикладів. Перш ніж робити що-небудь інше, користувач повинен вирішити, які дані будуть використовуватися як навчальний набір.

- У випадку аналізу почерку, наприклад, це може бути один рукописний символ, ціле рукописне слово, ціле речення почерку або, можливо, повний абзац рукописного введення.

- Зберіть тренувальний набір. Навчальний набір має бути репрезентативним для реального використання функції. Таким чином, збирається набір вхідних об’єктів, а також збираються відповідні вихідні дані або від експертів-людей, або від вимірювань.

- Визначте представлення вхідних ознак вивченої функції. Точність вивченої функції сильно залежить від того, як представлений вхідний об’єкт. Як правило, вхідний об’єкт перетворюється на вектор ознак, який містить ряд ознак, що є описовими для об’єкта. Кількість ознак не повинна бути занадто великою через прокляття розмірності; але має містити достатньо інформації для точного прогнозування результату.

- Визначте структуру вивченої функції та відповідний алгоритм навчання. Наприклад, інженер може вибрати використання машин опорних векторів або дерев рішень.

- Завершіть дизайн. Запустіть алгоритм навчання на зібраному навчальному наборі. Деякі алгоритми навчання під керуванням вимагають від користувача визначення певних

параметрів керування. Ці параметри можна налаштувати шляхом оптимізації продуктивності підмножини (так званої набором перевірки) навчального набору або за допомогою перехресної перевірки.

- Оцініть точність вивченої функції. Після налаштування параметрів і навчання продуктивність отриманої функції слід виміряти на тестовому наборі, який є окремо від навчального набору.

Доступний широкий спектр алгоритмів навчання з наглядом, кожен з яких має свої сильні та слабкі сторони. Не існує єдиного алгоритму навчання, який найкраще працює з усіма проблемами навчання під керівництвом.

Є чотири основні проблеми, які слід враховувати під час навчання з наглядом:

Компроміс зміщення-дисперсія

Першою проблемою є компроміс між упередженістю та дисперсією[3]. Уявіть, що ми маємо доступ до кількох різних, але однаково цінних навчальних наборів даних. Алгоритм навчання є упередженим для певного вхідного стилю відображення xx, якщо при навчанні на кожному з цих наборів даних він неправильно прогнозує правильний результат для стилю відображення xx. Алгоритм навчання має високу дисперсію для заданого вхідного стилю відображення xx, якщо він прогнозує різні вихідні значення при навчанні на різних навчальних наборах. Помилка передбачення навченого класифікатора пропорційна сумі зсуву та дисперсії алгоритму навчання[4]. Зазвичай існує компроміс між зсувом та

дисперсією. Алгоритм навчання з низьким зміщенням повинен бути "гнучким", щоб він міг ефективно підлаштовуватися під дані. Однак, якщо алгоритм навчання є надмірно адаптивним, він буде по-різному підходити до кожного набору навчальних даних, що призведе до високої дисперсії. Ключовою особливістю багатьох методів навчання з навчанням під наглядом є те, що вони можуть змінювати компроміс між зміщенням і дисперсією (або автоматично, або за допомогою параметра зміщення/дисперсії, що налаштовується).

Складність функції та обсяг навчальних даних

Друге питання - це кількість доступних навчальних даних у порівнянні зі складністю "істинної" функції (класифікатора або регресійної функції). Якщо істинна функція є базовою, то "негнучкий" алгоритм навчання з високим зміщенням і низькою дисперсією може навчити її на невеликій кількості даних. Використовуючи "гнучкий" алгоритм навчання з низьким зсувом і високою дисперсією, функція зможе навчитися на дуже великій кількості навчальних даних, тільки якщо істинна функція є дуже складною (наприклад, через те, що вона включає складні взаємодії між багатьма різними вхідними функціями і поводиться по-різному в різних частинах вхідного простору). Існує чіткий поділ між вхідними даними та бажаним результатом.

Розмірність вхідного простору

Третьою проблемою є розмірність вхідного простору. Навіть якщо істинна функція залежить від обмеженої кількості вхідних векторів ознак, проблема навчання може бути складною, якщо розмірність вхідних векторів ознак надзвичайно велика. Це пов'язано з тим, що велика кількість "зайвих" вимірів може заплутати алгоритм навчання і призвести до його високої дисперсії. Отже, висока вхідна розмірність, як правило,

вимагає налаштування класифікатора на низьку дисперсію та високу похибку. Якщо інженер може вручну видалити нерелевантні ознаки з вхідних даних, точність навченої ознаки, ймовірно, покращиться. Крім того, існує безліч алгоритмів відбору ознак, які намагаються визначити релевантні ознаки та усунути нерелевантні. Перед виконанням алгоритму керованого навчання наведемо приклад більш загальної стратегії зменшення розмірності, яка має на меті відобразити вхідні дані у простір зі зменшеною розмірністю.

Шум у вихідних значеннях

Четвертим питанням є ступінь збурення цільових вихідних значень (цільових змінних керування). Якщо заплановані вихідні значення часто неточні (через помилки людини або датчиків), алгоритм навчання не повинен намагатися знайти функцію, яка точно відповідає навчальним прикладам. Спроба занадто точно підігнати дані призводить до надмірного припасування. Навіть якщо немає помилок вимірювання (випадкового шуму), ви можете отримати перенавчання, якщо функція, яку ви намагаєтеся вивчити, занадто складна для вашої навчальної моделі. У цьому випадку частина цільової функції, яку неможливо змоделювати, "псує" ваші навчальні дані; це явище відоме як детермінований шум. Якщо присутня будь-яка форма збурення, краще використовувати оцінки з більшими зміщеннями і меншими дисперсіями.

На практиці існує кілька підходів до зменшення шуму у вихідних значеннях, таких як рання зупинка, щоб запобігти переобладнанню, а також виявлення та видалення галасливих прикладів навчання перед навчанням алгоритму навчання під керівництвом. Існує кілька алгоритмів, які ідентифікують зашумлені навчальні приклади, і видалення

передбачуваних прикладів зашумливого навчання перед навчанням зменшило помилку узагальнення зі статистичною значущістю.[5][6].

Інші фактори, які слід враховувати

Інші фактори, які слід враховувати при виборі та застосуванні алгоритму навчання, включають наступне:

- Неоднорідність даних. Коли вектори ознак містять широкий спектр типів ознак (дискретні, дискретно впорядковані, відліки, неперервні значення), деякі алгоритми простіше реалізувати, ніж інші. Численні алгоритми, такі як машини опорних векторів, лінійна регресія, логістична регресія, нейронні мережі та методи найближчого сусіда, вимагають, щоб вхідні ознаки були числовими і масштабованими до подібних діапазонів (наприклад, [-1,1]). Це особливо чутливо для методів, які використовують функцію відстані, таких як метод найближчого сусіда та машини опорних векторів з ядрами Гауса. Перевага дерев рішень полягає в тому, що вони можуть легко керувати розрізненими наборами даних.

- Надмірність в даних. Через числову нестабільність певні алгоритми навчання (наприклад, лінійна регресія, логістична регресія та методи на основі відстані) будуть працювати неадекватно, якщо вхідні об'єкти містять надлишкову інформацію (наприклад, ознаки з високою кореляцією). Ці проблеми часто вирішуються шляхом застосування певної форми регуляризації.

- Наявність взаємодій і нелінійності. Якщо кожна функція робить незалежний внесок у вихідні дані, лінійні функції (наприклад, лінійна регресія, логістична регресія, машини опорних

векторів, наївні байєсівські машини) і функції відстані (методи найближчого сусіда, машини опорних векторів з ядрами Гауса) зазвичай є ефективними. Однак, якщо між функціями існують складні взаємодії, дерева рішень і нейронні мережі працюють краще, оскільки вони призначені для виявлення цих взаємодій. Також застосовуються лінійні методи, для яких інженер повинен явно вказати взаємодії.

Оцінюючи нову програму, інженер може порівняти кілька алгоритмів навчання та експериментально визначити, який з них є найефективнішим для вирішення проблеми (див. перехресна перевірка). Налаштування ефективності алгоритму навчання може бути складним процесом. З огляду на фіксовані ресурси, часто краще витратити більше часу на збір додаткових навчальних даних і більш інформативних функцій, ніж на вдосконалення алгоритмів навчання.

Алгоритми

Найбільш широко використовувані алгоритми навчання:

- Опорно-векторні машини
- Лінійна регресія
- Логістична регресія
- Наївний Байєс
- Лінійний дискримінантний аналіз

- Дерева рішень
- Алгоритм K-найближчого сусіда
- Нейронні мережі (багатошаровий персептрон)
- Навчання подібності

Підходи та алгоритми

- Аналітичне навчання
- Штучна нейронна мережа
- Зворотне поширення
- Підвищення (мета-алгоритм)
- Байєсівська статистика
- Міркування на основі випадків
- Навчання дерева рішень
- Індуктивне логічне програмування
- Регресія процесу Гаусса

- Генетичне програмування
- Груповий метод обробки даних
- Оцінки ядра
- Навчальні автомати
- Навчання систем класифікації
- Мінімальна довжина повідомлення (дерева рішень, графіки рішень тощо)
- Мультилінійне підпросторове навчання
- Наївний байєсівський класифікатор
- Класифікатор максимальної ентропії
- Умовне випадкове поле
- Алгоритм найближчого сусіда
- Ймовірно, приблизно правильне навчання (PAC).
- Правила Ripple down, методологія отримання знань
- Алгоритми символічного машинного навчання

- Підсимволічні алгоритми машинного навчання
- Опорно-векторні машини
- Машини мінімальної складності (MCM)
- Випадкові ліси
- Ансамблі класифікаторів
- Порядкова класифікація
- Попередня обробка даних
- Обробка незбалансованих наборів даних
- Статистичне реляційне навчання
- Proaftn, багатокритеріальний алгоритм класифікації

Навчання без учителя

Навчання без вчителя - це форма машинного навчання, в якій навчальні дані не маркуються і не оцінюються заздалегідь. Тому алгоритми неконтрольованого навчання повинні спочатку самостійно ідентифікувати будь-які закономірності, що з'являються в цьому наборі навчальних даних. Поширеними прикладами є кластеризація, коли алгоритм автоматично групує навчальні приклади в категорії зі схожими ознаками, і аналіз головних компонент, коли алгоритм знаходить способи стиснути

навчальний набір, визначаючи, які ознаки є найбільш корисними для розрізнення різних навчальних прикладів і відкидаючи решту. Це на відміну від навчання під контролем, де навчальні дані містять попередньо призначені мітки категорій (зазвичай призначені людиною або результати алгоритму класифікації без контролю).[3] Навчання з підкріпленням, де для кожного навчального прикладу доступні лише числові оцінки замість детальних міток, і напівконтрольоване навчання, де лише частина навчальних даних має мітки, є додатковими проміжними рівнями на континуумі контролю.

Переваги неконтрольованого навчання включають мінімальне навантаження на підготовку та перевірку навчальних наборів, на відміну від методів навчання з контролем, які вимагають значної експертної людської праці для призначення та перевірки початкових міток, а також більшу свободу у виявленні та використанні раніше невиявлених закономірностей, які могли бути пропущені "експертами". Це часто відбувається за рахунок неконтрольованих методів, які вимагають більше навчальних даних і повільніше досягають прийнятної продуктивності, підвищених вимог до обчислень і зберігання даних під час процесу дослідження, а також, можливо, більшої вразливості до невиявлених артефактів або аномалій у навчальних даних. нерелевантними або визнаними шахрайськими людьми, але яким алгоритм неконтрольованого навчання надає непропорційно велику вагу.

Поширені сімейства алгоритмів, які використовуються в навчанні без нагляду, включають: (1) кластеризацію, (2) виявлення аномалій, (3) нейронні мережі (зауважте, що не всі нейронні мережі є безконтрольними; їх можна навчати керованими, неконтрольованими, напівкерованими або методи підкріплення) та (4) моделі прихованих змінних.

- Методи кластеризації включають ієрархічну кластеризацію, [4] k-середніх, [5] моделі сумішей, DBSCAN [6] та алгоритм OPTICS

- Методи виявлення аномалій включають локальний фактор викидів і ізоляційний ліс

- Підходи до вивчення моделей латентної змінної включають алгоритм очікування-максимізації, метод моментів і методи сліпого поділу сигналів (аналіз основних компонентів, аналіз незалежних компонентів, невід'ємна матрична факторизація, розкладання сингулярних значень)

- Методи нейронних мереж включають автокодери, мережі глибоких переконань, Hebbian навчання, генеративні змагальні мережі (GAN) і самоорганізуючі карти.

Метод моментів

Метод моментів є одним із статистичних підходів до неконтрольованого навчання[7]. Метод моментів пов'язує невідомі параметри моделі, що цікавлять нас, з моментами однієї або декількох випадкових величин. Емпірична оцінка цих моментів з наявних вибірок даних використовується для обчислення найбільш ймовірних розподілів значень параметрів. Крім спостережуваних змінних, наявних у навчальній та вхідній вибірках даних, припускається, що існує ряд неспостережуваних латентних змінних, які визначають категоризацію кожної з них. Показано, що метод моментів є ефективним для навчання параметрів моделей латентних змінних, коли ряд неспостережуваних

латентних змінних визначають категоризацію кожної з них. Одним із практичних прикладів застосування моделей латентних змінних у машинному навчанні є тематичне моделювання - статистична модель для передбачення слів (спостережуваних змінних) у документі на основі його теми (латентної змінної). Було продемонстровано, що за певних умов метод моментів (метод тензорної декомпозиції) послідовно відновлює параметри великого класу моделей латентних змінних.

Алгоритм максимізації математичного сподівання є ще одним корисним методом для навчання моделей з латентними змінними. Однак він може потрапити в пастку локального оптимуму, і збіжність до істинних невідомих параметрів моделі не гарантується. На відміну від нього, метод моментів гарантує глобальну збіжність за певних умов[8].

Завдання проти методів

Традиційно контрольовані методи використовуються для завдань розпізнавання, тоді як неконтрольовані - для завдань генерації. З розвитком технологій деякі задачі використовують обидва методи, тоді як інші переключаються між ними. Наприклад, розпізнавання зображень починалося як суворо контрольоване, потім перетворилося на гібрид з неконтрольованим попереднім навчанням, а потім повернулося до контрольованого з введенням коефіцієнта відсіву, релевантності та адаптивного навчання.

Навчання.

На етапі навчання некерована мережа намагається відтворити надані дані і використовує помилки в змодельованих результатах для самокорекції (тобто коригування ваг та упереджень). Це можна порівняти з імітаційною поведінкою підлітків, які вивчають мову. Іноді помилка виражається як

низька ймовірність того, що відбудеться помилковий вихід, або як нестабільний, високоенергетичний стан в мережі.

Неконтрольовані методи використовують різноманітні алгоритми навчання, включаючи правило навчання Хопфілда, правило навчання Больцмана, контрастну дивергенцію, сон-неспання, варіаційний висновок, максимальне апостеріорне значення, вибірку Гіббса, а також зворотне поширення помилки реконструкції або репараметризацію прихованих станів.

3.3 Інструментарій прогнозування валютних ринків

Python - це високорівнева мова програмування загального призначення, що інтерпретується. Філософія проектування Python надає пріоритет читабельності коду завдяки використанню великих відступів. Структура мови та об'єктно-орієнтований підхід покликані допомогти програмістам у написанні зрозумілого, логічного коду для великих і складних проектів.

Python динамічно збирає сміття. Він підтримує структуровані (в тому числі процедурні), об'єктно-орієнтовані та функціональні парадигми програмування. Оскільки Python має всеосяжну стандартну бібліотеку, її часто називають мовою програмування на батарейках.

Гвідо ван Россум почав досліджувати Python як наступника мови програмування ABC наприкінці 1980-х років і випустив Python 0.9.0 у 1991 році. Python 2.0 був випущений у 2000 році з новими можливостями, такими як розуміння списків і система збору сміття, заснована на підрахунку посилань. Версія 2.7.18 стане останнім випуском Python перед його припиненням у 2020 році. У 2008 році вийшла версія Python 3.0, яка стала значним оновленням мови програмування. Мова не є повністю зворотньо сумісною, і більшість програм, написаних на Python 2, не можуть нормально функціонувати на Python 3. Python завжди був однією з найпопулярніших мов програмування.

Python був задуманий наприкінці 1980-х років Гвідо ван Россумом з Нідерландських Centrum Wiskunde & Informatica (CWI). Він є наступником мови програмування ABC, натхненної SETC, яка може обробляти винятки та взаємодіяти з операційною системою Amoeba. Його реалізація розпочалась у грудні 1989 року. Як головний розробник проекту, Ван Россум брав на себе повну відповідальність за проект до 12 липня 2018

року, коли оголосив про своє "вічне свято" титулом "дружнього диктатора життя" Пітона. Він знайшов своє відображення у своїй довгій історії спільнотою Python. Зобов'язання бути головним менеджером проекту. Зараз він займає керівну посаду в якості члена ради наглядових органів із п'яти осіб. У січні 2019 року активні розробники ядра Python обрали Бретта Кеннона, Ніка Коглана, Баррі Варшаву, Керол Віллінг та Ван Россума членами комітету з п'яти осіб. З тих пір Гвідо ван Россум відкликав свою кандидатуру до ради 2020 року.

Python 2.0 був випущений 16 жовтня 2000 року з багатьма основними новими можливостями, включаючи збирач сміття для виявлення циклів та підтримку Unicode.

Python 3.0 був випущений 3 грудня 2008 року. Це серйозний перегляд мови, який не повністю сумісний з іншою стороною. Багато його основних функцій було перенесено до версій Python 2.6.x та 2.7.x. До розподілу Python 3 входить утиліта 2to3, яка автоматично (принаймні частково) перетворює код Python 2 у Python 3.

Спочатку планувалося, що Python 2.7 закінчиться у 2015 році, але цей термін було перенесено на 2020 рік через побоювання, що значний обсяг застарілого коду не може бути легко перенесений на Python 3. Жодних додаткових оновлень безпеки чи інших покращень не буде. Наприкінці свого життя Python 2 підтримує лише Python 3.6.x та наступні версії.

Оскільки всі версії Python містять уразливості в безпеці, Python 3.9.2 і 3.8.8 були прискорені, що призвело до можливості віддаленого виконання коду і отруєння веб-кешу.

Python - багата на парадигми мова програмування. Багато її можливостей підтримують функціональне та аспектно-орієнтоване програмування, включаючи проектування метапрограм та мета-об'єктів (магічний метод). Розширення підтримує численні додаткові парадигми, такі як контрактне проектування та логічне програмування.

Python використовує підрахунок посилань і збирачі сміття в поєднанні з динамічним введенням тексту. Збирач сміття реалізує цикл керування пам'яттю. Також підтримується динамічне розпізнавання імен (пізнє зв'язування) для зв'язування імен методів та змінних під час виконання.

Python призначений для надання певної підтримки функціонального програмування на основі Lisp. За допомогою функцій фільтрування, відображення та скорочення; перелічено розуміння генераторів, словників, наборів та виразів. Стандартна бібліотека має два модулі (itertools та functools) для реалізації функціональних інструментів, запозичених у Haskell та Standard ML.

Основні принципи мови викладені в документі Zen of Python (PEP 20), який включає такі сентенції:

- Красиво-краще, ніж потворно.
- Явне краще, ніж неявне.
- Просте - краще, ніж складне.
- Комплекс краще, ніж комплекс.

- Розрахунок читабельності.

Ядро Python не містить усіх функцій, але його розроблено таким чином, щоб його можна було розширювати (за допомогою модулів). Ця компактна модульність робить її популярним методом додавання програмованих інтерфейсів до існуючих додатків. Натхненням для бачення компактних базових мов, великих стандартних бібліотек та легко розширюваних трансляторів стало незадоволення Ван Россама системою ABC, яка підтримує протилежний підхід. Дозволяючи розробникам обирати методи кодування, Python прагне спростити та зменшити граматичну плутанину. Python підкреслює філософію проектування "Повинен бути один спосіб зробити це - бажано лише один очевидний спосіб" на відміну від девізу Perl "Існує безліч способів зробити це". Алекс Мартеллі, науковий співробітник Python Software Foundation та автор книги про мову, заявив: "Термін "розумний" не вважається компліментом у культурі Python".

Програмісти Python уникають необачної оптимізації та відмовляються виправляти несуттєві компоненти еталонної реалізації CPython. Ці зміни відбуваються за рахунок наочності, що призводить до незначного збільшення швидкості. Коли продуктивність має вирішальне значення, Python-програмісти можуть переносити чутливі до часу функції до розширень, написаних мовами на кшталт C, або використовувати чутливий до часу компілятор PyPy. Також доступний Cython, який перетворює Python-скрипти на C і робить виклики API на рівні C безпосередньо до інтерпретатора Python.

Важливою метою розробників Python є збереження задоволення від використання. Це відображається в назві мови (данина поваги британській

комедійній компанії Monty Python), а іноді і в цікавих підручниках та довідкових методах, таких як стандартний foo та bars.

Користувачів та шанувальників Python, особливо тих, хто вважається обізнаним або досвідченим, часто називають Pythonistas. Python має бути простою для читання мовою. Його формат візуально не бентежить, він часто використовує англійські ключові слова, тоді як інші мови використовують розділові знаки. На відміну від багатьох інших мов, він не використовує фігурні дужки для розділення блоків і допускає крапку з комою після операторів, але крапка з комою використовується рідко. Він має менше граматичних винятків та особливих випадків, ніж C або Pascal.

PyCharm – це інтегроване середовище розробки (IDE), що використовується в комп'ютерному програмуванні, зокрема для мови Python. Він розроблений чеською компанією JetBrains. Він забезпечує аналіз коду, графічний налагоджувач, інтегрований тестер одиниць, інтеграцію із системами контролю версій (VCSes), а також підтримує веб-розробку з Django, а також аналіз даних з Anaconda.

PyCharm є крос-платформним, з версіями для Windows, macOS та Linux. Видання Community випускається під ліцензією Apache, а також існує Professional Edition з додатковими функціями - випущено під власною ліцензією.

Особливості

- Допомога та аналіз коду із завершенням коду, виділенням синтаксису та помилок, інтеграцією лінтера та швидкими виправленнями

- Навігація по проекту та коду: спеціалізовані перегляди проекту, перегляди структури файлів і швидкий перехід між файлами, класами, методами та використанням

- Рефакторинг Python: включає перейменування, метод вилучення, введення змінної, введення константи, підтягування, натискання та інші

- Підтримка веб-фреймворків: Django, web2py і Flask [лише професійна версія][8]

- Інтегрований налагоджувач Python

- Інтегроване модульне тестування з порядковим покриттям коду

- Розробка Google App Engine Python [лише професійна версія]

- Інтеграція контролю версій: уніфікований інтерфейс користувача для Mercurial, Git, Subversion, Perforce і CVS зі списками змін і злиттям

- Підтримка наукових інструментів, таких як matplotlib, numpy і scipy [лише професійне видання][9]

Він конкурує в основному з низкою інших IDE, орієнтованих на Python, включаючи PyDev від Eclipse і більш широку IDE Komodo.

PyCharm надає API, щоб розробники могли писати власні плагіни для розширення можливостей PyCharm. Кілька плагінів з інших IDE JetBrains також працюють з PyCharm. Існує більше 1000 плагінів, які сумісні з PyCharm.[10]

Бета-версія була випущена в липні 2010 року, а версія 1.0 з’явилася через 3 місяці. Версія 2.0 була випущена 13 грудня 2011 року, версія 3.0 - 24 вересня 2013 року, а версія 4.0 - 19 листопада 2014 року

PyCharm Community Edition, версія з відкритим кодом PyCharm, стала доступною 22 жовтня 2013 р.

scikit-image (раніше scikits.image) - це бібліотека зображень з відкритим кодом для мови програмування Python. [2] Він включає алгоритми сегментації, геометричного перетворення, маніпулювання кольоровим простором, аналізу, фільтрації, морфології, виявлення особливостей тощо. Він призначений для взаємодії з числовою та науковою бібліотеками Python NumPy та SciPy.

Проект scikit-image розпочався на scikits.image, а його автором був Стефан ван дер Вальт. Його назва походить від ідеї, що це "SciKit" (набір інструментів SciPy), розроблений та розповсюджений окремо стороннім розширенням SciPy. [4] Оригінальна база коду пізніше була широко переписана іншими розробниками. У листопаді 2012 року в різних наукових роботах наукові знання та наукові знання були описані як «ретельно модифіковані та популярні» [5]. Scikit-image також дуже активний у Google Summer of Code. [6]

scikit-image в основному написаний на Python, а деякі основні алгоритми написані на Cython для підвищення продуктивності.

РОЗДІЛ 4

РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ ВАЛЮТНИХ РИНКІВ

4.1 Аналіз варіантів використання системи

Найпростіша діаграма варіантів використання зображує взаємодію між користувачем і системою. Вона ілюструє зв'язок між користувачем і різними варіантами використання, в яких він бере участь. Діаграма варіантів використання слугує для ідентифікації різних категорій користувачів системи та варіантів використання. Зазвичай вона супроводжується додатковими формами діаграм. Варіанти використання зображуються у вигляді кіл або еліпсів.

Діаграма варіантів використання пропонує загальний огляд системи, незважаючи на те, що окремі варіанти використання можуть детально розглядати кожну можливість. Варіанти використання допомагають розвинути концептуальне розуміння системи, незважаючи на те, що їм бракує конкретики.

Варіанти використання є ефективним методом комунікації зі стейкхолдерами завдяки своїй простоті. Ці ілюстрації допомагають зацікавленим сторонам зрозуміти, як буде функціонувати система, моделюючи реальний світ.

Мета кейсу використання - продемонструвати динамічну природу системи. Додаткові діаграми та документація можуть бути використані для отримання вичерпної функціональної та технічної інформації про систему. Діаграми варіантів використання забезпечують графічне і спрощене представлення передбачуваної функціональності системи.

Елементи діаграми використання включають:

- Прямокутник з назвою системи у верхній частині, що представляє контекст системи.

- Еліпси, що представляють прецеденти (випадки використання) всередині прямокутника. Прецеденти описують систематичні дії, які користувач може виконувати, і результати, які вони призводять.

- Стилізовані людські персонажі, які називаються акторами, представляють ролі користувачів або інші сутності, що взаємодіють з системою.

На рисунку 4.1 зображена діаграма випадків використання, яка описує можливі дії користувача в системі.

Рисунок 4.1 — Діаграма варіантів використання

4.2 Діаграма IDEF0

IDEF0, складний акронім ("Icam DEFinition for Function Modeling", де ICAM означає "Інтегроване автоматизоване виробництво"), - це методологія функціонального моделювання для опису виробничих функцій, яка забезпечує мову функціонального моделювання для аналізу, розробки, реінжинірингу та інтеграції інформаційних систем, бізнес-процесів або аналізу інженерії програмного забезпечення.

IDEF0 базується на мові функціонального моделювання Structured Analysis and Design Technique (SADT) і є членом сімейства мов моделювання програмної інженерії IDEF.

Техніка функціонального моделювання IDEF0 призначена для моделювання рішень, дій та діяльності системи або організації[2]. Вона розвинулася з мови графічного моделювання Structured Analysis and Design Technique (SADT), створеної Дугласом Т. Россом (Douglas T. Ross) та компанією SofTech, Inc. У своєму початковому втіленні IDEF0 визначає мову графічного моделювання (синтаксис і семантику) та комплексну методологію розробки моделей[3]. Військово-повітряні сили США поставили перед розробниками SADT завдання "розробити метод аналізу та передачі функціональної перспективи системи за допомогою функціональної моделі. Використовуючи спрощені графічні засоби, IDEF0 має допомогти координувати системний аналіз і сприяти ефективному спілкуванню між аналітиком і замовником.

На рисунку 4.3 зображено діаграму IDEF0, яка описує процес прогнозування ціни.

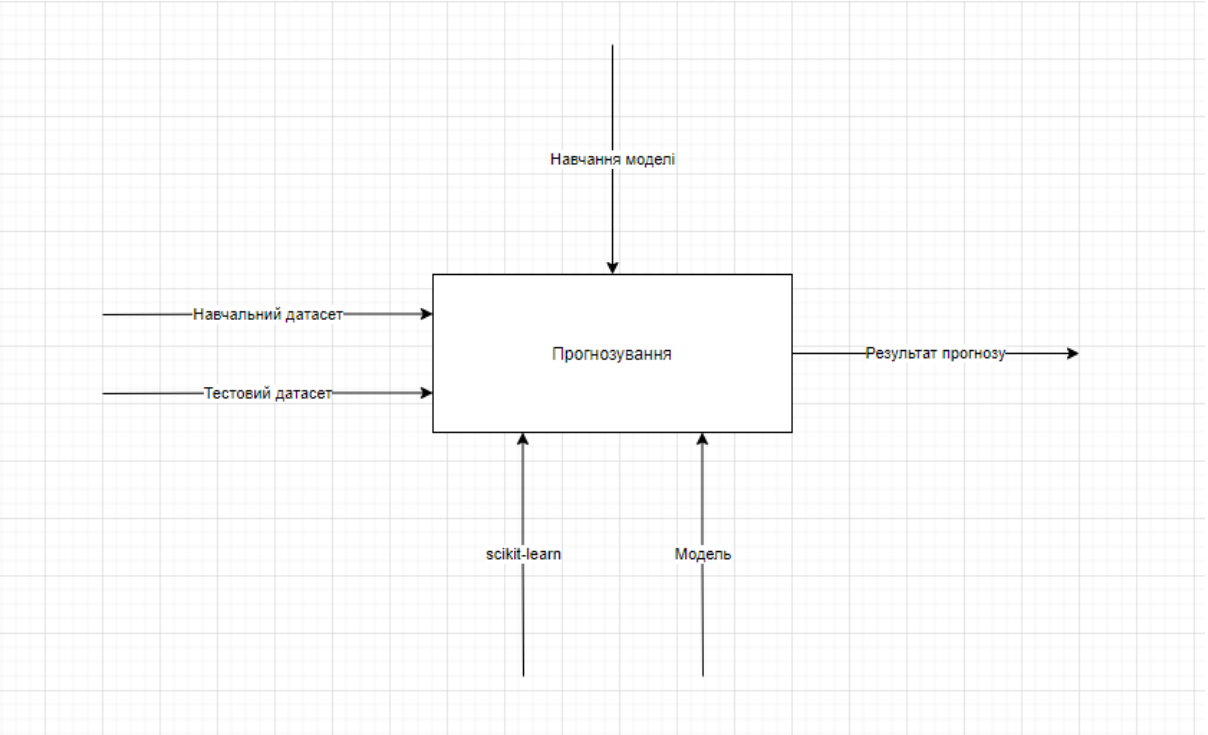

Рисунок 4.2 — Діаграма IDEF0

4.3 Діаграма компонентів

Діаграми компонентів в уніфікованій мові моделювання (UML) ілюструють, як компоненти об'єднуються для створення більших компонентів або програмних систем. Діаграми використовуються для зображення структури будь-якої складної системи.

Діаграма компонентів дозволяє переконатися, що необхідна функціональність системи є прийнятною. Ці діаграми також використовуються як засіб комунікації між розробником і складовими

системи. Використовуючи діаграми, програмісти та розробники можуть формалізувати дорожню карту реалізації, що дозволяє приймати більш обґрунтовані рішення щодо розподілу завдань та необхідних удосконалень. Діаграми компонентів дозволяють системним адміністраторам планувати заздалегідь, відображаючи логічні компоненти програмного забезпечення та їх взаємозв'язки.

Діаграма компонентів деталізує інформацію, що міститься в елементі мітки компонента. Прямокутний відсік, прикріплений до елемента компонента, є одним з методів ілюстрації наявних і необхідних інтерфейсів певного компонента.[2] Графічна конвенція "куля і розетки" є додатковим прийнятним методом для представлення інтерфейсів. Зв'язок між компонентом та інтерфейсом зображується суцільною лінією до компонента, який використовує інтерфейс з "льодяника" або кулі з назвою інтерфейсу. Півколо або вмістилище, позначене назвою інтерфейсу і з'єднане суцільною лінією з компонентом, якому потрібен цей інтерфейс, представляє необхідну залежність використання між ними. Перед міткою імені успадковані інтерфейси можуть відображатися за допомогою льодяника і символу повернення каретки. Щоб проілюструвати їхню взаємозалежність, з'єднайте розетку з льодяником суцільною лінією з правильною вершиною.

На рисунку 4.3 зображена діаграма компонентів системи, яка показує взаємодію між компонентами системи та утворення з них цілісної системи.

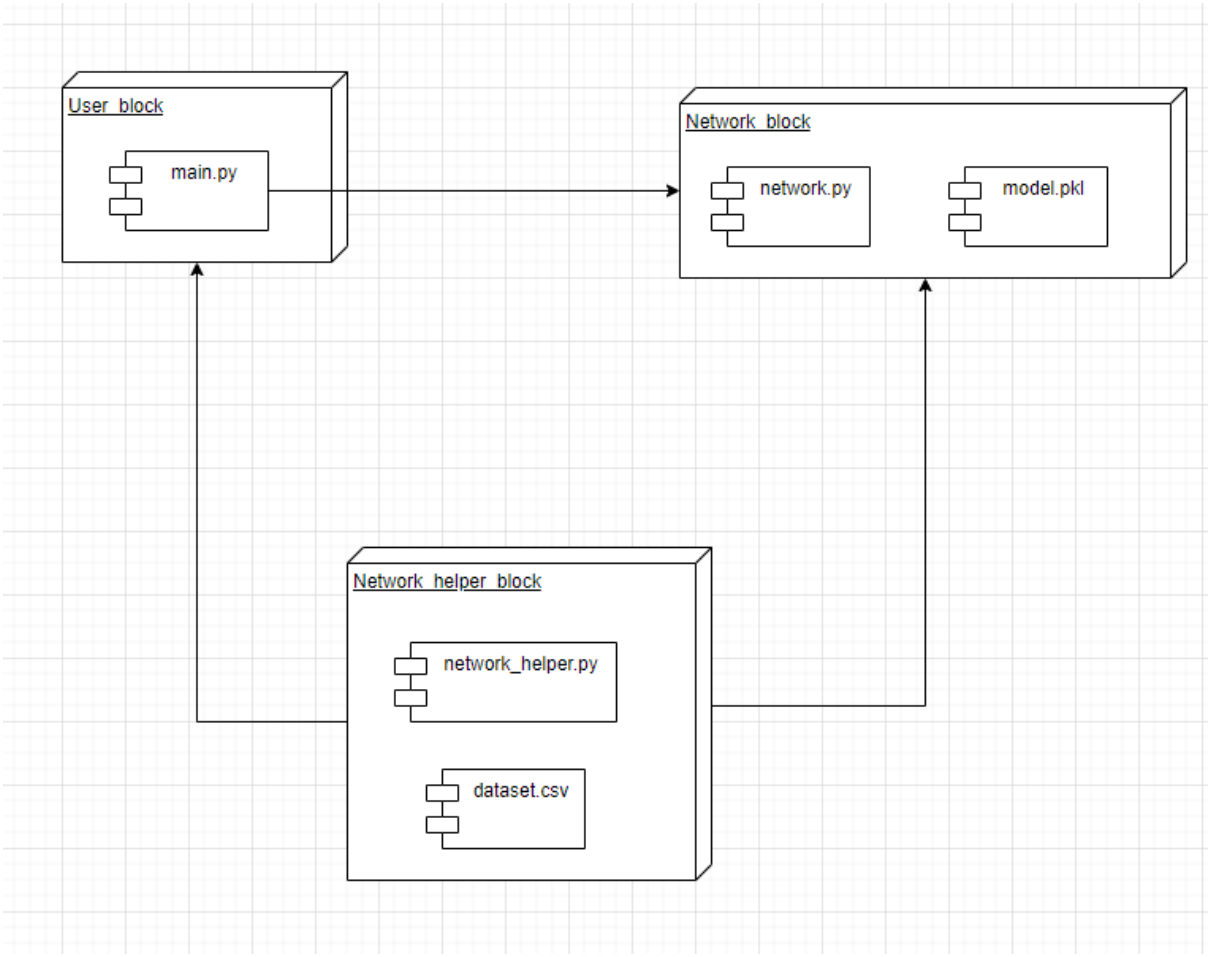

Рисунок 4.3 — Діаграма компонентів системи

4.4 Розробка основних механізмів

Початок роботи над створенням нейронної мережі починається з розроблення і навчання моделі нейронної мережі. Модель навчається на

своєму окремому датасеті. Частина програмного коду даного етапу представлена нижче.

```
df = pd.read_csv("dataset.csv")

x = df.iloc[:, :4]

y = df.iloc[:, 4:5]



model = tree.DecisionTreeClassifier(criterion="entropy")

model.fit(x, y)

model.score(x, y)



with open('model_year.pkl', 'wb') as f:

pickle.dump(model_year, f)

df = pd.read_csv("dataset.csv")

x = df.iloc[:, :9]

y = df.iloc[:, 9:10]
```

Для створення і навчання моделей в даному випадку використовується бібліотека scikit-learn.

Після навчання і збереження моделі слід написати функціональну частину, яка буде посередником між інтерфейсом користувача і моделями.

Частина коду «посередника» представлена нижче.

```
def get_month(arr):

arr = np.array(arr).reshape((1,-1))

with open("model.pkl", 'rb') as f:

model = pickle.load(f)

return str(model.predict(arr))
```

Посередник отримує набори вхідних даних, розпаковує моделі та передає їм вхідні дані для обробки, а результати повертає користувачеві в інтерфейс.

Згідно з принципами проектування програмного забезпечення, модуль користувацького інтерфейсу повинен відповідати виключно за взаємодію між користувачем і програмою, не несучи ніякого функціонального навантаження. Саме тому модуль інтерфейсу користувача призначений для того, щоб перехоплювати вхідні дані користувача, надсилати їх "посереднику", а потім виводити результат на екран. Нижче наведено частину вихідного коду інтерфейсного модуля.

```
import network as nh

def get_month():

feature1 = float(input('Enter feature 1 of 9: '))

feature2 = float(input('Enter feature 2 of 9: '))
```

```
feature3 = float(input('Enter feature 3 of 9: '))

feature4 = float(input('Enter feature 4 of 9: '))

feature5 = float(input('Enter feature 5 of 9: '))

feature6 = float(input('Enter feature 6 of 9: '))

feature7 = float(input('Enter feature 7 of 9: '))

feature8 = float(input('Enter feature 8 of 9: '))

feature9 = float(input('Enter feature 9 of 9: '))

data_array = [feature1, feature2, feature3, feature4, feature5, feature6, feature7, feature8, feature9]

print(nh.getres(data_array))
```

Як видно з програмного коду, модуль інтерфейсу відповідає лише за збір даних та вивід результатів, ніякі функціональні навантаження не реалізовані всередині цього модуля.

4.5 Створення навчальних вибірок

Набір даних - це збірник інформації. У випадку табличних даних набір даних відповідає одній або декільком таблицям бази даних, де кожен стовпець представляє змінну, а кожен рядок відповідає запису у відповідному наборі даних. Набір даних містить значення для кожної змінної, наприклад, висоти і ваги об'єкта, для кожного члена набору даних.

Кожне значення називається даними. Набори даних можуть також включати групу документів або файлів.

У сфері відкритих даних набір даних - це одиниця виміру інформації, опублікованої у сховищі відкритих даних з публічним доступом. Європейський портал відкритих даних містить понад 500 000 наборів даних. Інші фактори (джерела даних у режимі реального часу, нереляційні набори даних тощо) ускладнюють досягнення консенсусу з цього питання.

Для того, щоб навчити нейронну мережу, необхідно створити великі набори даних, що містять десятки тисяч прикладів даних. Існує пряма залежність між розміром навчальної вибірки та точністю прогнозів навченої моделі для таких розмірів наборів даних.

Для навчання моделі прогнозування ми зібрали історичні дані в набір даних.

```
January,February,March,April,May,June,July,August,September,October
48010,48010,54500,54900,54900,71800,71800,66050,72994,73600
48000,47900,54000,55000,55000,69800,69900,67000,71436,70000
48900,48900,55000,54500,54750,72000,72400,65800,73400,74800
```

Рисунок 4.4 — Приклад датасету

4.6 Тестування

Для оцінки розробленої системи було обрано процедуру тестування "чорного ящика". Тестування "чорного ящика" - це метод тестування програмного забезпечення, який перевіряє функціональність програми без вивчення її внутрішніх структур або операцій. Цей підхід до тестування застосовується майже на всіх рівнях тестування програмного забезпечення, включаючи тестування модулів, інтеграції, системи та прийнятності. Іноді його називають тестуванням на основі специфікацій.

Знання коду програми, її внутрішньої структури або програмування в цілому не є необхідним. Тестувальник знає про передбачувану функціональність програмного забезпечення, але не знає про її реалізацію. Наприклад, оцінювач знає, що певні вхідні дані дають незмінний результат, але не знає, як програма генерує цей результат.

На основі специфікацій та вимог до програми будуються тестові кейси. Зазвичай, тестові кейси отримують із зовнішніх описів програмного забезпечення, таких як специфікації, вимоги та параметри проектування. Хоча функціональні тести є найпоширенішими, також можуть використовуватися нефункціональні тести. Розробник тестів обирає як допустимі, так і недопустимі вхідні дані і визначає правильний вихід, часто використовуючи тестовий оракул або відомий попередній результат, не маючи жодних знань про внутрішню структуру об'єкта тестування.

Тест-кейси представлені в таблиці 4.1.

Таблиця 4.1

Тестування моделі прогнозування на рік

№ п.п.	**Опис**	**Очікуваний результат**	**Отриманий результат**
1	Введення даних, що вибиваються з ряду	Виведення результату, який не відповідає дійсності, проте є точним з огляду на введені дані	Виведення результату, який не відповідає дійсності, проте є точним з огляду на введені дані
2	Введення текстових даних замість числових	Відповідне повідомлення про помилку в роботі програми	Відповідне повідомлення про помилку в роботі програми
3	Введення занадто великого числа	Виведення результату, який не відповідає	Виведення результату, який не відповідає

		дійсності, проте є точним з огляду на введені дані	дійсності, проте є точним з огляду на введені дані
4	Введення порожніх полів	Відповідне повідомлення про помилку в роботі програми	Відповідне повідомлення про помилку в роботі програми
5	Введення усіх полів великими числами	Виведення результату, який не відповідає дійсності, проте є точним з огляду на введені дані	Виведення результату, який не відповідає дійсності, проте є точним з огляду на введені дані
6	Введення адекватних вхідних даних	Виведення прогнозу	Виведення прогнозу
7	Введення усіх полів порожніми	Відповідне повідомлення про	Відповідне повідомлення про

		помилку в роботі програми	помилку в роботі програми

Тестування показує повну функціональну вірність, відсутність неточностей і помилок у написаній системі.

ВИСНОВКИ

У цій дипломній роботі був проведений аналіз і прогнозування трендів валютних ринків методами машинного навчання. Головною метою було визначити ефективні моделі для прогнозування змін курсів валют і розробити стратегії торгівлі на основі цих прогнозів. Була проведена попередня обробка і аналіз даних, включаючи видалення аномалій, заповнення пропущених значень та нормалізацію даних.

Для розробки моделі прогнозування був використаний метод машинного навчання “дерева рішень”. Модель була навчена на історичних даних та протестована на незалежних вибірках для оцінки її точності та прогностичної здатності.

Результати дослідження показали, що методи машинного навчання можуть бути успішно застосовані для прогнозування трендів валютних ринків. Деякі моделі, зокрема нейронні мережі, демонстрували високу точність і здатність передбачати майбутні зміни курсів валют.

Отримані результати можуть бути корисні для трейдерів, інвесторів і компаній, які займаються торгівлею на валютному ринку. Вони можуть використовувати розроблену модель для створення прогнозів і підтримки прийняття рішень щодо купівлі або продажу валютних позицій.

Однак, слід зазначити, що ринки є складними та непередбачуваними, і жодна модель не може забезпечити абсолютну точність у прогнозуванні курсів валют. Прийняття торговельних рішень має базуватися не тільки на результатах моделей, але й на фундаментальному аналізі, геополітичних факторах та інших зовнішніх впливах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259.

2. Kleene, S.C. (1956). "Representation of Events in Nerve Nets and Finite Automata". Annals of Mathematics Studies (34). Princeton University Press. pp. 3–41. Retrieved 17 June 2017.

3. Hebb, Donald (1949). The Organization of Behavior. New York: Wiley. ISBN 978-1-135-63190-1.

4. Farley, B.G.; W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". IRE Transactions on Information Theory. 4 (4): 76–84. doi:10.1109/TIT.1954.1057468.

5. Haykin (2008) Neural Networks and Learning Machines, 3rd edition

6. Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain". Psychological Review. 65 (6): 386–408. CiteSeerX 10.1.1.588.3775. doi:10.1037/h0042519. PMID 13602029.

7. Werbos, P.J. (1975). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.

8. Rosenblatt, Frank (1957). "The Perceptron—a perceiving and recognizing automaton". Report 85-460-1. Cornell Aeronautical Laboratory.

9. Olazaran, Mikel (1996). "A Sociological Study of the Official History of the Perceptrons Controversy". Social Studies of Science. 26 (3): 611–659. doi:10.1177/030631296026003005. JSTOR 285702. S2CID 16786738.

10. Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.

11. Ivakhnenko, A. G. (1973). Cybernetic Predicting Devices. CCM Information Corporation.

12. Ivakhnenko, A. G.; Grigor′evich Lapa, Valentin (1967). Cybernetics and forecasting techniques. American Elsevier Pub. Co.

13. Schmidhuber, Jürgen (2015). "Deep Learning". Scholarpedia. 10 (11): 85–117. Bibcode:2015SchpJ..1032832S. doi:10.4249/scholarpedia.32832.

14. Dreyfus, Stuart E. (1 September 1990). "Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure". Journal of Guidance, Control, and Dynamics. 13 (5): 926–928. Bibcode:1990JGCD...13..926D. doi:10.2514/3.25422. ISSN 0731-5090.

15. Mizutani, E.; Dreyfus, S.E.; Nishio, K. (2000). "On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application". Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium. IEEE: 167–172 vol.2. doi:10.1109/ijcnn.2000.857892. ISBN 0-7695-0619-4. S2CID 351146.

16. Kelley, Henry J. (1960). "Gradient theory of optimal flight paths". ARS Journal. 30 (10): 947–954. doi:10.2514/8.5282.

17. "A gradient method for optimizing multi-stage allocation processes". Proceedings of the Harvard Univ. Symposium on digital computers and their applications. April 1961.

18. Minsky, Marvin; Papert, Seymour (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press. ISBN 978-0-262-63022-1.

19. Linnainmaa, Seppo (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters) (in Finnish). University of Helsinki. pp. 6–7.

20. Linnainmaa, Seppo (1976). "Taylor expansion of the accumulated rounding error". BIT Numerical Mathematics. 16 (2): 146–160. doi:10.1007/bf01931367. S2CID 122357351.

21. Dreyfus, Stuart (1973). "The computational solution of optimal control problems with time lag". IEEE Transactions on Automatic Control. 18 (4): 383–385. doi:10.1109/tac.1973.1100330.

22. Werbos, Paul (1982). "Applications of advances in nonlinear sensitivity analysis" (PDF). System modeling and optimization. Springer. pp. 762–770.

23. Mead, Carver A.; Ismail, Mohammed (8 May 1989). Analog VLSI Implementation of Neural Systems (PDF). The Kluwer International Series in Engineering and Computer Science. 80. Norwell, MA: Kluwer Academic Publishers. doi:10.1007/978-1-4613-1639-8. ISBN 978-1-4613-1639-8.

24. David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams , "Learning representations by back-propagating errors ," Nature', 323, pages 533–536 1986.

25. J. Weng, N. Ahuja and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively," Proc. International Joint Conference on Neural Networks, Baltimore, Maryland, vol I, pp. 576–581, June 1992.

26. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation of 3-D objects from 2-D images," Proc. 4th International Conf. Computer Vision, Berlin, Germany, pp. 121–128, May 1993.

27. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation using the Cresceptron," International Journal of

Computer Vision, vol. 25, no. 2, pp. 105–139, Nov. 1997.

28. J. Schmidhuber., "Learning complex, extended sequences using the principle of history compression," Neural Computation, 4, pp. 234–242, 1992.

29. Domingos, Pedro (22 September 2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. chapter 4: Basic Books. ISBN 978-0465065707.

30. Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory.". In D. E. Rumelhart; J. L. McClelland; PDP Research Group (eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. 1. pp. 194–281. ISBN 978-0-262-68053-0.

31. Ng, Andrew; Dean, Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning". arXiv:1112.6209 [cs.LG].

32. Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press.

33. Cireşan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". Neural Computation. 22 (12): 3207–3220. arXiv:1003.0358. doi:10.1162/neco_a_00052. ISSN 0899-7667. PMID 20858131. S2CID 1918673.

34. Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," In 20th International Conference Artificial Neural Networks (ICANN), pp. 92–101, 2010. doi:10.1007/978-3-642-15825-4_10.

35. 2012 Kurzweil AI Interview Archived 31 August 2018 at the Wayback Machine with Jürgen Schmidhuber on the eight competitions won by his Deep Learning team 2009–2012

36. "How bio-inspired deep learning keeps winning competitions | KurzweilAI". www.kurzweilai.net. Archived from the original on 31 August 2018. Retrieved 16 June 2017.

37. Graves, Alex; and Schmidhuber, Jürgen; Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks, in Bengio, Yoshua; Schuurmans, Dale; Lafferty, John; Williams, Chris K. I.; and Culotta, Aron (eds.), Advances in Neural Information Processing Systems 22 (NIPS'22), 7–10 December 2009, Vancouver, BC, Neural Information Processing Systems (NIPS) Foundation, 2009, pp. 545–552.

38. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855–868. CiteSeerX 10.1.1.139.4502. doi:10.1109/tpami.2008.137. PMID 19299860. S2CID 14635907.

39. Graves, Alex; Schmidhuber, Jürgen (2009). Bengio, Yoshua; Schuurmans, Dale; Lafferty, John; Williams, Chris editor-K. I.; Culotta, Aron (eds.). "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks". Neural Information Processing Systems (NIPS) Foundation. Curran Associates, Inc: 545–552.

40. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (May 2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855–868. CiteSeerX 10.1.1.139.4502. doi:10.1109/tpami.2008.137. ISSN 0162-8828. PMID 19299860. S2CID 14635907.