

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

До захисту допущено
Завідувач кафедри ММСА
Оксана ТИМОЩУК
«___» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системний аналіз і управління»

спеціальності 124 «Системний аналіз»

на тему: «Альтернативні методи прогнозування макроекономічних процесів»

Виконав:

студент IV курсу, групи КА-71

Орел Євгеній Владиславович _____

Керівник:

професор, д.т.н., Бідюк Петро Іванович _____

Консультант з економічного розділу:

доцент, к.е.н., Розщина Надія Василівна _____

Консультант з нормоконтролю:

доцент, к.т.н., Коваленко Анатолій Єпіфанович _____

Рецензент:

професор, д.фіз-мат.н., Телейник Сергій Федорович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань

Студент _____

Київ
2021

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

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
Завідувач кафедри ММСА
Оксана ТИМОЩУК

«__» _____ 2021 р.

ЗАВДАННЯ

на дипломну роботу студенту
Орлу Євгенію Владиславовичу

1. Тема роботи «Альтернативні методи прогнозування макроекономічних процесів», керівник роботи Бідюк Петро Іванович, професор кафедри ММСА, затверджені наказом по університету від «26» травня 2021 р. №1344-с
2. Термін подання студентом роботи 08.06.2021.
3. Вихідні дані до роботи: макроекономічні та фінансово економічні дані. Дані отримані з The World Bank та Yahoo.
4. Зміст роботи: аналітичний огляд автомобільного ринку, теоретичні основи регресійного аналізу, дерева рішень та їх модифікації, прогнозування ціни на автомобіль на основі автомобільного ринку України, функціонально-вартісний аналіз програмного продукту.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) робота ПП, результати.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н. В., доцент		

7. Дата видачі завдання 1.04.2021

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою	13.04.2021	
2	Підготовка першого розділу	22.04.2021	
3	Підготовка другого розділу	1.05.2021	
4	Розробка програмного продукту	15.05.2021	
5	Підготовка третього розділу	18.05.2021	
6	Підготовка економічної частини	30.05.2021	
7	Оформлення розділів відповідно до нормоконтролю	2.06.2021	
8	Оформлення дипломної роботи	4.06.2021	
9	Підготовка презентації доповіді	7.06.2021	

Студент _____

Євгеній ОРЕЛ

Керівник _____

Петро БІДЮК

РЕФЕРАТ

Дипломна робота: 143., 2 табл., 78 рис., 2 додатків та 26 джерел.

МАКРОЕКОНОМІКА, FACEBOOK, NETFLIX, ВВП,
ПРОГНОЗУВАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ.

Прогнозування має величезне значення для розрахунку цін на електроенергію, скільки потрібно виготовити її, очікувані прибутки від бізнесу, очікуване зростання ВВП країни тощо. Зачасту використовують математичні або статистичні моделі для прогнозу на ринку, однак вони не завжди працюють в такому разі використовують альтернативні моделі для прогнозування які дають кращі результати ніж математичні.

Метою даної роботи стало дослідити альтернативні моделі для прогнозу макроекономічних та фінансових процесів які використовуються на ринку та дослідити їх якість на вибраних часових рядах, а саме: ВВП України, ВВП Польщі, курси акцій компаній Facebook та Netflix. Та використати ці моделі для моделювання та прогнозу заданих процесів.

Щоб досягти поставленої мети необхідно розв'язати такі задачі:

1. Аналіз інформаційного ринку для отримання даних.
2. Аналіз статистичних моделей для прогнозування процесів
3. Аналіз методів перетворення часових рядів
4. Вибір альтернативних методів
5. Підготовка даних для застосування альтернативних методів
6. Моделювання альтернативних моделей на кожному з часових рядів
7. Порівняння роботи моделей з різними рядами
8. Аналіз ефективності створених моделей

Об'єктом дослідження є історичні дані коливання ВВП та акцій, які потребують підготовки для подальшого застосування.

Предметом дослідження є альтернативні методи прогнозування макроекономічних даних.

В даній роботі проведено дослідження застосування альтернативних моделей на обраних часових рядах, та приведено найважливіші регресори для кожної з моделей.

В результаті дослідження були отримані результати що на підготовлених даних найраці результати дала лінійна та Баєсівська регресії. В подальшому можна розвивати працю для дослідження інших моделей.

ABSTRACT

Thesis: 143., 2 tables., 78 fig., 2 appendices. and 26 sources.

MACROECONOMICS, FACEBOOK, NETFLIX, GDP, FORECASTING, INTELLECTUAL DATA ANALYSIS.

Forecasting is of great importance for calculating electricity prices, how much to produce, expected business profits, expected GDP growth, and so on. Often use mathematical or statistical models for forecasting in the market, but they do not always work in this case use alternative models for forecasting that give better results than mathematical.

The aim of this work was to investigate alternative models for forecasting macroeconomic and financial processes used in the market and to investigate their quality on selected time series, namely: GDP of Ukraine, GDP of Poland, stock prices of Facebook and Netflix. And use these models to model and predict given processes.

To achieve this goal you need to solve the following tasks:

1. Analysis of the information market to obtain data.
2. Analysis of statistical models for process forecasting
3. Analysis of methods for converting time series
4. Choice of alternative methods
5. Preparation of data for the application of alternative methods
6. Modeling of alternative models on each of the time series
7. Comparison of models with different series
8. Analysis of the effectiveness of the created models

The object of the study is historical data on GDP fluctuations and stocks that need to be prepared for further application.

The subject of the study is alternative methods of forecasting macroeconomic data.

In this paper, a study of the application of alternative models on selected time series, and the most important regressors for each of the models.

As a result of the study, the results were obtained that on the prepared data the best results were given by linear and Bayesian regressions. In the future, you can develop work to study other models.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ	11
ВСТУП.....	12
РОЗДІЛ 1 ЗАДАЧІ ПРОГНОЗУВАННЯ ТА МОДЕЛЮВАННЯ НЕСТАЦІОНАРНИХ ТА СТАЦІОНАРНИХ МАКРОЕКОНОМІЧНИХ ПРОЦЕСІВ	14
1.1 Описові статистики процесу та графічний аналіз.....	14
1.2 Регресійний аналіз.....	15
1.3 Нелінійна регресія	16
1.4 Моделі із дискретною залежною змінною	17
1.5 Селекція моделей.....	18
1.6 Моделювання нестационарних процесів.....	19
1.7 Види нестационарних процесів	20
ВИСНОВКИ ДО РОЗДІЛУ 1	22
ПОСТАНОВКА ЗАДАЧІ	22
РОЗДІЛ 2 МЕТОДИ ПРОГНОЗУВАННЯ МАКРОЕКОНОМІЧНИХ ПРОЦЕСІВ	23
2.1 Моделі прогнозу макроекономічних стаціонарних процесів	23
2.2 Поліноміальна регресія.....	24
2.3 Лінійна модель регресії.....	25
2.4 Регресійна модель дерева рішень.....	26
2.5 Модель прогнозу на основі ковзного середнього.....	27
2.6 Модель прогнозу на основі авторегресії	28
2.7 Модель найближчих сусідів	29

ВИСНОВОК ДО РОЗДІЛУ 2	30
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ПРОГНОЗУ ТА ЇХ	
ОЦІНКА	31
3.1 Алгоритм побудови моделей прогнозу	31
3.2 Підготовка даних для прогнозування	32
3.3 Дослідження та моделювання ВВП України	33
3.3.1 Описові характеристики процесу	33
3.3.2. Моделювання дерева рішень.....	36
3.3.3. Моделювання багатовимірної лінійної регресії.....	38
3.3.4. Моделювання методу найближчих сусідів.....	40
3.3.5. Моделювання Бассової регресії	41
3.4 Дослідження та моделювання ВВП Польщі	43
3.4.1 Описові характеристики процесу	43
3.4.2. Моделювання дерева рішень.....	46
3.4.3. Моделювання багатовимірної лінійної регресії.....	48
3.4.4. Моделювання методу найближчих сусідів.....	50
3.4.5. Моделювання Бассової регресії	51
3.5 Дослідження та моделювання курсу акцій Facebook	53
3.5.1 Описові характеристики процесу.....	53
3.5.2. Моделювання дерева рішень.....	56
3.5.3. Моделювання багатовимірної лінійної регресії.....	58
3.5.4. Моделювання методу найближчих сусідів.....	60
3.5.5. Моделювання Бассової регресії	61
3.5 Дослідження та моделювання курсу акцій Netflix	63
3.5.1 Описові характеристики процесу.....	63
3.5.2. Моделювання дерева рішень.....	66
3.5.3. Моделювання багатовимірної лінійної регресії.....	68
3.5.4. Моделювання методу найближчих сусідів.....	70
3.5.5. Моделювання Бассової регресії	71
ВИСНОВОК ДО РОЗДІЛУ 3	73
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ	
ПРОГРАМНОГО ПРОДУКТУ	74
4.1 Постановка задачі проектування	75

4.2 Обґрунтування функцій програмного продукту	75
5.3 Обґрунтування системи параметрів ПП.....	78
4.4 Аналіз експертного оцінювання параметрів.....	81
4.5 Аналіз рівня якості варіантів реалізації функцій	86
4.6 Економічний аналіз варіантів розробки ПП.....	88
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	95
ВИСНОВКИ ДО РОЗДІЛУ 4	96
ВИСНОВКИ	97
ПЕРЕЛІК ПОСИЛАНЬ.....	98
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	101
ДОДАТОК Б ПЕРЕЛІК ГРАФІЧНОГО МАТЕРІАЛУ ДО ЗАХИСТУ	128

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

БД - база даних

ДР – дерева рішень

ПЗ - програмне забезпечення

MAPE (англ. Mean Absolute Error) – Абсолютна середня похибка в процентах

СППР – система підтримки прийняття рішень

ARIMA (англ. Autoregressive Integrated Moving Average) – авторегресія з інтегрованим ковзним середнім

RMSE (англ. Root Mean Squared Error) – Середньо квадратична помилка

IDE (англ. Integrated Development Environment) - інтегроване середовище розробки

АКФ – Автокореляційна функція

ЧАКФ – Частково автокореляційна функція

ВСТУП

Майбутнє багатьох процесів не відоме, однак в людей та компаній є величезна необхідність в прогнозі на наступний проміжок часу як рік, квартал, місяць тощо. Для того щоб прийняти відповідне рішення, по закупці товару, видатків на певні сектори економіки в бюджеті країни тощо. Прогнозування – один з найважливіших компонентів для прийняття рішень, для визначення які дії приведуть до яких результатів.

Прогноз необхідний для визначення можливостей, в обмеженнях яких повинні ставитися відповідні задачі планування, які б були реалістичні для виконання з точки зору економіки або функціонування приватного підприємства, корпорацій тощо. Для якісного прогнозування використовуються математичні методи, які можуть викликати складність розуміння для певного сектору людей, хоча з іншого боку виконують доволі якісні прогнози. Розвиток ІТ технологій робить величезний внесок у розвиток методів прогнозування та на даний етап ринку пропонуються не тлише математичні моделі, а й складніші моделі машинного навчання. Які в свою чергу використовуються коли не дають хороших результатів або зовсім не можна застосувати математичні моделі.

У даній роботі досліджено огляд існуючих на сьогоднішній час продуктів та засобів для обробки та аналізу даних покладених в основу не стаціонарних процесів, та їх класифікації. Розглянуто пакети, які використовуються для моделювання СППР. Описано популярні методи оцінки параметрів моделей та прогнозування для нестаціонарних процесів. На основі чотирьох моделей таких як багатовимірна регресія, дерева рішень, баясівська регресія та метод найближчих сусідів реалізована СППР, за допомогою якої можна визначити найкращу модель для виконання прогнозу.

У першому розділі розглянуто аналіз існуючих на сьогоднішній час продуктів для статистичної підготовки даних, аналіз роду нестаціонарних процесів та їх класифікації. Розглянуто види моделей, які використовуються для

розробки СППР. В другому розділ описує методи прогнозування не стаціонарних процесів та оцінювання параметрів моделей. В третьому розділі продемонстровано та основні характеристики моделей та описано проектування СППР. В розділі чотири розглянуто приклади роботи моделей на реальних фінансово-економічних та макро-економічних процесах. П'ятий розділ містить фінансно-вартісного аналізу створеного програмного продукту.

РОЗДІЛ 1 ЗАДАЧІ ПРОГНОЗУВАННЯ ТА МОДЕЛЮВАННЯ НЕСТАЦІОНАРНИХ ТА СТАЦІОНАРНИХ МАКРОЕКОНОМІЧНИХ ПРОЦЕСІВ

1.1 Описові статистики процесу та графічний аналіз

Побудова графіків розподілу, корелограм, діаграм потрібна для візуалізації процесів які нам дані, та часткового розуміння як розподілені ці процеси для підбору відповідної моделі для прогнозу. За допомогою Python пакетів здійснюється візуалізація вихідних даних в графічній інтерпритації.

Оскільки для кожної моделі машинного навчання є відповідні припущення до вхідних даних на поотрібно зрозуміти, а також перевірити чи ці процеси можуть використовуватися саме для вибраних моделей чи ні, та перевірити правдивість свідчень.

1.2 Регресійний аналіз

Регресійний аналіз – розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої. Займається пошуком моделі зв'язку, вираженої у функції регресії.

Якщо комбінація регресорів може бути зображена через первину комбінацію цих регресорів, тоді використовують регресійний аналіз. Отриманий вираз використовується для прогнозування значень, що приймає цільова змінна, в свою чергу отримується завдяки комбінації незалежних змінних. Однак більшість реальних процесів не підлягає для лінійної регресії, тому для цього використовуються інші моделі, а також якісна обробка даних. В свою чергу це рухає науковий світ для розвитку моделей машинного навчання.

1.3 Нелінійна регресія

Не завжди працює якісно лінійна регресія, особливо коли не регресори між собою залежні, або цільова змінна містить в собі комбінації їх. Для такого роду задач використовують нелінійні моделі такі як: дерева рішень, нелійна регресія, симплексний метод, тощо. В свою чергу ми скористаємося баясівським підходом, а також деревами рішень.

1.4 Моделі із дискретною залежною змінною

Більшість моделей приймають за вхідні та вихідні дані змінні з дискретною залежною змінною оскільки для комп'ютерного моделювання це покращує швидкість роботи моделі.

1.5 Селекція моделей

Селекція моделей відбувається за допомогою R^2 , MAPE, RMSE, критерій Дарбіна-Уотсона на тестувальній вибірці. Порівняння цих статистик між моделями можна визначити кращу з них. Більші статистик обраховується за допомогою бібліотеки Sklearn.

1.6 Моделювання нестационарних процесів

Зазвичай для моделювання нестационарних процесів використовуються модель ARIMA, де визначається порядок диференціювання відповідно до порядку тренду або диференціювання відбувається до того, поки ряд не набуде стаціонарного вигляду. Перевірка на нестационарність відбувається спешу за допомогою візуально-аналітичного аналізу. Проте такий метод є доволі не точним тому широко поширений роширений метод Діки-Фуллера. Були вибрані моделі такі які не потребують обовязковості стаціонарності ряду. Які вже були згадані раніше.

1.7 Види нестационарних процесів

Розрізняють два види нестационарності процесів – сильна та слабка. Хоча також виділюють з слабкої стаціонарності ще види:

1. Стаціонарність за коваріацією;
2. Стаціонарність другого порядку;
3. Стаціонарність в широкому сенсі;

Стаціонарність за коваріацією визначається як процес який має скінченне математичне сподівання та дисперсію та виконуються наступні умови:

- математичне сподівання не залежить від часу:

$$E[y(k)] = E[y(k - s)] = \mu = \text{const} \quad (1.1)$$

- дисперсія залишається постійною, на всьому часовому інтервалі досліджуваного процесу.

$$E\{[y(k) - \mu]^2\} = E\{[y(k - s) - \mu]^2\} = \sigma^2 = \text{const} \quad (1.2)$$

або

$$\text{var}[y(k)] = \text{var}[y(k - s)] = \sigma^2 = \text{const} \quad (1.3)$$

- Автоковаріація залишається незмінною в часі на всьому часовому інтервалі:

$$\begin{aligned} E\{[y(k) - \mu][y(k - s) - \mu]\} = \\ E\{[y(k - j) - \mu][y(k - j - s) - \mu]\} = y(k) = \text{const} \end{aligned} \quad (1.4)$$

або

$$\begin{aligned} cov[y(k)y(k-s)] = \\ cov[y(k-j), y(k-j-s)] = y(s) = const \end{aligned} \quad (1.5)$$

Однак незмінні значення середнього та дисперсії не є обов'язковістю для сильної стаціонарності. Можна зробити висновок, що обмеження слабкої стаціонарності є більш сильними порівнянно з умовами сильної стаціонарності. Для виконання умов нестаціонарності процеси вистачатиме однієї з умов (1.1), (1.2) або (1.3).

Для перевірки на нестаціонарність достатньо обчислити основні статистичні характеристики наданого процесу, а саме середнє, дисперсія та коваріація.

Висновки до розділу 1

Отже, одним з кращих мов програмування та популярним на ринку є Python, оскільки надає широкі можливості для аналізу, підготовки даних, а також надає певні реалізовані бібліотеки для моделей. Що в свою чергу спрощує аналіз даних та використання, так як на ринку велика Python спільнота, яка стикалася з певними роду проблемами при комп'ютерному моделювання процесів, та в свою чергу може допомогти починаючим аналітикам відповідно на різні питання. До недоліків можна віднести те, що необхідно розробляти всі етапи підготовки даних, перетворення їх до стаціонарності, якщо це вимагає модель, самостійно перевіряючи за допомогою різних тестів на стаціонарність процесу.

Однак, зважаючи на всі переваги та недоліки сформуємо основні напрямки роботи. Необхідно зробити обробку даних, підготувати їх до належного чину з одновимірного часового ряду перетворити у багатовимірний. Визначити найкращі параметри моделей, переконатися в адекватності моделі, а також проаналізувати важливість кожного з регресорів, якщо ще можливо з точки зору реалізації моделі.

Постановка задачі

1. Аналіз інформаційного ринку для отримання даних.
2. Аналіз статистичних моделей для прогнозування процесів
3. Аналіз методів перетворення часових рядів
4. Вибір альтернативних методів
5. Підготовка даних для застосування альтернативних методів
6. Моделювання альтернативних моделей на кожному з часових рядів
7. Порівняння роботи моделей з різними рядами
8. Аналіз ефективності створених моделей

РОЗДІЛ 2 МЕТОДИ ПРОГНОЗУВАННЯ МАКРОЕКОНОМІЧНИХ ПРОЦЕСІВ

2.1 Моделі прогнозу макроекономічних стаціонарних процесів

Наступним етапом полягає вибір структури моделей які будуть використані для прогнозування процесів. Більші поширених статистичних моделей, які прогнозують часовий ряд, потребують стаціонарності процесу. Тому було використано моделі ML для того щоб отримати кращі результати. З іншого боку використовувалися стратегії статистичних моделей для підготовки даних. Назвемо лаг до якої будуть відповідати моделі. Лагом порядку K називається час запізнення на K одиниць по входу. Також можна включити до структури моделі збурення які можуть набувати мультиплікативного, неперервного чи випадкового виглядів.

2.2 Поліноміальна регресія

Одна з найпоширеніших описових методів для опису нестационарних процесів з трендом є поліноміальна регресія:

$$y = w_0 + w_1 k + w_2 k^2 + \dots + w_m k^m + \varepsilon \quad (2.1)$$

де y – цільова змінна процесу;

$k = 0, 1, 2, \dots$ – дискретний час,

$w_i, i = 0, \dots, m$, параметри моделі;

m – порядок поліному.

Визначити порядок полінома допоможе кількість похідних, які можуть бути обчислені за допомогою найкращої поліноміальної регресії для процесу. Помилка вимірювання або недетерміновані збурення враховуються в $\varepsilon(k)$ та зазвичай припускається що вона розподілена нормально. Параметри моделі можуть знаходитися за допомогою різних оптимізаційних методів.

2.3 Лінійна модель регресії

$$y = \sum a_i x_i + \varepsilon(k) \quad (2.2)$$

де y – цільова змінна процесу;

$k = 0, 1, 2, \dots$ – дискретний час,

$w_i, i = 0, \dots, m$, параметри моделі;

m – порядок поліному.

Помилка вимірювання або недетерміновані збурення враховуються в $\varepsilon(k)$ та зазвичай припускається що вона розподілена нормально. Параметри моделі як і в попередньому пункті можуть знаходитися за допомогою різних оптимізаційних методів методів.

2.4 Регресійна модель дерева рішень

Дерева рішень є однією з найефективніших моделей для задачі регресії. Для цього визначають декілька визначень. Кореневий вузол – вузол в якого немає батьківської вершини. Розділення – вузол для дерева рішень розділяється на два можливі або більше. Вузол для рішення – це вузол який розділяється на наступні підвузли. Лист – вузол після якого немає розділення, тобто він не містить дочірніх вузлів. Відрізання – процес видалення дочірніх вузлів вузла для рішення.

Дерево рішень отримує оцінку, задаючи низку запитань до даних, кожне питання звужує наші можливі значення, поки модель не отримає достатню впевненість, щоб зробити єдиний прогноз. Порядок запитання, а також їх зміст визначаються моделлю. Окрім того, усі задані запитання мають форму так/ні.

Головною задачею для побудови дерева рішень як будуть обиратися незалежні змінні на кожному кроці прийняття рішення, та який критерій обрати для того щоб зробити рішення. Для цього зазвичай обирають такі метрики як коефіцієнт Джині або ентропію. Основною ‘роботою’ для регресійної моделі дерева рішень полягає ділити простір на два півпростори, поки кожна точка тренувальної вибірки не буде правильно розділена.

2.5 Модель прогнозу на основі ковзного середнього

Ковзне середнє використовується для того щоб позбутися різних невилуних випадкових осциляцій тобто для згладжування часового ряду. Якщо найбільш популярними згладжування використовують такі як: експонентні та проті зважені.

Якщо наводити наочний приклад, то просте ковзне середнє можна визначити як середнє за акцію за останній проміжок часу, зазвичай це дні, години. Відповідно якщо брати ковзне середнє 7 днів, отримуватимемо середню ціну за попередні тижні. Формула для ковзного середнього має вигляд:

$$MA(k) = \frac{y(k-n)+y(k-(n-1))+\dots+y(k)}{m} \quad (2.3)$$

Оскільки вибірка усереднюється в часі, тому має саме таку назву, як ковзне середнє. В той самий час формула загального ковзного середнього має вигляд:

$$MA(k) = \frac{w_1y(k)+w_2y(k-1)+\dots+w_ny(k-n+1)}{w_1+\dots+w_n} \quad (2.4)$$

де n – розмір ковзного вікна,

w_i – вагові коефіцієнти,

$y(t)$ – часовий ряд.

Якщо вагові коефіцієнти розподілені як експоненційний розподіл, то таке ковзне середнє називається експоненційним. Цей вид згладжування дає перевагу в тому, що більш важливі, нещодавні, значення часового ряду мають більшу вагу ніж ті що були раніше.

2.6 Модель прогнозу на основі авторегресії

Авторегресією визначають модель в якій з точки зору регресорів містяться її попередні значення. Це означає що модель враховує те, що було в минулому буде впливати на теперішнє, а також майбутнє. Модель авторегресії записують у вигляді:

$$y(k) = a_0 + a_1y(k - 1) + \dots + a_Ny(k - N) \quad (2.5)$$

Для того щоб визначити параметер авторегресії, тобто скільки попередніх значень нам необхідно застосовують АКФ та ЧАКФ. Для цього обирають число найбільш відмінних, або значущих, від нуля коефіцієнтів АКФ. Параметри моделі як і в попередніх лінійних моделях шукаються з допомогою різних оптимізаційних методів методів.

2.7 Модель найближчих сусідів

Модель найближчих сусідів розглядає елементи вибірки як вектори в багатовимірному просторі ознак, кожен із позначкою класу. В даній моделі є параметер K який визначає K найближчих сусідів серед яких буде використовуватися незалежної змінної значення. Якщо береться один вектор для того щоб зробити прогноз тоді з тренувально набору даних обирається K найближчих сусідів, які в свою чергу визначаються за допомогою норми векторного простору. Зазвичай норму беруть Евклідову. Тоді прогнозоване значення буде як середнє для K найближчих сусідів.

Висновок до розділу 2

В даному розділі було розглянуто моделі, які використовуються для моделювання стаціонарних та нестаціонарних рядів. Статистичні методи які враховують минулі стани процеси, яку в свою чергу можна використати для того щоб підготувати дані для використання в моделях машинного навчання. Також статистичні методи для вибору параметрів для авторегресії. Однією з недоліків використання моделей може слугувати складність машинного обчислення, однак з розвитком інформаційних технологій такі моделі з кожним роком стають працювати все швидше і швидше.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ПРОГНОЗУ ТА ЇХ ОЦІНКА

3.1 Алгоритм побудови моделей прогнозу

Для вибору найкращої моделі для прогнозу було використано наступний алгоритм прийняття рішення зображений на рис 3.1.

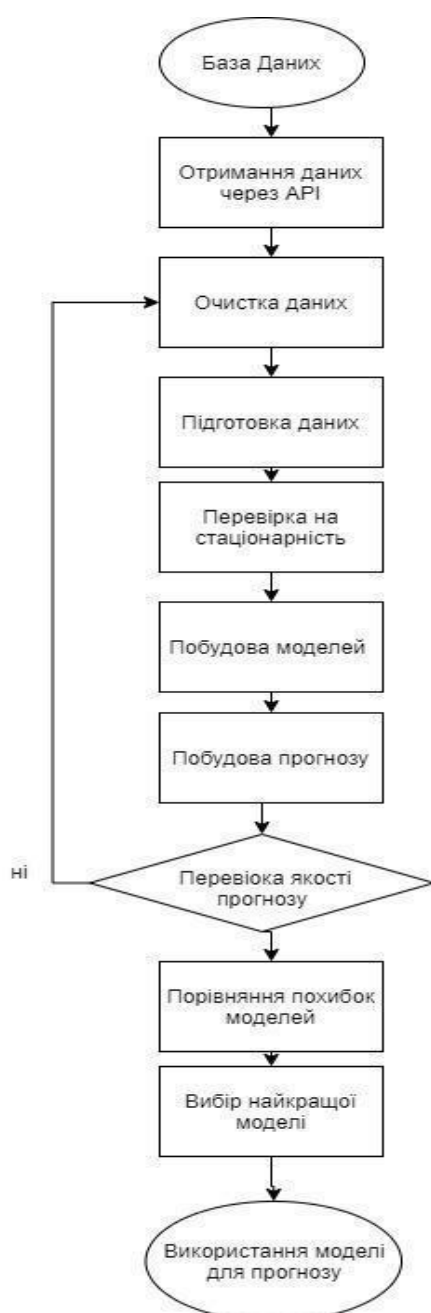


Рисунок 3.1 — Блок-схема алгоритм побудови прогнозу та вибору найкращої моделі

3.2 Підготовка даних для прогнозування

Одним з найважливіших етапів побудови якісного прогнозу є попередня підготовка даних для того, щоб модель отримала максимальну кількість інформації з даних. Для цього було перетворено часовий ряд до вигляду таблиці де таблиця опису змінних зображена в таблиці 3.1

Таблиця 3.1 — Опис змінних для набору даних

Назва змінної	Опис
Value	Значення реальне часового ряду
Y - 1	Ляг першого порядку
Y - 2	Ляг другого порядку
Y - K	Ляг K-того порядку
WINDOW_MEAN_1	Ковзне середнє 1-шого порядку
WINDOW_MEAN_K	Ковзне середнє K-того порядку
WINDOW_STD_1	Ковзна дисперсія 1-того порядку
WINDOW_STD_K	Ковзна дисперсія K-того порядку

Вхідний набір даних розбивається у відношенні 80:20 на навчальні дані та тестові дані. Кількість лагів обиралася за допомогою порядку АКФ. Ковзне середнє обиралося відповідно ЧАКФ, аналогічно ковна дисперсія. Кожен часовий ряд з ВВП був нормалізований, тобто кожне значення якого було поділено на 10^{10} для зменшення складності обчислення.

3.3 Дослідження та моделювання ВВП України

3.3.1 Описові характеристики процесу

Зобразимо як виглядає ВВП України на рис. 3.2

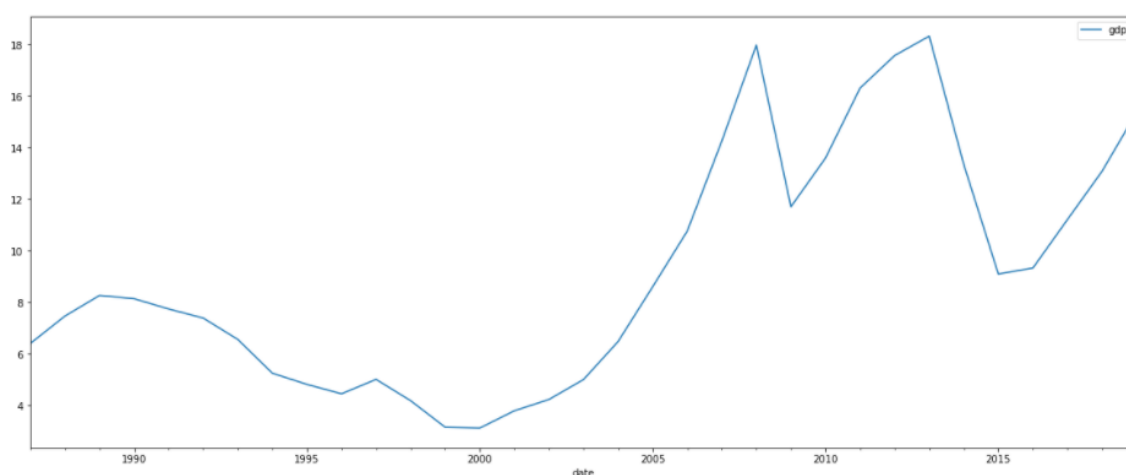


Рисунок 3.2 — ВВП України в період 1988-2019.

Використовуючи зображення графіка ВВП, очевидно що ряд є нестационарним, перевіривши стаціонарність з адопомогою тесту Дікі-Фуллера який підтвердив даний результат Розширений тест Дікі-Фуллера з $p\text{-value}=0.731$.

Зобразимо описові характеристики часового ряду в рисунку 3.3

	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	30.00000000	6.59765625	31.92187500	28.06250000	331.25000000	0.11840820	-1.69042969	3.32421875	0.18969727

Рисунок 3.3— Описові статистики для ВВП України в період 1988-2019.

Для кращого розуміння розподілу часового ряду необхідно поглянути на гістограму та графік щільності зображені відповідно на рис. 3.4 та рис. 3.5.

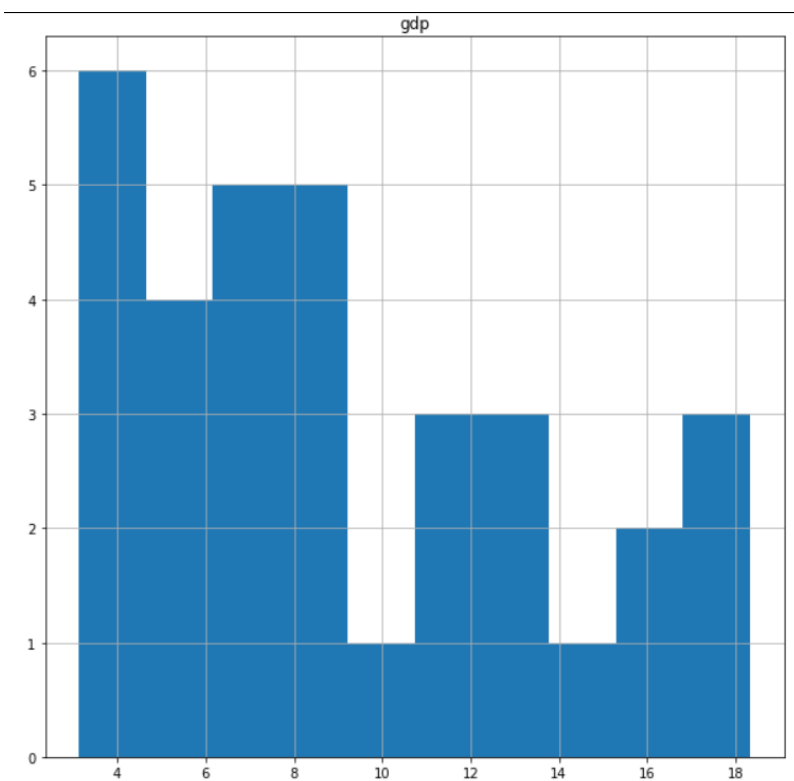


Рисунок 3.4 —Гістограма ВВП України в період 1988-2019.

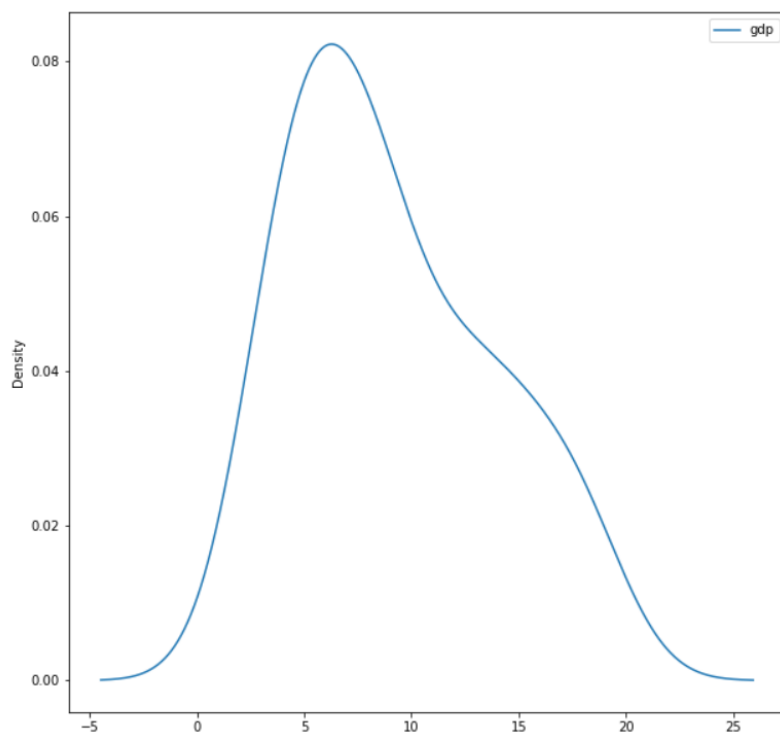


Рисунок 3.5 —Щільність ряду ВВП України в період 1988-2019.

Для визначення порядку лагу та ковзних статистик побудували АКФ та ЧАКФ відповідно рис. 3.6 та рис. 3.7.

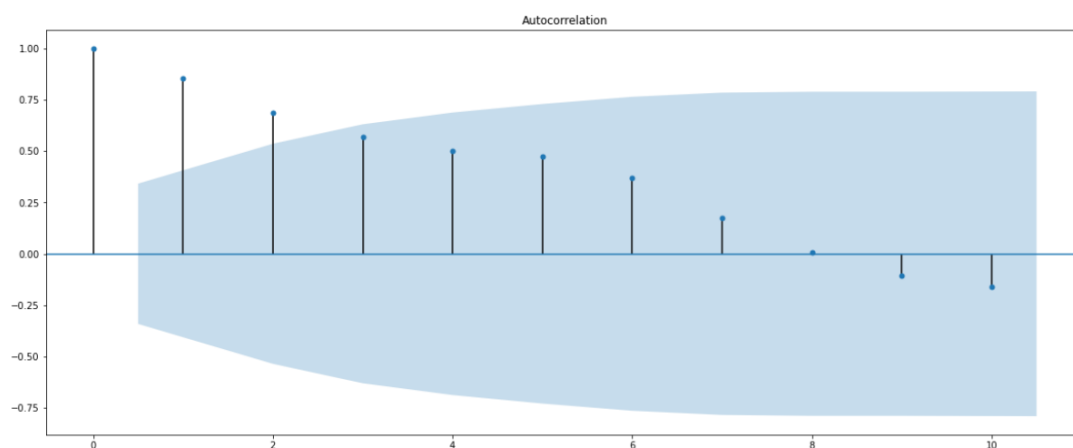


Рисунок 3.6 — АКФ для ряду ВВП України в період 1988-2019.

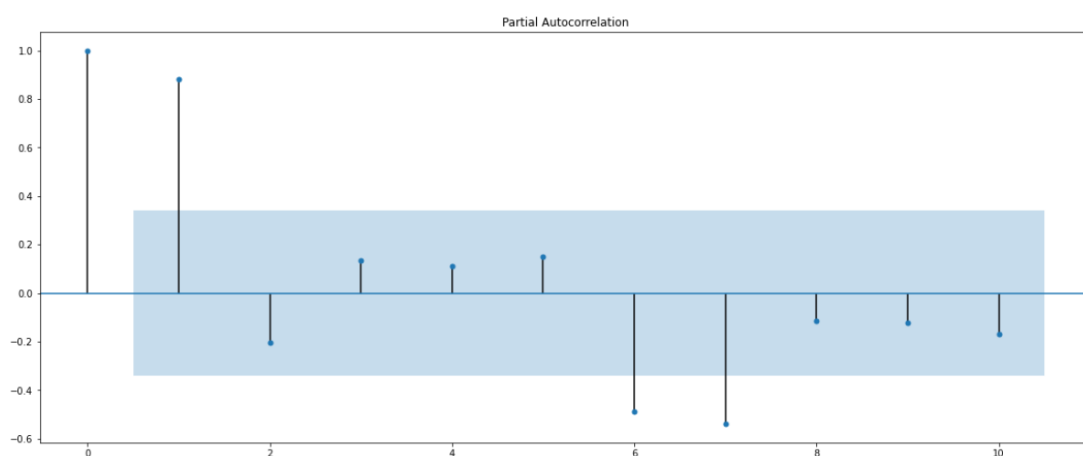


Рисунок 3.7 — ЧАКФ для ряду ВВП України в період 1988-2019.

Відповідно було обрано лаг порядку 2 та ковзні статистики порядку 2. Перейдемо до етапу моделювання моделей.

3.3.2. Моделювання дерева рішень

Підібрали параметри для моделі Дерева Рішень за допомогою жадібного алгоритму та кросс валідації тренувального дата-сету з кросвалідацією 5. Отримали відповідні оцінки зображені на рис. 3.8 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
DecisionTreeRegressor_train	0.980456	12.041186	8.029118	0.655776	2.179328	0.032688
DecisionTreeRegressor_test	0.188973	7.037999	10.202293	1.531666	2.669322	0.056672

Рисунок 3.8 —Статистика оцінки моделі для моделі Дерева Рішень

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.9.

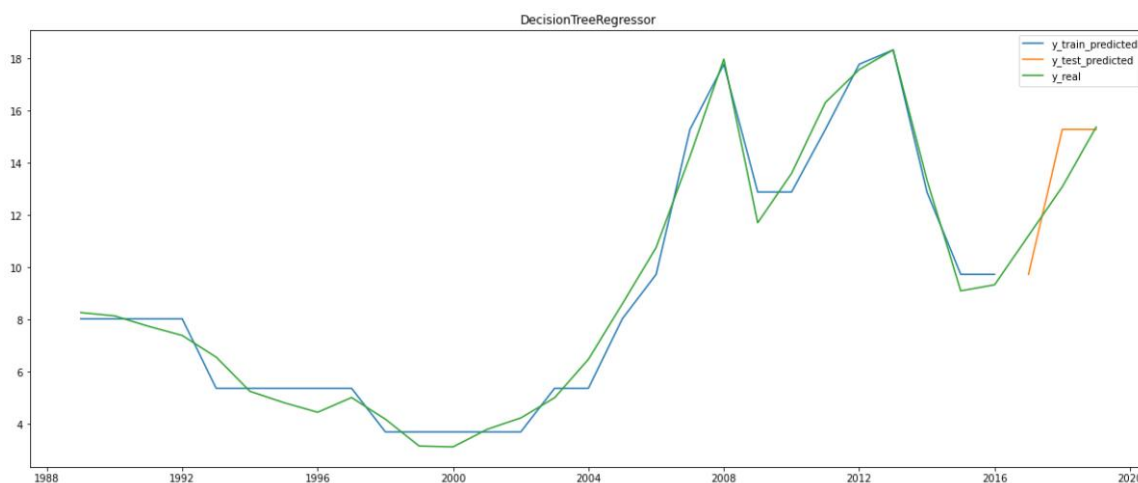


Рисунок 3.9 —Графік прогнозу моделі Дерева Рішень

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.10.

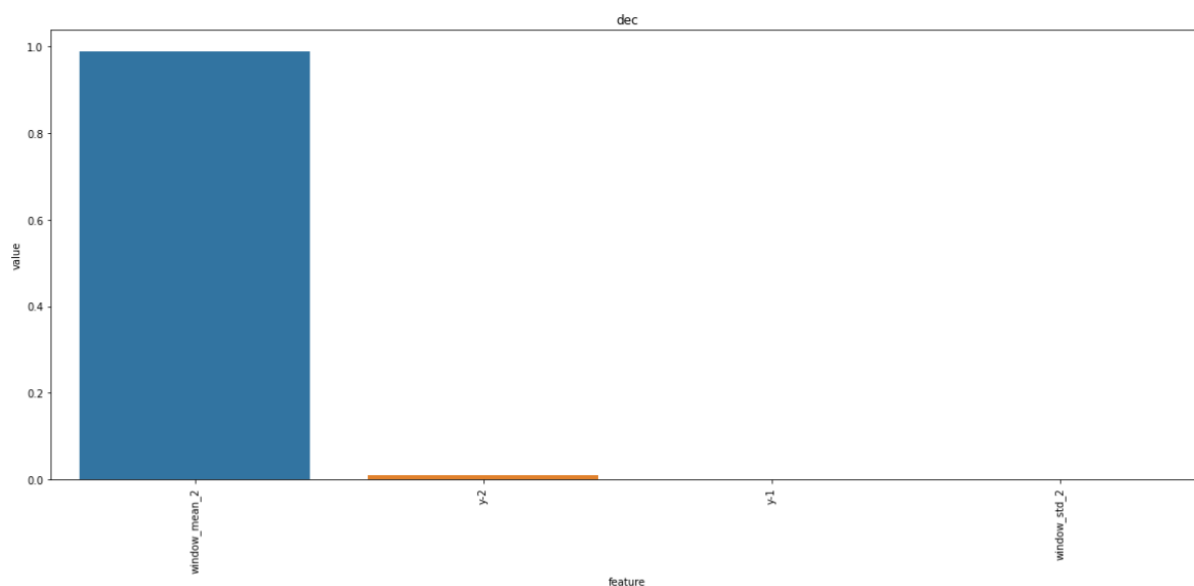


Рисунок 3.10— Стовпчикова діаграма для важливості регресорів для моделі Дерева Рішень

Як бачимо вище найбільш важливою змінною для прогнозування ВВП України для моделі Дерева Рішень це ковзне середнє порядку 2, а інші регресори майже не несуть інформативності даній моделі. Дана модель показала неадекватність свого прогнози на тестовій частині. Тому обрати її для подальшого застосування не можна.

3.3.3. Моделювання багатовимірної лінійної регресії

Знайшовши вагові значення для регресорів Отримали відповідні оцінки зображені на рис. 3.11 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
LinearRegression_train	1.00000000	0.00000000	0.00000000	0.00000000	0.56890375	0.00000000
LinearRegression_test	1.00000000	0.00000000	0.00000000	0.00000000	0.31551639	0.00000000

Рисунок 3.11— Статистика оцінки моделі для моделі лінійної регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.12.

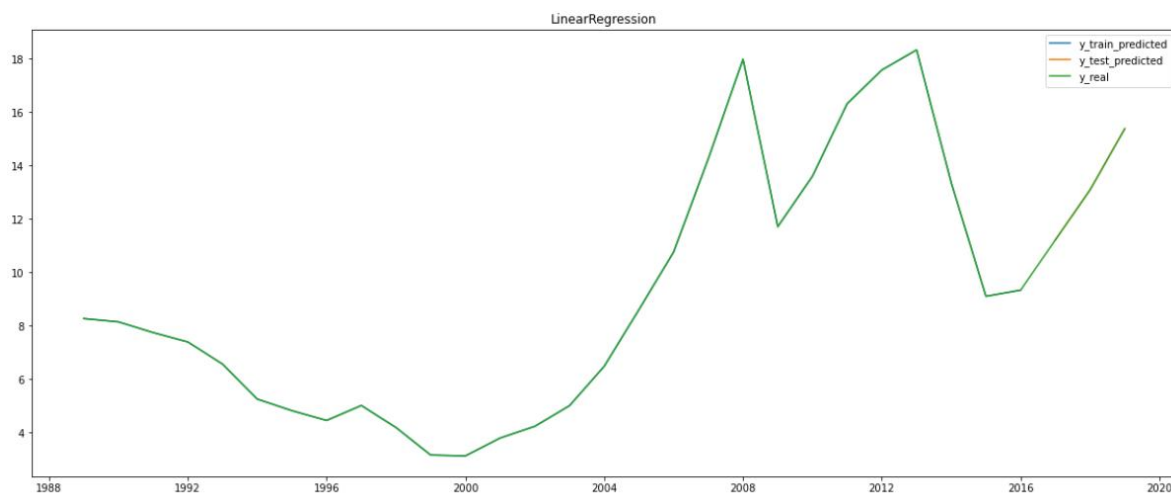


Рисунок 3.12 —Графік прогнозу моделі лінійної регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.13.

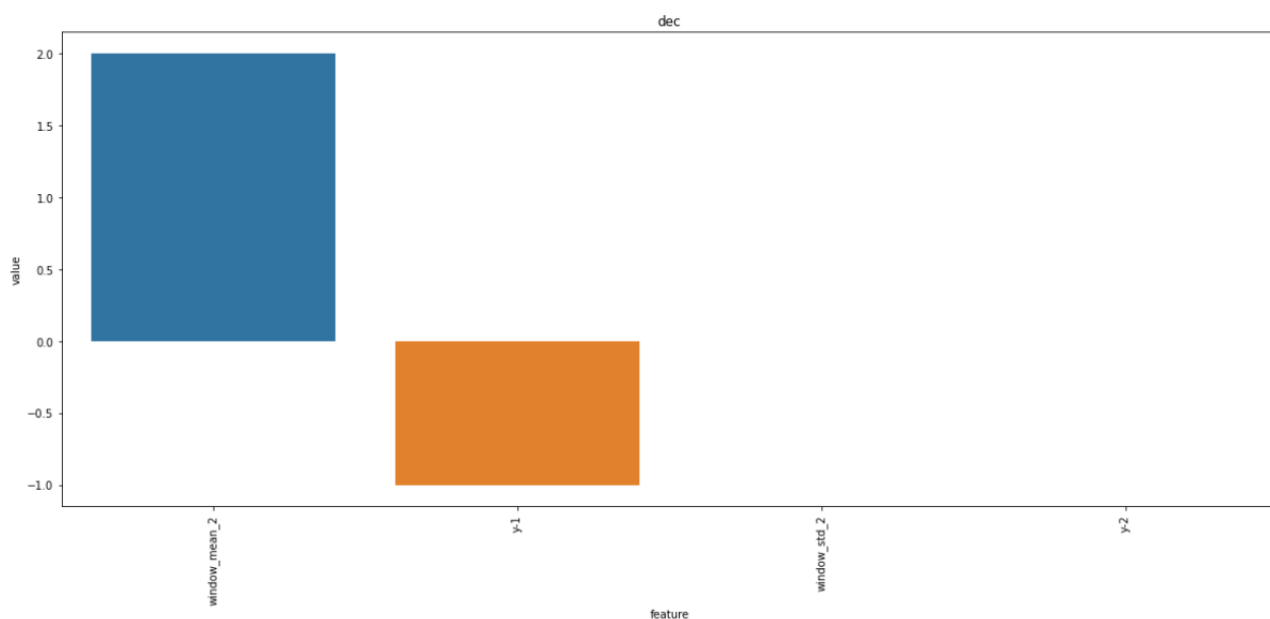


Рисунок 3.13 —Стовбчикова діаграма для важливості регресорів для моделі лінійної регресії

Як бачимо вище найбільш важливою змінною для прогнозування ВВП України для моделі лінійної регресії це ковзне середнє порядку 2 та лаг порядку 1, а інші регресори майже не несуть інформативності даним моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

3.3.4. Моделювання методу найближчих сусідів\

Підібравши за допомогою крос-валідації та жадібного алгоритму параметри. Отримали відповідні оцінки зображені на рис. 3.14 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
KNeighborsRegressor_train	0.905477	58.236289	12.534785	1.442175	2.014156	0.073199
KNeighborsRegressor_test	0.697647	2.623787	6.066802	0.935198	1.481656	0.035323

Рисунок 3.14 — Статистика оцінки моделі для моделі методу найближчих сусідів

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.15.

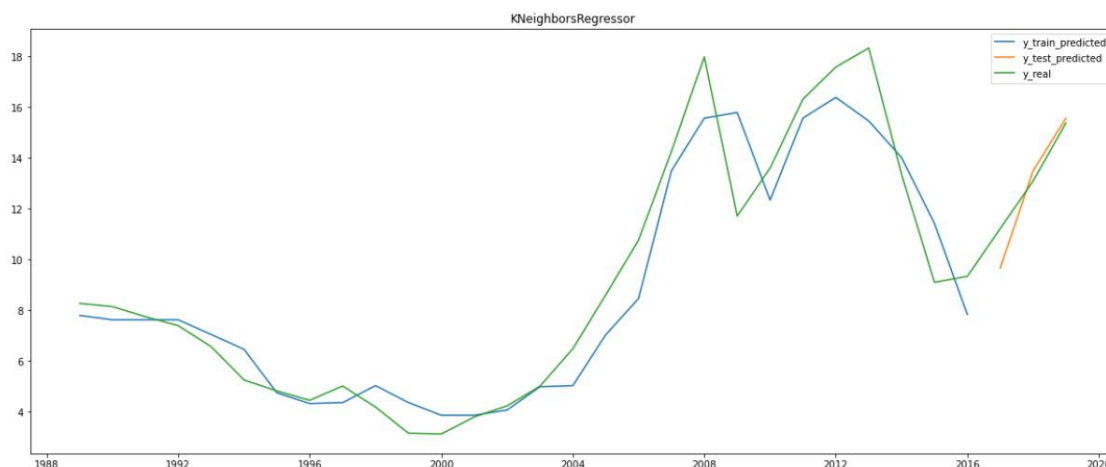


Рисунок 3.15 — Графік прогнозу моделі методу найближчих сусідів

Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування не можна.

3.3.5. Моделювання Баєсової регресії

Побудувавши модель отримали відповідні оцінки зображені на рис. 3.16 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
BayesianRidge_train	1.0	1.667951e-15	6.560700e-08	7.718140e-09	1.739015	3.843079e-10
BayesianRidge_test	1.0	1.260664e-16	4.901150e-08	6.482449e-09	0.130501	2.430070e-10

Рисунок 3.16 — Статистика оцінки моделі для моделі Баєсової регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.17.

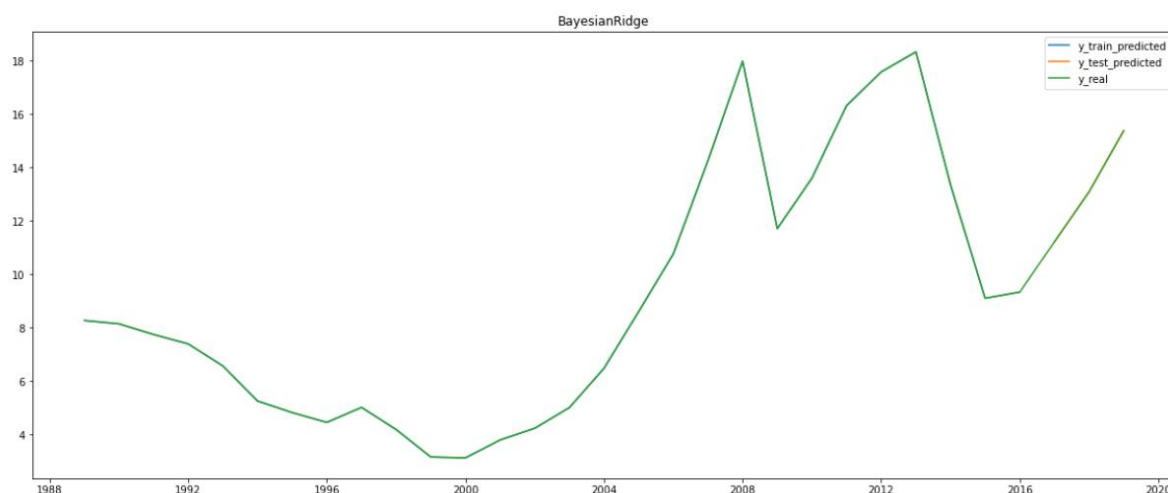


Рисунок 3.17— Графік прогнозу моделі Баєсової регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.18.

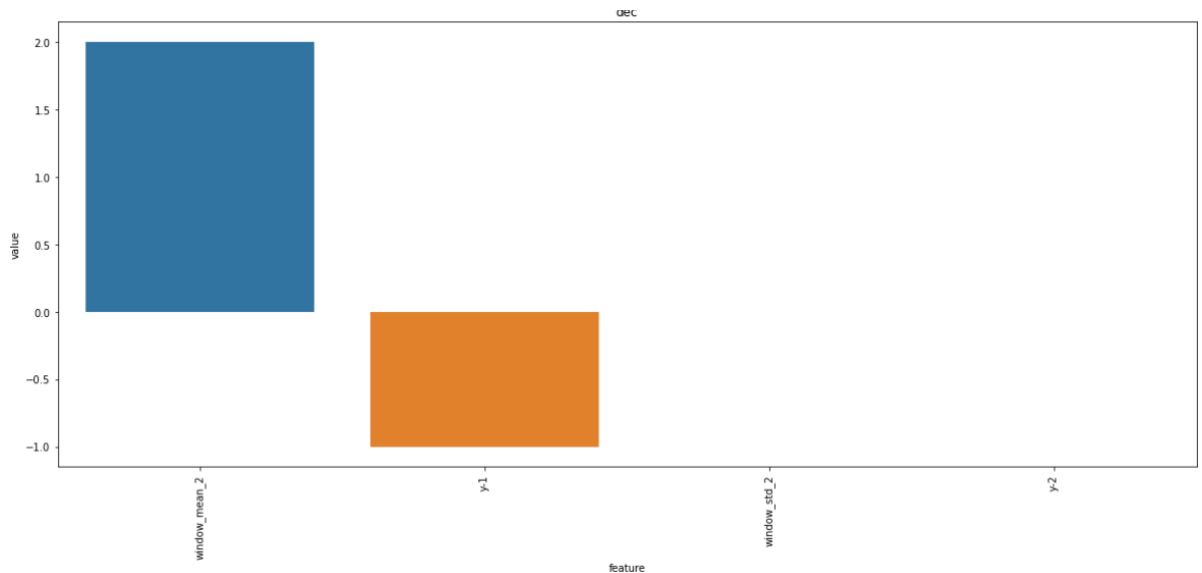


Рисунок 3.18 — Стовпчикова діаграма для важливості регресорів для моделі Баєсової регресії

Як бачимо вище найбільш важливою змінною для прогнозування ВВП України для моделі Баєсової регресії це ковзне середнє порядку 2 та лаг порядку 1, а інші регресори майже не несуть інформативності даній моделі. Дана модель показала адекватність свого прогнози на тестовій частині. Тому обрати її для подальшого застосування можна.

3.4 Дослідження та моделювання ВВП Польщі

3.4.1 Описові характеристики процесу

Зобразимо як виглядає ВВП Польщі на рис. 3.20.

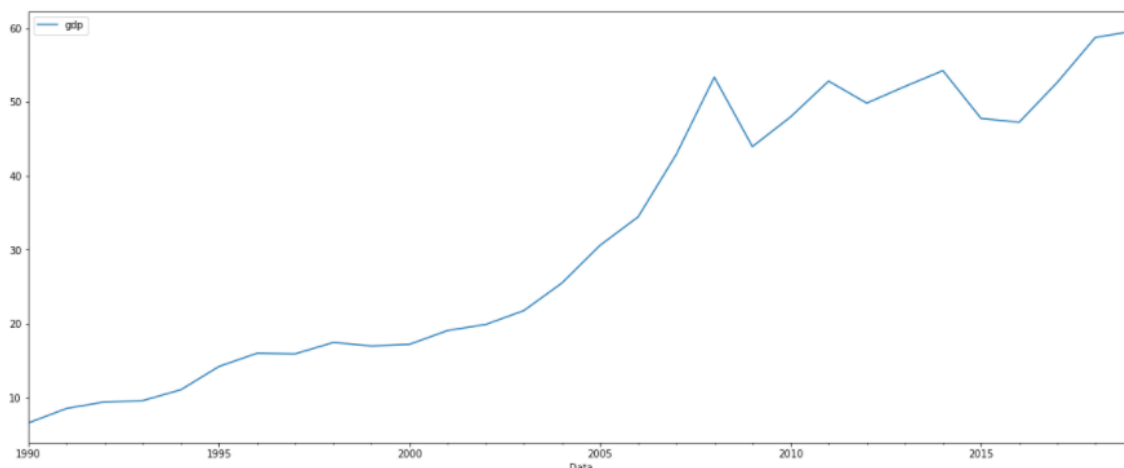


Рисунок 3.20 — ВВП Польщі в період 1990-2019.

Використовуючи зображення графіка ВВП, очевидно що ряд є нестационарним, перевіривши стаціонарність з адопомогою тесту Дікі-Фуллера який підтвердив даний результат Розширений тест Дікі-Фуллера з $p\text{-value}=0.91$.

Зообразимо описові характеристики часового ряду в рисунку 3.21.

	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	30.00000000	6.59765625	31.92187500	28.06250000	331.25000000	0.11840820	-1.69042969	3.32421875	0.18969727

Рисунок 3.21 — Описові статистики для ВВП Польщі в період 1990-2019.

Для кращого розуміння розподілу часового ряду необхідно поглянути на гістограму та графік щільності зображені відповіддно на рис. 3.22 та рис. 3.23.

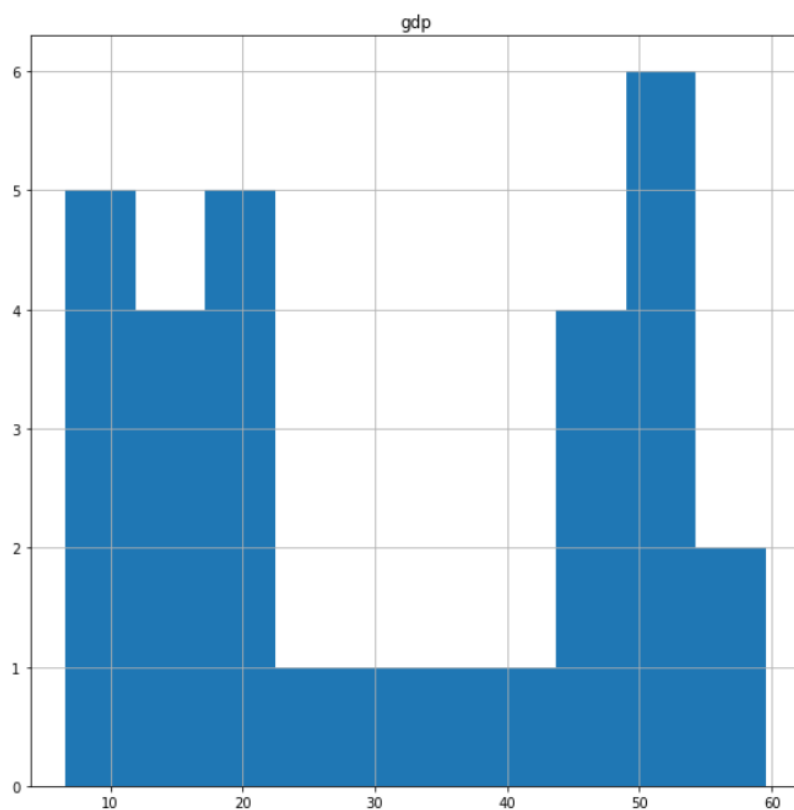


Рисунок 3.22 —Гістограма ВВП Польщі в період 1990-2019.

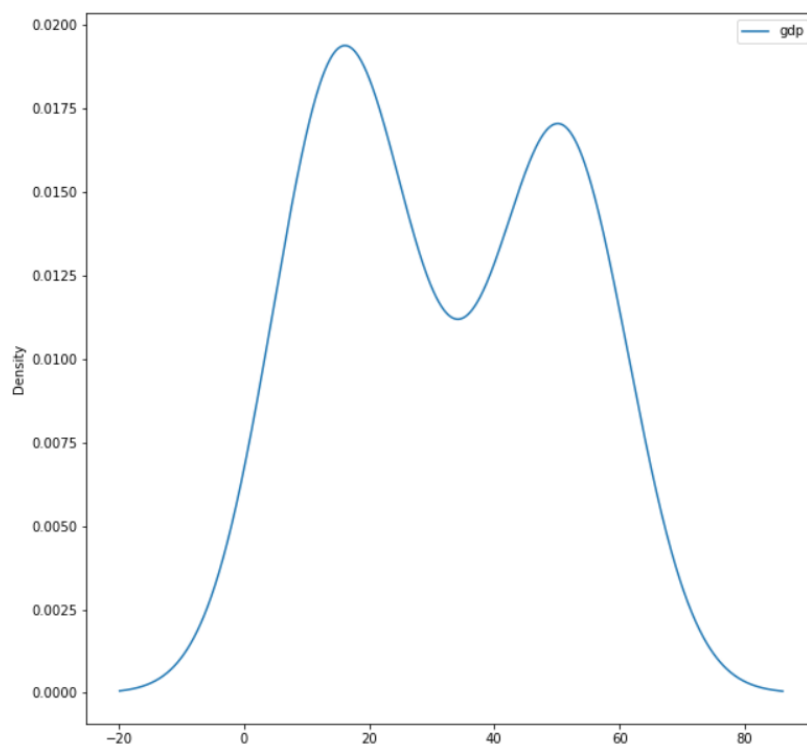


Рисунок 3.23— Щільність ряду ВВП Польщі в період 1990-2019.

Для визначення порядку лагу та ковзних статистик побудували АКФ та ЧАКФ відповідно рис. 3.24 та рис. 3.25.

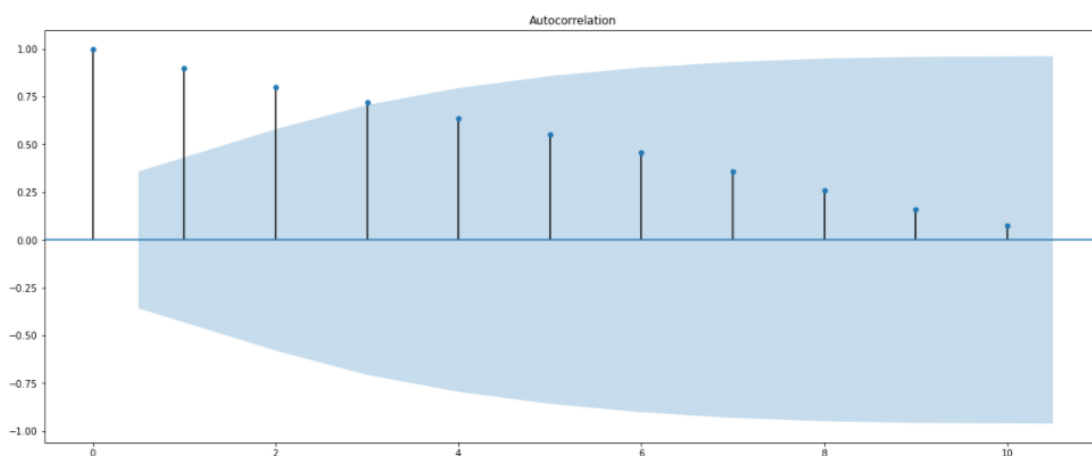


Рисунок 3.24 — АКФ для ряду ВВП Польщі в період 1990-2019.

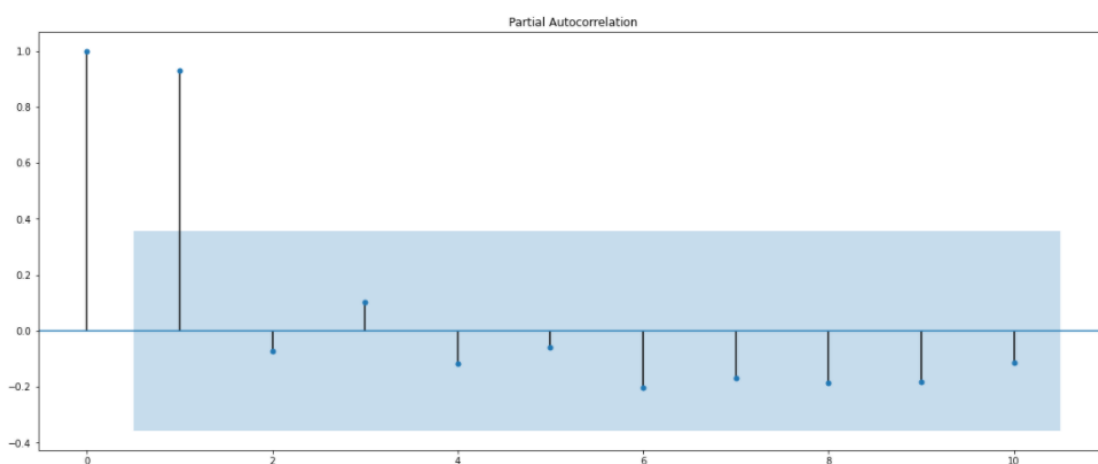


Рисунок 3.25 — ЧАКФ для ряду ВВП Польщі в період 1990-2019.

Відповідно було обрано лаг порядку 3 та ковзні статистики порядку 3. Перейдемо до етапу моделювання моделей.

3.4.2. Моделювання дерева рішень

Підібрали параметри для моделі Дерева Рішень за допомогою жадібного алгоритму та кросвалідації тренувального дата-сету з кросвалідацією 5. Отримали відповідні оцінки зображені на рис. 3.26 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
DecisionTreeRegressor_train	0.98467331	91.90670008	5.43217798	1.95689699	2.24260626	0.02760187
DecisionTreeRegressor_test	-6.39462339	211.72856462	12.12190704	8.40096353	0.54682912	0.07809657

Рисунок 3.26 —Статистика оцінки моделі для моделі Дерева Рішень

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.27.

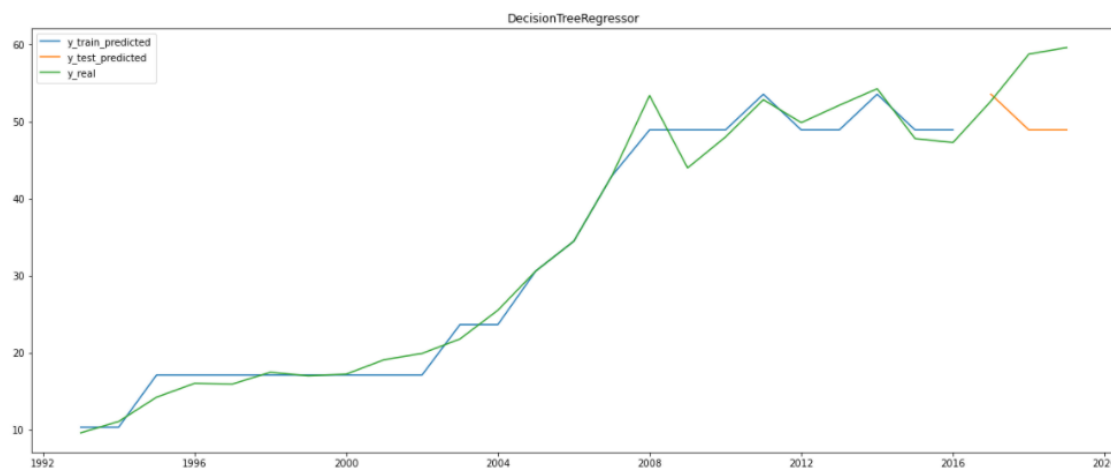


Рисунок 3.27 —Графік прогнозу моделі Дерева Рішень

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.28.

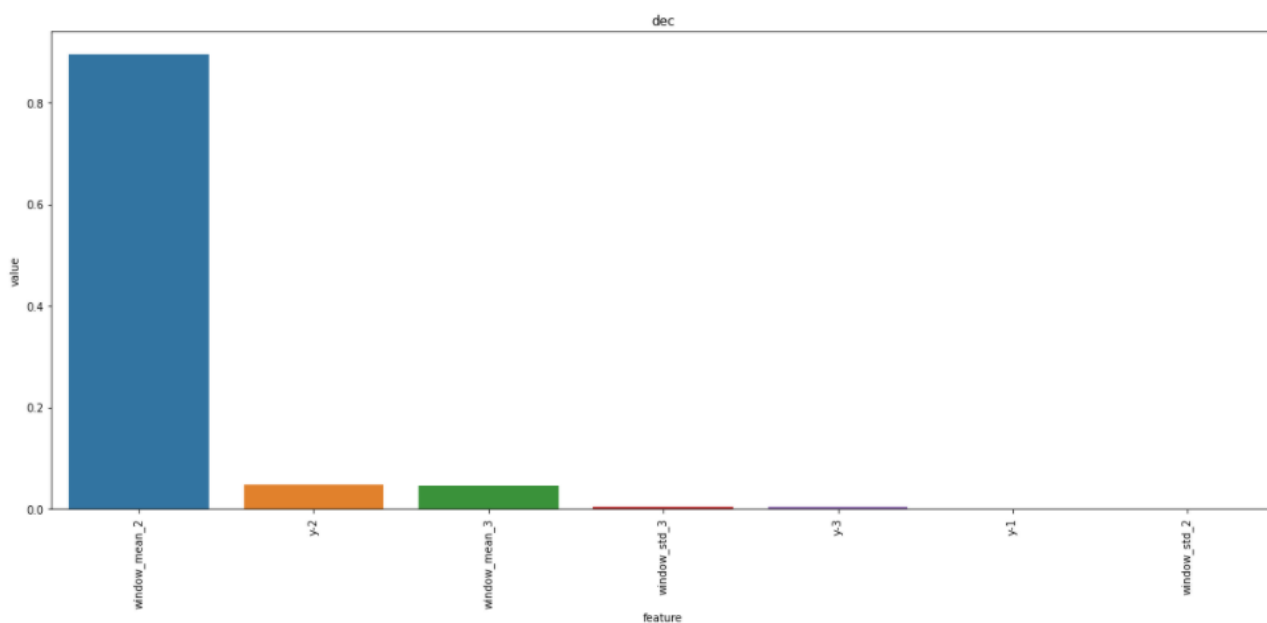


Рисунок 3.28 —Стовпчикова діаграма для важливості регресорів для моделі Дерева Рішень

Як бачимо вище найбільш важливою змінною для прогнозування ВВП Польщі для моделі Дерева Рішень це ковзне середнє порядку 2 яке має найбільше значення далі це лаг порядку 2 та ковзне середнє порядку 3, а інші регресори майже не несуть інформативності даній моделі. Дана модель показала не адекватність свого прогнози на текстовій частині. Тому обрати її для подальшого застосування не можна.

3.4.3. Моделювання багатовимірної лінійної регресії

Знайшовши вагові значення для регресорів Отримали відповідні оцінки зображені на рис. 3.29 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
LinearRegression_train	1.00000000	0.00000000	0.00000000	0.00000000	0.56890375	0.00000000
LinearRegression_test	1.00000000	0.00000000	0.00000000	0.00000000	0.31551639	0.00000000

Рисунок 3.29 — Статистика оцінки моделі для моделі лінійної регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.30.

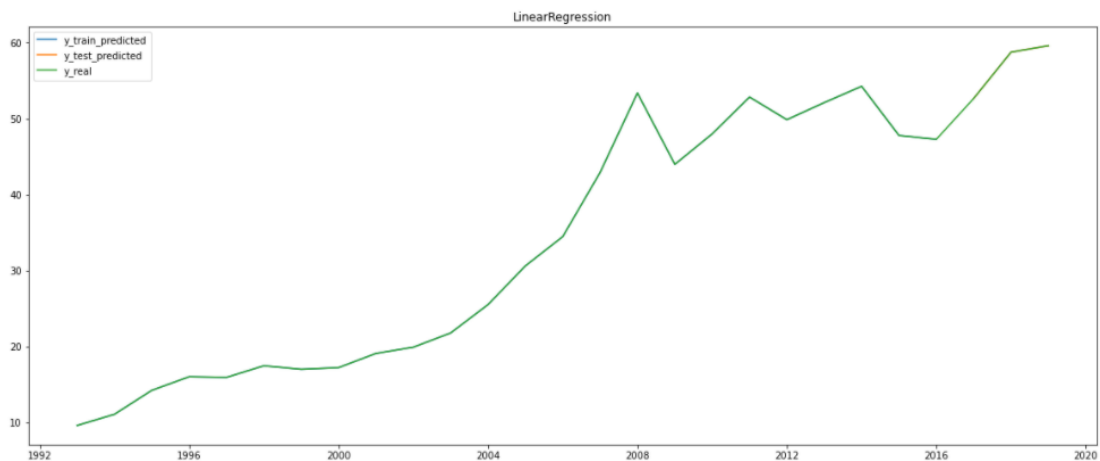


Рисунок 3.30 — Графік прогнозу моделі лінійної регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.31.

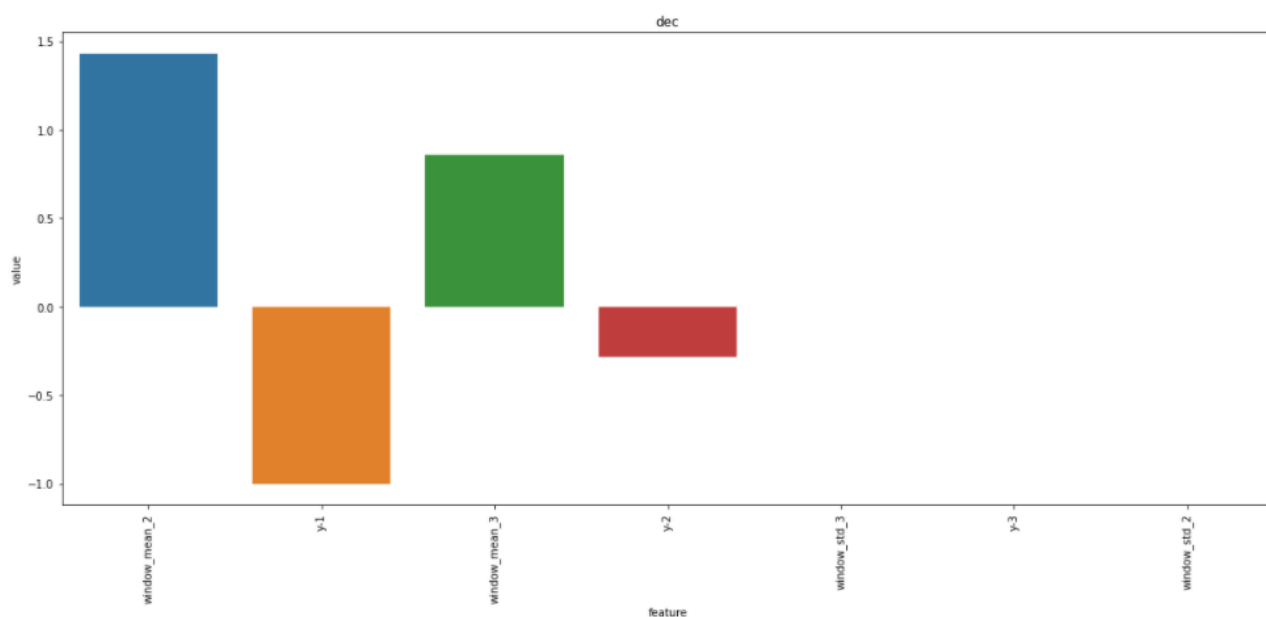


Рисунок 3.31 — Стовбчикова діаграма для важливості регресорів для моделі лінійної регресії

Як бачимо вище найбільш важливою змінною для прогнозування ВВП Польщі для моделі лінійної регресії це ковзне середнє порядку 2 та 3, лаг порядку 1 та 2, а інші регресори майже не несуть інформативності даній моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

3.4.4. Моделювання методу найближчих сусідів

Підібравши за допомогою крос-валідації та жадібного алгоритму параметри. Отримали відповідні оцінки зображені на рис. 3.32 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
KNeighborsRegressor_train	1.00000000	0.00000000	0.00000000	0.00000000	nan	0.00000000
KNeighborsRegressor_test	-3.17807557	119.62988476	9.07353818	6.31479440	0.49090345	0.05782379

Рисунок 3.32— Статистика оцінки моделі для моделі методу найближчих сусідів

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.33.

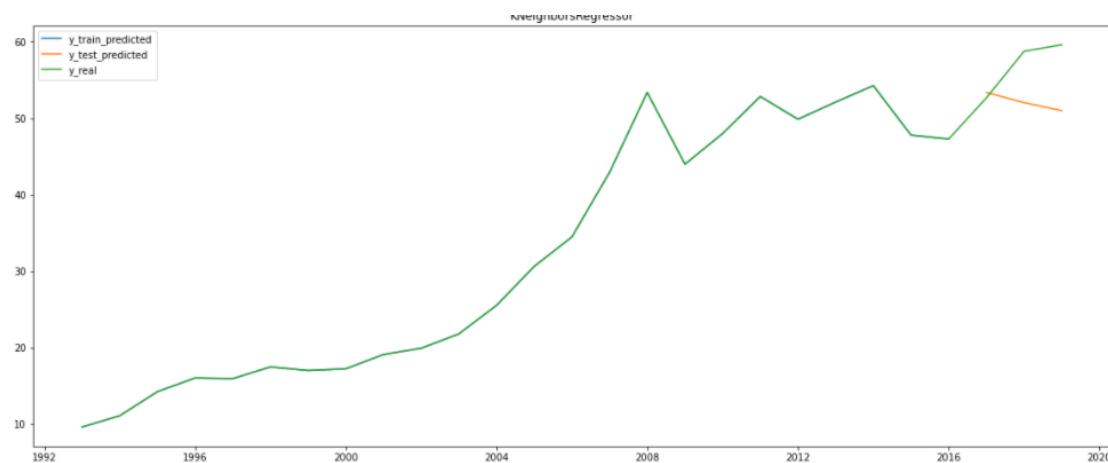


Рисунок 3.33— Графік прогнозу моделі методу найближчих сусідів

Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування не можна.

3.4.5. Моделювання Баєсової регресії

Побудувавши модель отримали відповідні оцінки зображені на рис. 3.34 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index	ex_var
BayesianRidge_train	1.00000000	0.00000000	0.00000639	0.00000278	1.32664848	0.00000004	1.00000000
BayesianRidge_test	1.00000000	0.00000000	0.00000392	0.00000287	0.71595828	0.00000003	1.00000000

Рисунок 3.34 — Статистика оцінки моделі для моделі Баєсової регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.35.

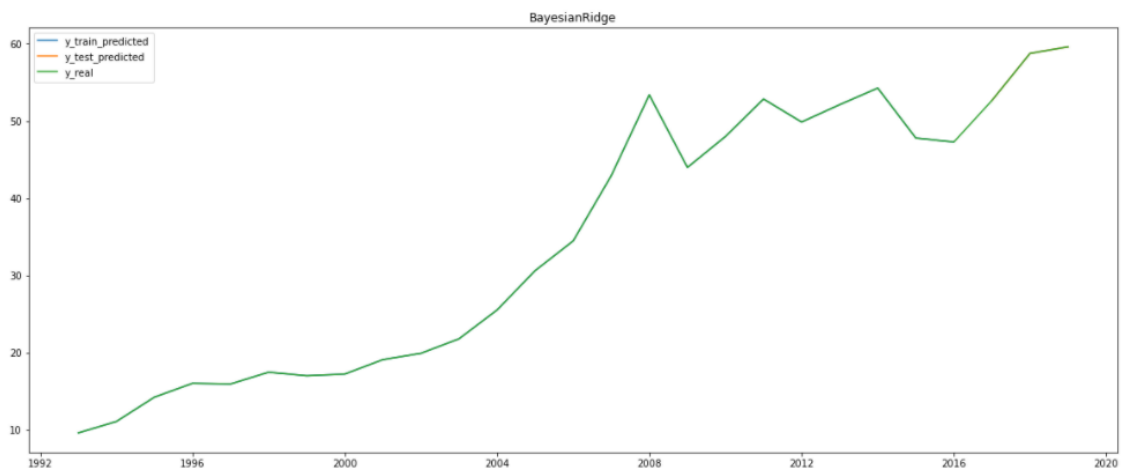


Рисунок 3.35 —Графік прогнозу моделі Баєсової регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.36.

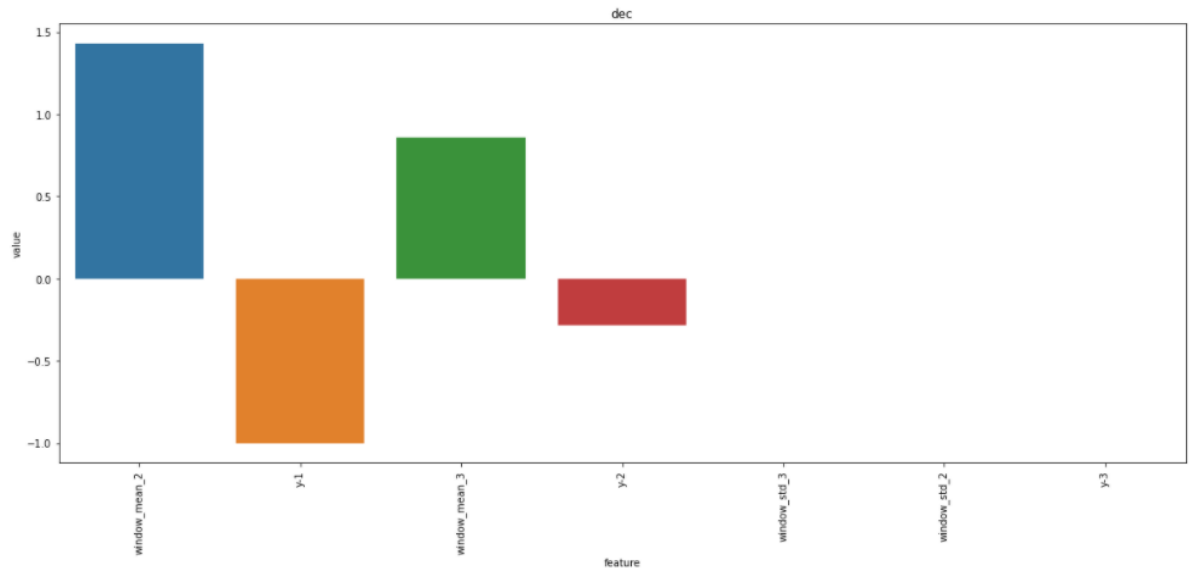


Рисунок 3.36 — Стовпчикова діаграма для важливості регресорів для моделі Баєсової регресії

Як бачимо вище найбільш важливою змінною для прогнозування ВВП Польщі для моделі Баєсової регресії це ковзне середнє порядку 2 та лаг порядку 1, а інші регресори майже не несуть інформативності даним моделі. Дана модель показала адекватність свого прогнози на тестовій частині. Тому обрати її для подальшого застосування можна.

3.5 Дослідження та моделювання курсу акцій Facebook

3.5.1 Описові характеристики процесу

Зобразимо як виглядає курс акцій Facebook на рис. 3.37.

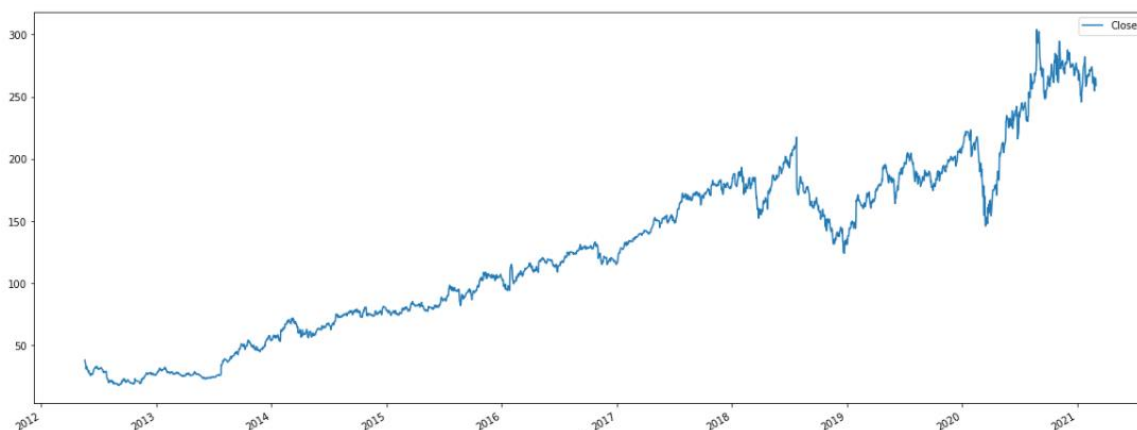


Рисунок 3.37 — Курс акцій Facebook в період 2012-2021.

Використовуючи зображення графіка курсу акцій Netflix, очевидно що ряд є нестационарним, перевіривши стаціонарність з адопомогою тесту Дікі-Фуллера який підтвердив даний результат розширений тест Дікі-Фуллера з $p\text{-value}=0.879$.

Зобразимо описові характеристики часового ряду в рисунку 3.38.

	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	2210.00000000	26.84375000	126.68750000	123.93750000	4912.00000000	0.27124023	-0.75097656	79.12500000	0.00000000

Рисунок 3.38 — Описові статистики для курсу акцій Facebook в період 2012-2021.

Для кращого розуміння розподілу часового ряду необхідно поглянути на гістограму та графік щільності зображені відповідно на рис. 3.39 та рис. 3.40.

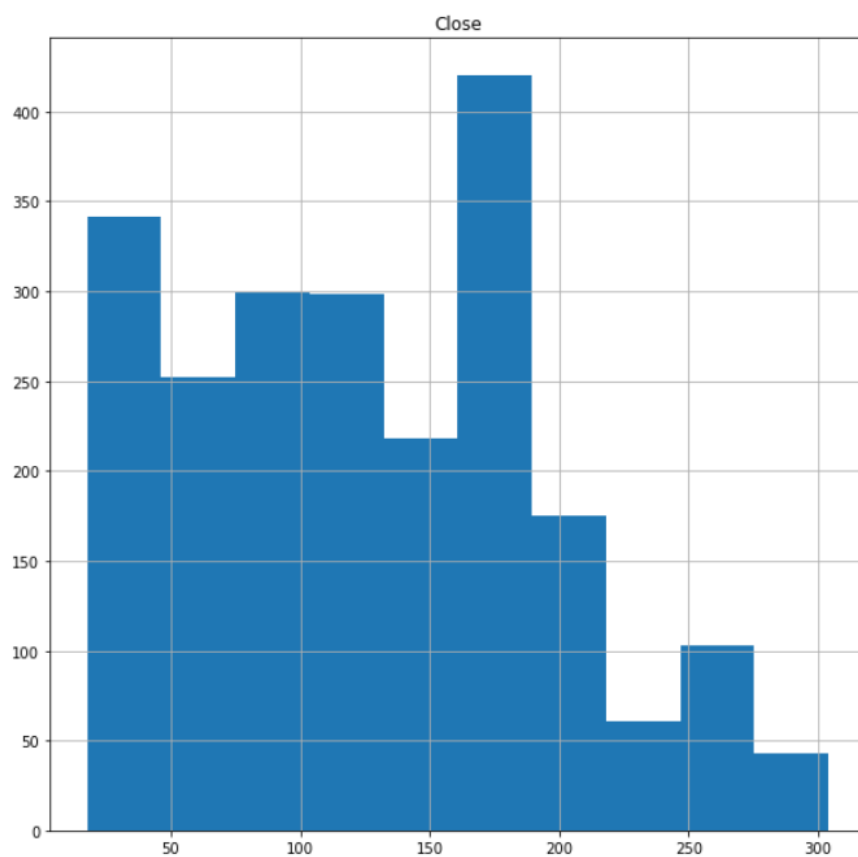


Рисунок 3.39 — Гістограма Курс акцій Facebook в період 2012-2021.

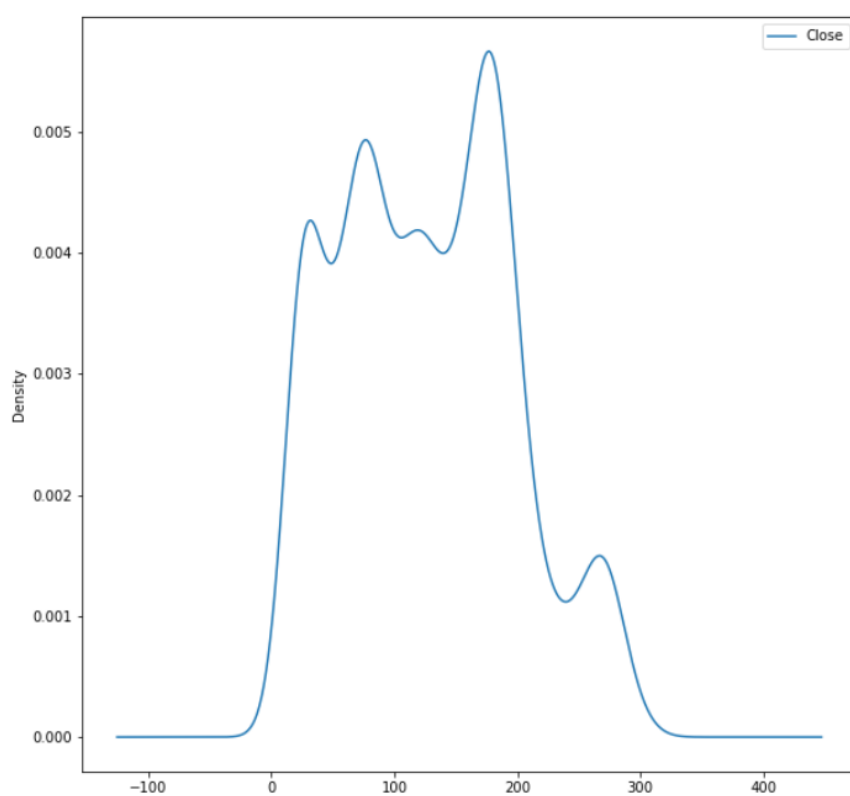


Рисунок 3.40 — Щільність ряду курс акцій Facebook в період 2012-2021.

Для визначення порядку лагу та ковзних статистик побудували АКФ та ЧАКФ відповідно рис. 3.41 та рис. 3.42.

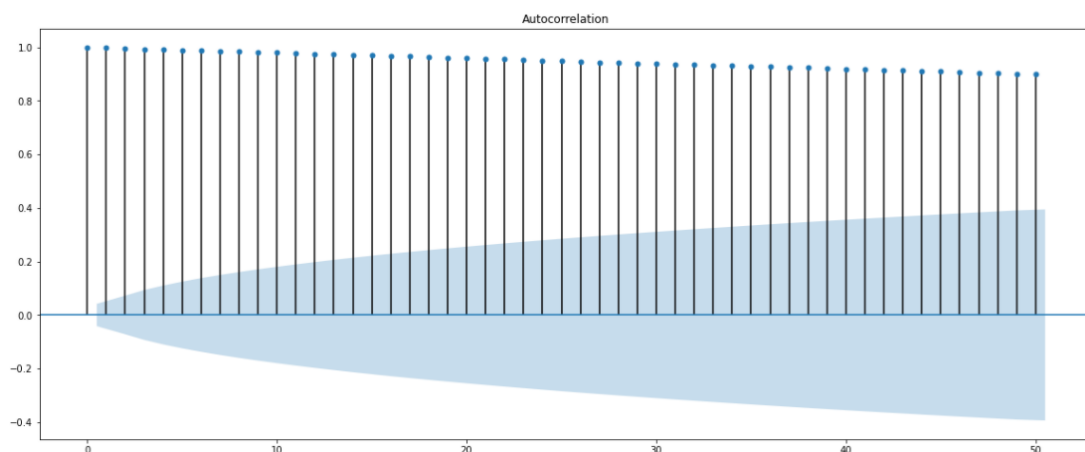


Рисунок 3.41 — АКФ для ряду курс акцій Facebook в період 2012-2021.

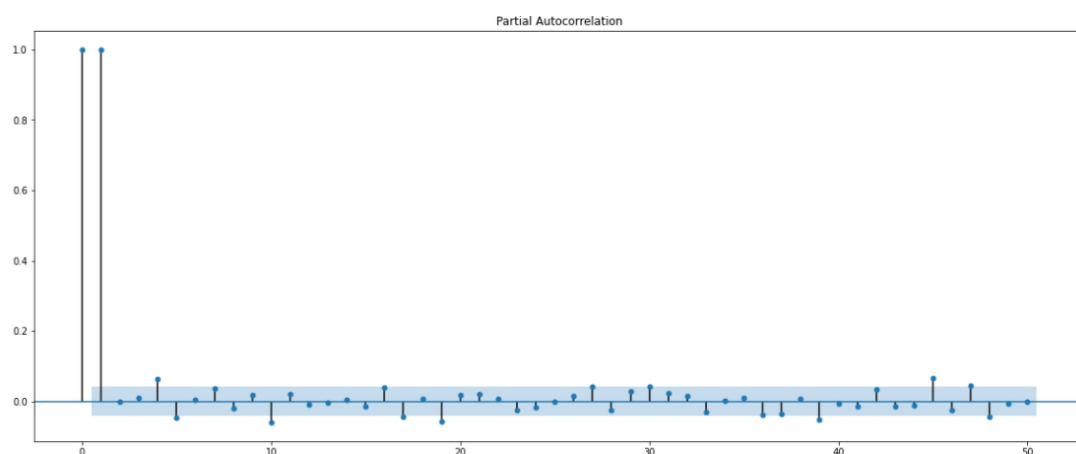


Рисунок 3.42 — ЧАКФ для ряду курс акцій Facebook в період 2012-2021.

Відповідно було обрано лаг порядку 10 та ковзні статистики порядку 4. Перейдемо до етапу моделювання моделей.

3.5.2. Моделювання дерева рішень

Підібрали параметри для моделі Дерева Рішень за допомогою жадібного алгоритму та кросвалідації тренувального дата-сету з кросвалідацією 5. Отримали відповідні оцінки зображені на рис. 3.41 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
DecisionTreeRegressor_train	0.99883598	7071.25227865	1.81102705	1.92917385	1.46768325	0.00781944
DecisionTreeRegressor_test	-0.67659124	699284.79345549	14.93501393	48.27990588	0.01546698	0.10951891

Рисунок 3.41 — Статистика оцінки моделі для моделі Дерева Рішень

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.42.

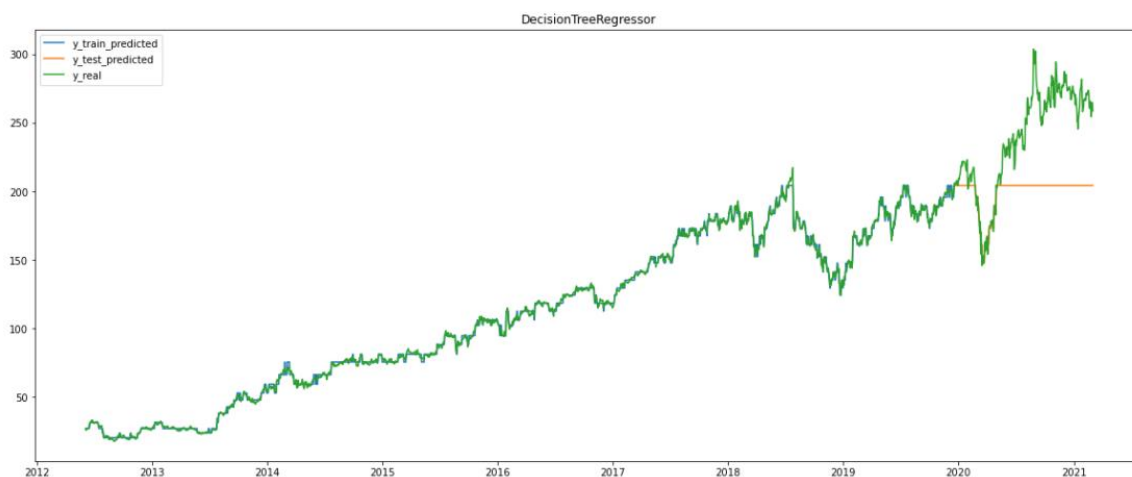


Рисунок 3.42 — Графік прогнозу моделі Дерева Рішень

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.43.

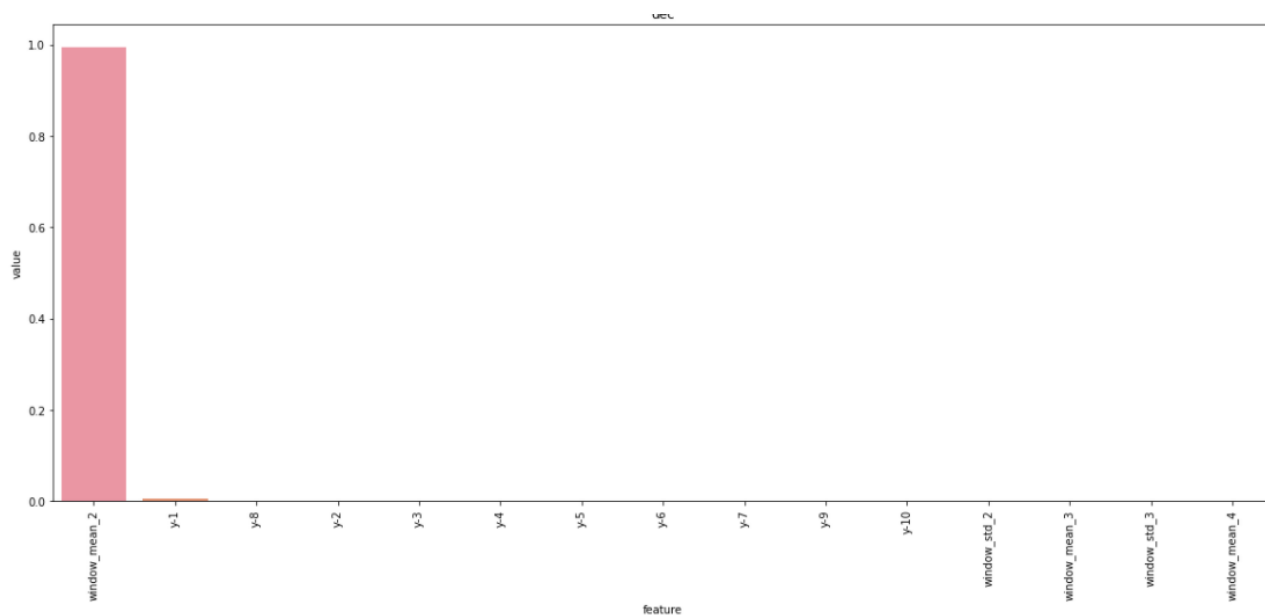


Рисунок 3.43 — Стовпчикова діаграма для важливості регресорів для моделі Дерева Рішень

Як бачимо вище найбільш важливою змінною для прогнозування Курс акцій Facebook для моделі Дерева Рішень це ковзне середнє порядку 2 яке має найбільше значення, а інші регресори майже не несуть інформативності даних моделі. Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування неможна.

3.5.3. Моделювання багатовимірної лінійної регресії

Знайшовши вагові значення для регресорів отримали відповідні оцінки зображені на рис. 3.44 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

	name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
0	LinearRegression_train	1.00000000	0.00000000	0.00000000	0.00000000	0.00138060	0.00000000
0	LinearRegression_test	1.00000000	0.00000000	0.00000000	0.00000000	0.00117067	0.00000000

Рисунок 3.44 — Статистика оцінки моделі для моделі лінійної регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.45.

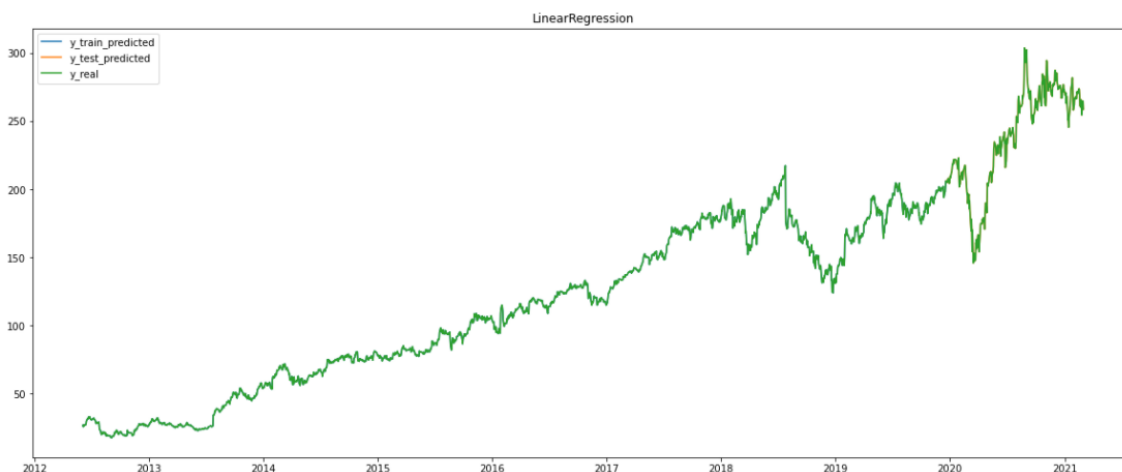


Рисунок 3.45 — Графік прогнозу моделі лінійної регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.46.

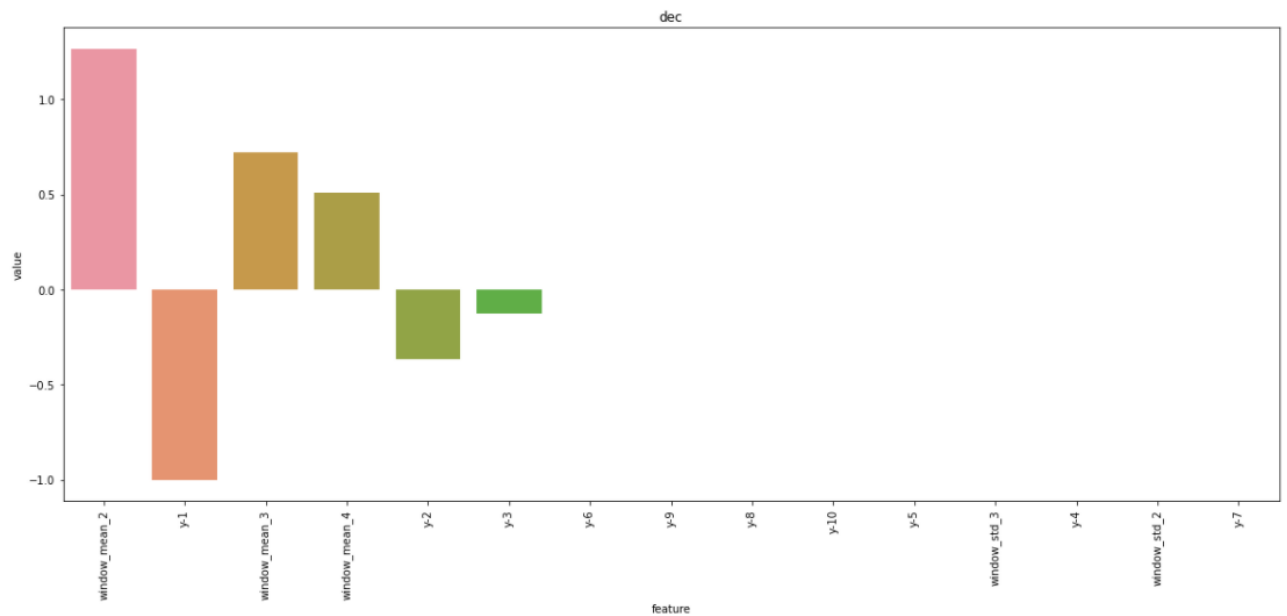


Рисунок 3.46 — Стовбчикова діаграма для важливості регресорів для моделі лінійної регресії

Як бачимо вище найбільш важливою змінною для прогнозування Курс акцій Facebook для моделі лінійної регресії це ковзне середнє порядку 2, 3 та 4, лаг порядку 1, 2,3. Інші регресори майже не несуть інформативності даних моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

3.5.4. Моделювання методу найближчих сусідів

Підібравши за допомогою крос-валідації та жадібного алгоритму параметри. Отримали відповідні оцінки зображені на рис. 3.47 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
KNeighborsRegressor_train	1.00000000	0.00000000	0.00000007	0.00000077	2.04342649	0.00000000
KNeighborsRegressor_test	-0.25510920	523490.01845476	12.56062226	41.77279890	0.02244335	0.09319514

Рисунок 3.47 — Статистика оцінки моделі для моделі методу найближчих сусідів

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.48.



Рисунок 3.48 — Графік прогнозу моделі методу найближчих сусідів

Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування не можна.

3.5.5. Моделювання Баєсової регресії

Побудувавши модель отримали відповідні оцінки зображені на рис. 3.49 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
BayesianRidge_train	0.99999987	0.81343660	0.03122976	0.02069117	0.00053057	0.00008386
BayesianRidge_test	0.99999828	0.71653561	0.01925711	0.04887179	0.00060406	0.00010143

Рисунок 3.49 — Статистика оцінки моделі для моделі Баєсової регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.50.



Рисунок 3.50 — Графік прогнозу моделі Баєсової регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.51.

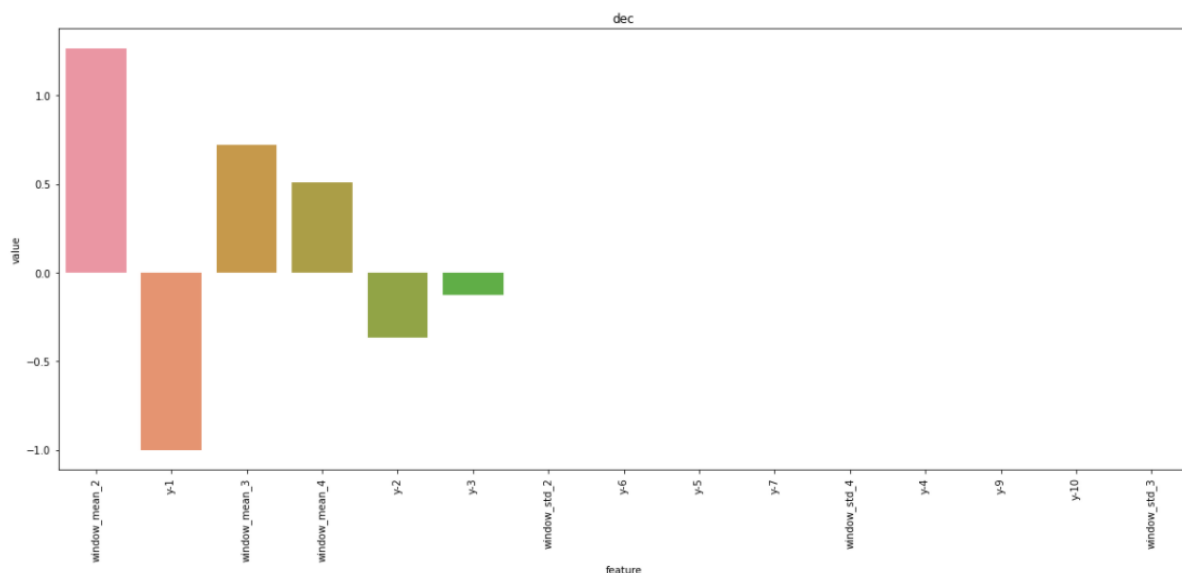


Рисунок 3.51 — Стовбчикова діаграма для важливості регресорів для моделі Баєсової регресії

Як бачимо вище найбільш важливою змінною для прогнозування курсу акцій Facebook для моделі Баєсової регресії це ковзне середнє порядку 2,3,4 та лаги порядку 1,2,3, а інші регресори майже не несуть інформативності даних моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

3.5 Дослідження та моделювання курсу акцій Netflix

3.5.1 Описові характеристики процесу

Зобразимо як виглядає курс акцій Netflix на рис. 3.52.

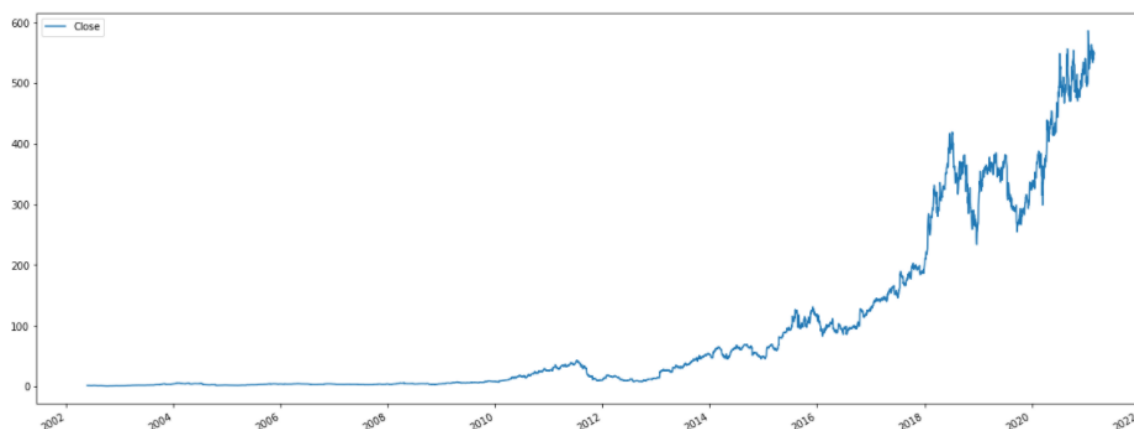


Рисунок 3.52 — Курс акцій Netflix в період 2002-2021.

Використовуючи зображення графіка курсу акцій Netflix, очевидно що ряд є нестационарним, перевіривши стаціонарність з допомогою тесту Дікі-Фуллера який підтвердив даний результат розширений тест Дікі-Фуллера з $p\text{-value}=0.999$.

Зобразимо описові характеристики часового ряду в рисунку 3.53.

observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
4728.00000000	1.60742188	91.43750000	16.98437500	19248.00000000	1.70605469	1.78710938	2916.00000000	0.00000000

Рисунок 3.53 — Описові статистики для курсу акцій Netflix в період 2002-2021.

Для кращого розуміння розподілу часового ряду необхідно поглянути на гістограму та графік щільності зображені відповідно на рис. 3.54 та рис. 3.55.

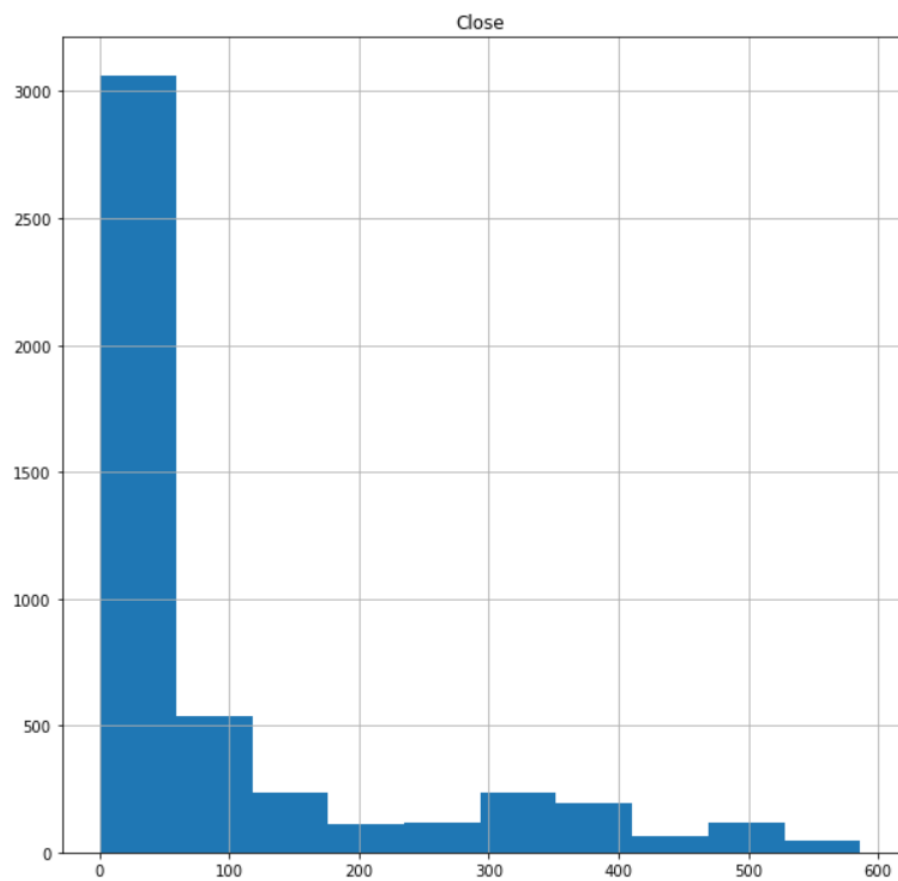


Рисунок 3.54 — Гістограма Курс акцій Netflix в період 2002-2021.

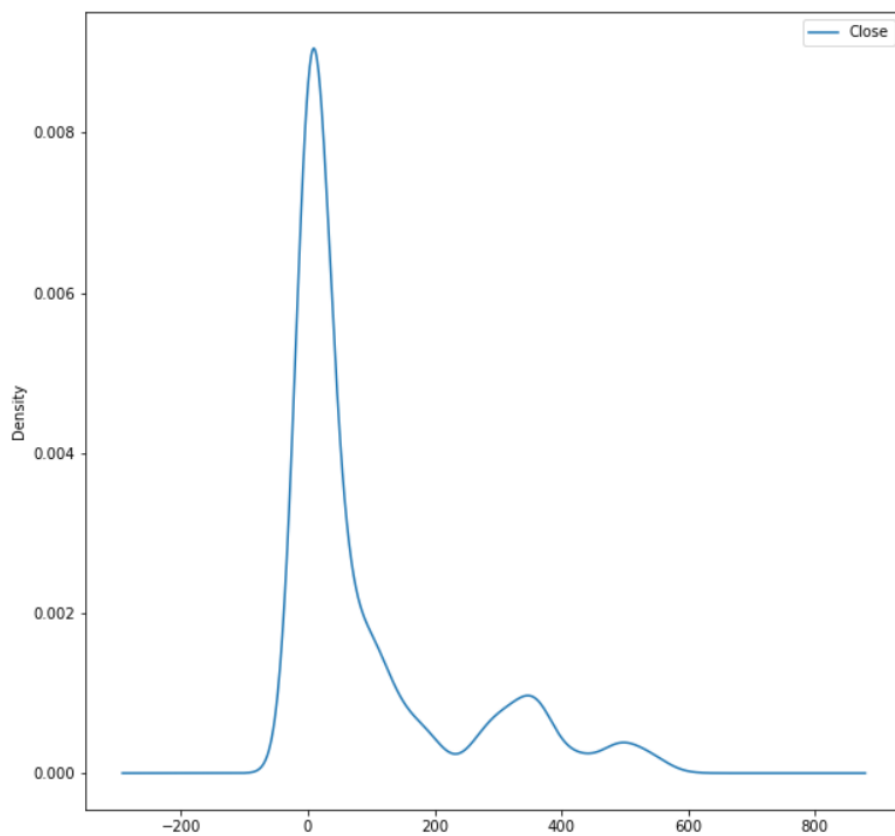


Рисунок 3.55 — Щільність ряду курс акцій Netflix в період 2002-2021.

Для визначення порядку лагу та ковзних статистик побудували АКФ та ЧАКФ відповідно рис. 3.56 та рис. 3.57.

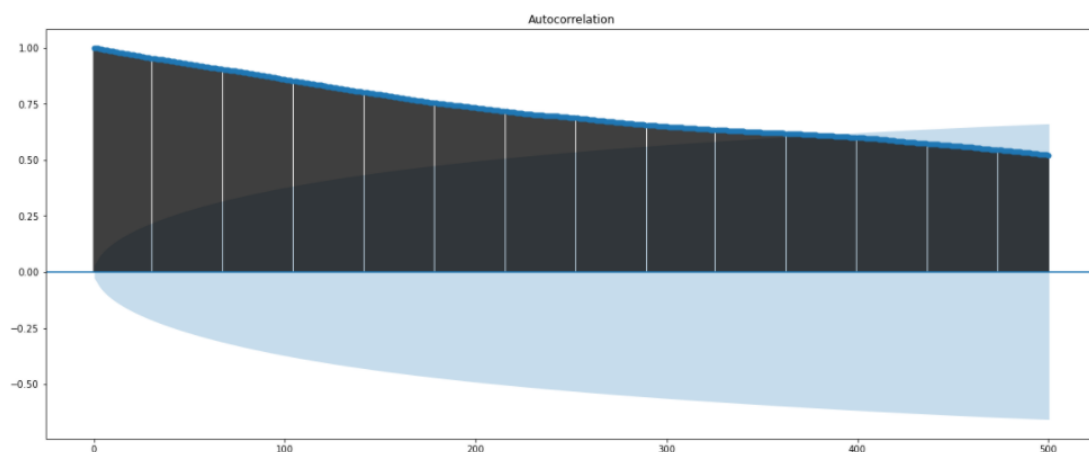


Рисунок 3.56 — АКФ для ряду курс акцій Netflix в період 2002-2021.

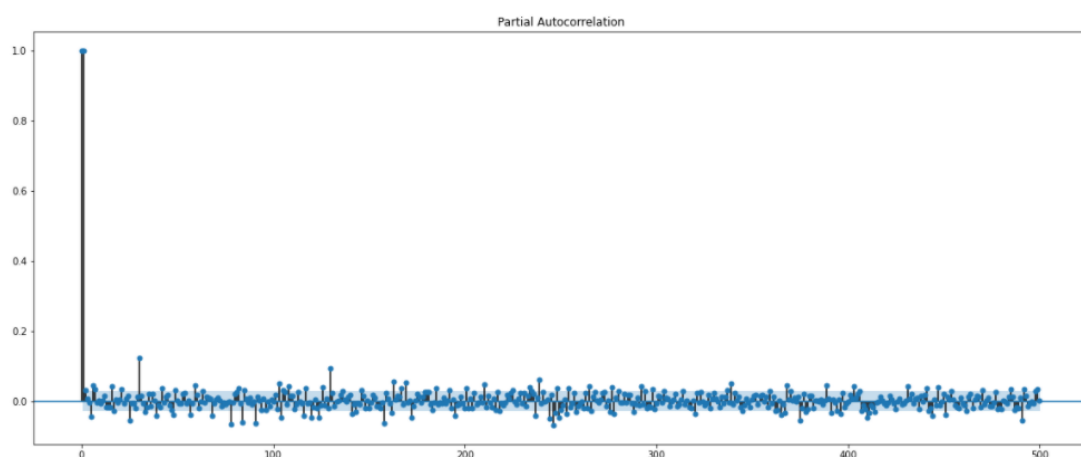


Рисунок 3.57 — ЧАКФ для ряду курс акцій Netflix в період 2002-2021.

Відповідно було обрано лаг порядку 50 та ковзні статистики порядку 4. Перейдемо до етапу моделювання моделей.

3.5.2. Моделювання дерева рішень

Підібрали параметри для моделі Дерева Рішень за допомогою жадібного алгоритму та кросвалідації тренувального дата-сету з кросвалідацією 5. Отримали відповідні оцінки зображені на рис. 3.58 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
DecisionTreeRegressor_train	0.99996402	1674.09224558	1.42013956	0.61851614	2.09437071	0.00251020
DecisionTreeRegressor_test	-0.16228630	1755970.37115921	11.78150592	76.50643483	0.02908394	0.08884721

Рисунок 3.58 — Статистика оцінки моделі для моделі Дерева Рішень

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.59.

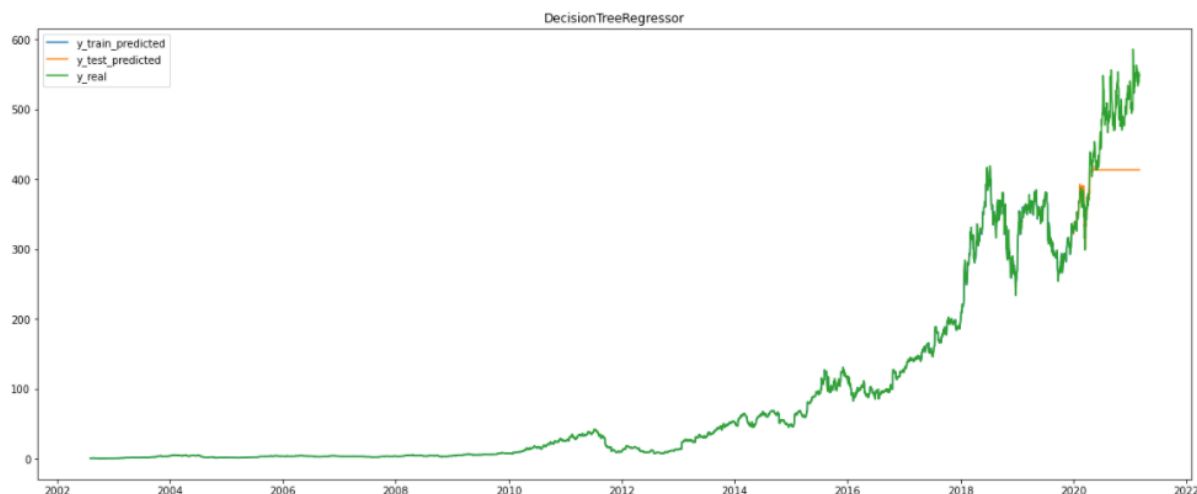


Рисунок 3.59 — Графік прогнозу моделі Дерева Рішень

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.60.

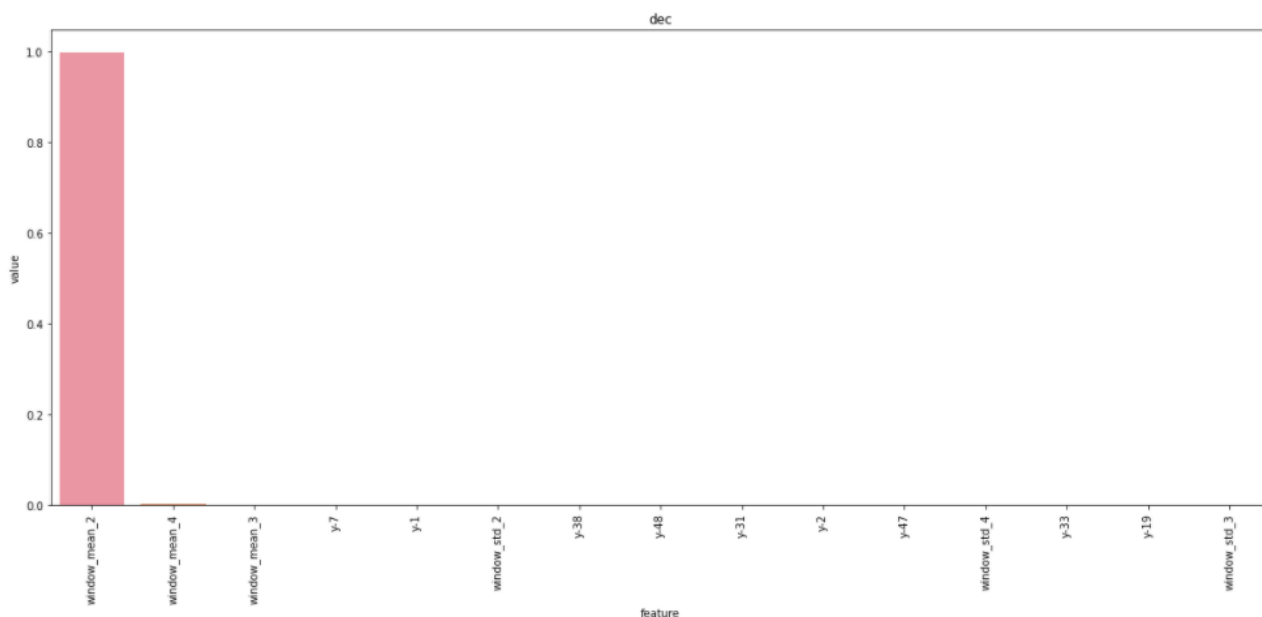


Рисунок 3.60 — Стовбчикова діаграма для важливості регресорів для моделі Дерева Рішень

Як бачимо вище найбільш важливою змінною для прогнозування Курс акцій Netflix для моделі Дерева Рішень це ковзне середнє порядку 2 яке має найбільше значення, а інші регресори майже не несуть інформативності даних моделі. Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування неможна.

3.5.3. Моделювання багатовимірної лінійної регресії

Знайшовши вагові значення для регресорів отримали відповідні оцінки зображені на рис. 3.70 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
LinearRegression_train	1.00000000	0.00000000	0.00000000	0.00000000	0.56890375	0.00000000
LinearRegression_test	1.00000000	0.00000000	0.00000000	0.00000000	0.31551639	0.00000000

Рисунок 3.70 — Статистика оцінки моделі для моделі лінійної регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеденим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.71.



Рисунок 3.71 — Графік прогнозу моделі лінійної регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.72.

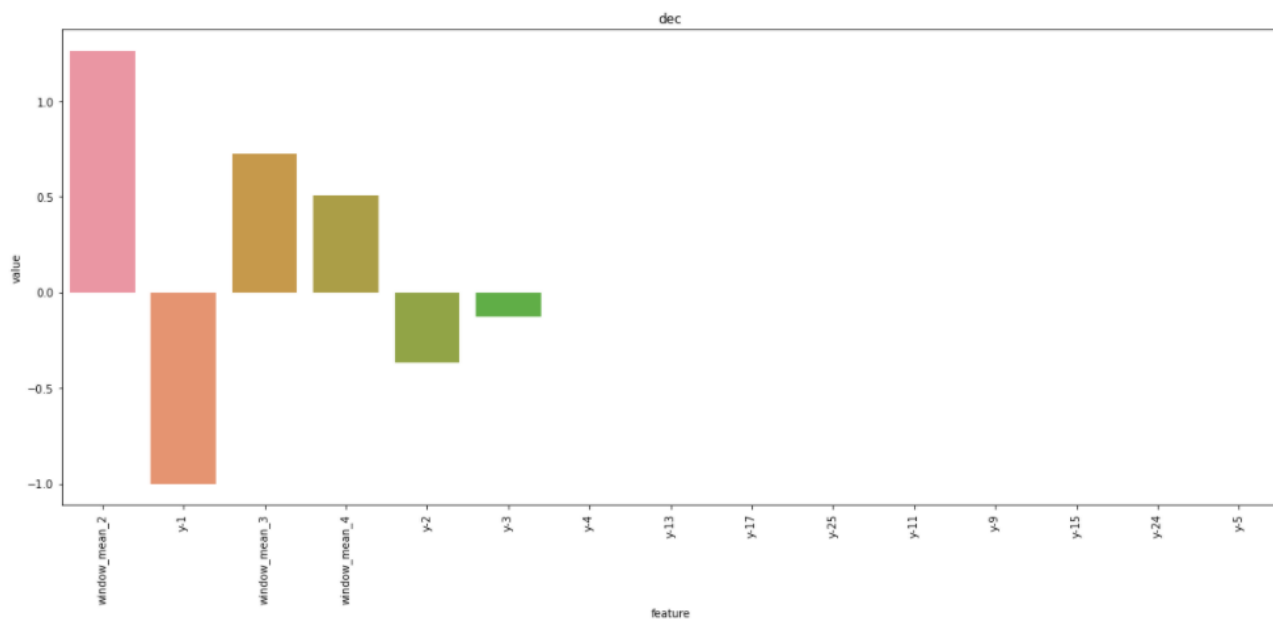


Рисунок 3.72 — Стовбчикова діаграма для важливості регресорів для моделі лінійної регресії

Як бачимо вище найбільш важливою змінною для прогнозування Курс акцій Netflix для моделі лінійної регресії це ковзне середнє порядку 2, 3 та 4, лаг порядку 1, 2,3. Інші регресори майже не несуть інформативності даній моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

3.5.4. Моделювання методу найближчих сусідів

Підібравши за допомогою крос-валідації та жадібного алгоритму параметри. Отримали відповідні оцінки зображені на рис. 3.73 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
KNeighborsRegressor_train	1.00000000	0.00000000	0.00000004	0.00000043	1.84689228	0.00000000
KNeighborsRegressor_test	-2.08733658	4664316.86228141	20.96599377	124.69050836	0.01363245	0.15244283

Рисунок 3.73 Статистика оцінки моделі для моделі методу найближчих сусідів

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеленим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.74.



Рисунок 3.74 — Графік прогнозу моделі методу найближчих сусідів

Дана модель показала не адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування не можна.

3.5.5. Моделювання Баєсової регресії

Побудувавши модель отримали відповідні оцінки зображені на рис. 3.75 де перший рядок відповідає тренувальному набору, а другий рядок тестовому.

name	r2	sum_squared_resid	MAPE	RMSE	durbin_watson	theil_index
BayesianRidge_train	1.00000000	0.00801821	0.01578874	0.00135363	0.00027530	0.00000549
BayesianRidge_test	0.99999999	0.00806394	0.00111288	0.00518457	0.00023816	0.00000561

Рисунок 3.75 — Статистика оцінки моделі для моделі Баєсової регресії

Для наочності зобразимо графік прогнозу на тестувальних даних та тренувальних де реальні дані зображені зеденим кольором, сині спрогнозовані значення для тренувального набору та помаранчеві для тестувального набору відповідно рисунок 3.76.



Рисунок 3.76 — Графік прогнозу моделі Баєсової регресії

Для того щоб зрозуміти важливість кожного з регресорів зобразимо найбільш важливі на рис. 3.77.

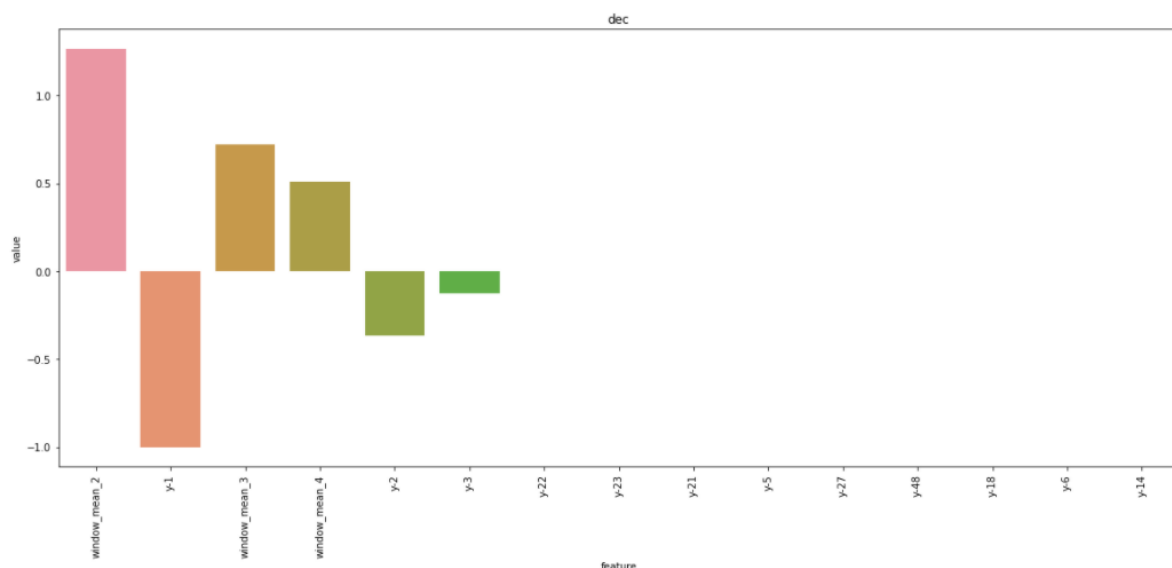


Рисунок 3.77 — Стовбчикова діаграма для важливості регресорів для моделі Баєсової регресії

Як бачимо вище найбільш важливою змінною для прогнозування ВВП Польщі для моделі Баєсової регресії це ковзне середнє порядку 2,3,4 та лаги порядку 1,2,3, а інші регресори майже не несуть інформативності даній моделі. Дана модель показала адекватність свого прогнозу на тестовій частині. Тому обрати її для подальшого застосування можна.

Висновок до розділу 3

Розглянувши чотири часові ряди, які є не стаціонарними, два з яких макроекономічні та два фінансовоекономічні. Використовуючи чотири моделі для того щоб спрогнозувати тенденцію процесів можемо зробити висновки, що найкращим з варіантів виявилася багатовимірна лінійна регресія. З іншого боку не сильно їй поступається модель Баєсівської регресії, яка має також хороші результати як на тренувальному так і на тестовому наборі. Моделі дерева рішень та метод найближчих сусідів виявилися слабкими для прогнозу нестаціонарних ряді, хоча показували гарні результати на тренувальному наборі на тестовому дали жахливі результати.

Також важливо зазначити що підготовка даних є доволі клопіткою роботою і в моделюванні моделей машинного навчання вважається однією з основних затрат часу. Оскільки дані були якісно підготовлені тому моделі лінійної регресії та баєсівської дали такі хороші результати. Проте не потрібно забувати, що був здійснений статичний прогноз за допомогою вище згаданих моделей, можливо якби був здійснений динамічний прогноз моделі не дали б такі хороші результати.

Також можна відмітити що після підготовки даних лінійна регресія показала бездоганні результати на усіх процес. Це говорить нам про те, що додаючи нові регресори збільшуючи тим розмірність моделі можна лінеаризувати дані.

Прийнятим рішенням можна вважати прийом моделі лінійної регресії для подальшого прогнозування процесів, оскільки модель дала найкращий результат для всіх процесів.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, розробленого для вирішення задач прогнозування нестационарних макроекономічних процесів за допомогою нечітких нейронних мереж.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозування нестаціонарних макроекономічних процесів. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по коронавірусу.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який вирішує задачу короткострокового прогнозування нестаціонарних макроекономічних процесів та будує його модель. Беручи за основу цю функцію, можна виділити наступні:

F_1 – вибір мови програмування;

F_2 – вибір фреймворку машинного навчання;

F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1 :

а) Python

б) Julia

Функція F_2 :

а) Tensorflow;

б) Java-ML

Функція F_3 :

а) Jupyter Notebook;

б) PyCharm.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 5.1).

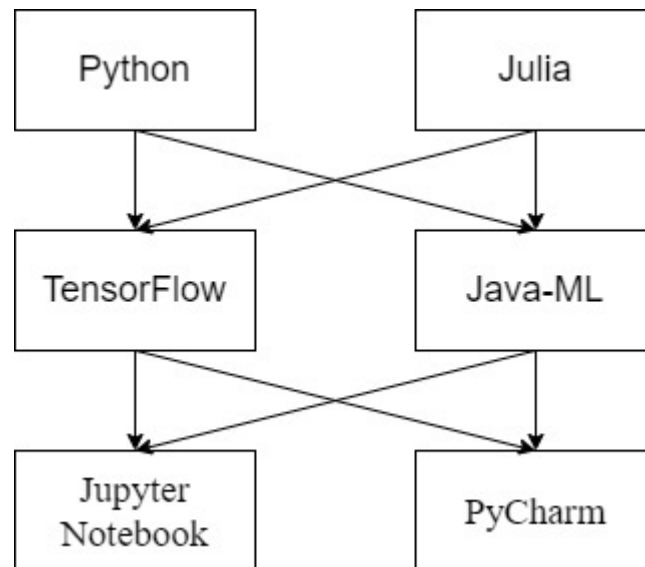


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіанти основних функцій.

Таблиця 5.2 - Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	A	Швидка розробка програми, доступність бібліотек, кросплатформеність	Низька швидкість роботи, особливо, якщо потрібно обробляти велику кількість даних
	B	Код швидко виконується	Іде багато часу на розробку програми
F_2	A	Надійно працює з складними проектами	Складність у використанні
	B	Надійність	Додатковий час на інсталяцію та вивчення
F_3	A	Підтримується багатьма мовами програмування, легко запускається на будь-якому сервері	Відсутня можливість роботи без інтернету
	B	IDE підтримує штучний інтелект для редагування коду	Підтримує одночасно лише одну мову програмування

На основі цієї карти будуємо позитивно-негативну матрицю варіантів основних функцій (Таб.4.2). Робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не

відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 :

Перевагу віддаємо швидкості вивчення, простоті використання та наявності стандартних бібліотек для обчислення. Для спрощення роботи по написанню коду варіант Б має бути відкинтий.

Функція F_2 :

Обидва варіанти можна використовувати в розробці.

Функція F_3 :

Віддаємо перевагу варіанту А в разі вибору мови програмування Python.

Таким чином, будемо розглядати такий варіанти реалізації ПП:

$$F_1a - F_2a - F_3a$$

$$F_1a - F_2b - F_3a$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів ПП

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- X_1 – швидкодія мови програмування;
- X_2 – об'єм пам'яті для обчислень та збереження даних;
- X_3 – час навчання даних;
- X_4 – потенційний об'єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 4.4.

Таблиця 4.4 - Основні параметри ПП

Назва Параметра	Умовні позначе ння	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	9000	13000	18000
Об'єм пам'яті	X2	Мб	512	128	32
Час попередньої обробки даних	X3	мс	10	5	3
Потенційний об'єм програмного коду	X4	кількість рядків коду	10000	5000	1300

За даними таблиці 4.3 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

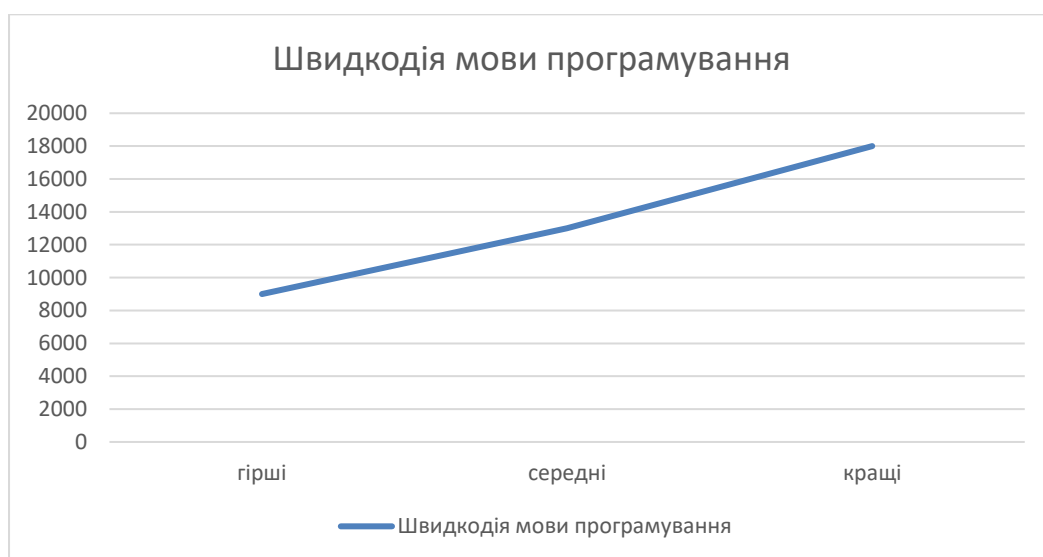


Рисунок 4.2 – X1, швидкодія мови програмування

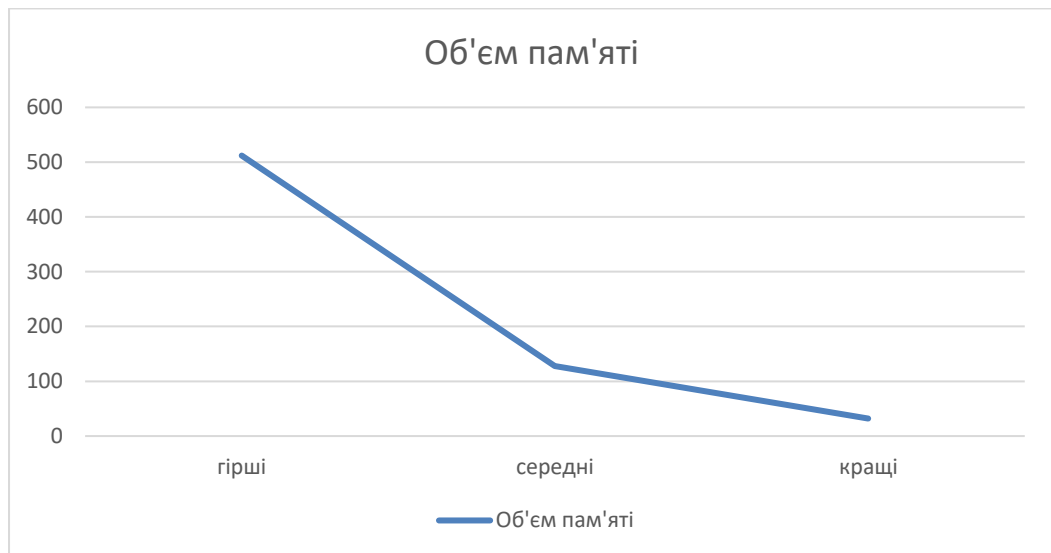


Рисунок 4.3 – X2, об'єм пам'яті

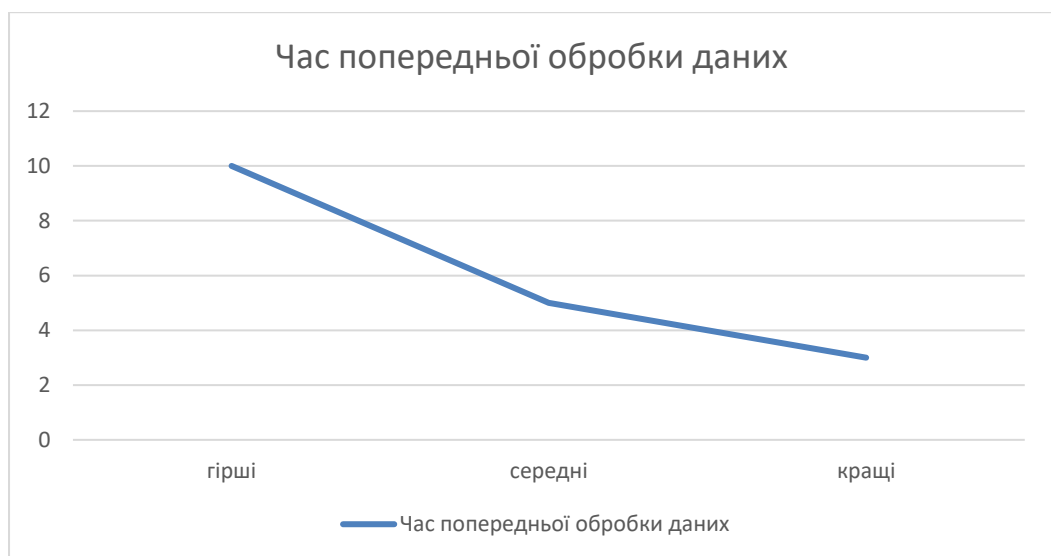


Рисунок 4.4 – X3, час попередньої обробки даних

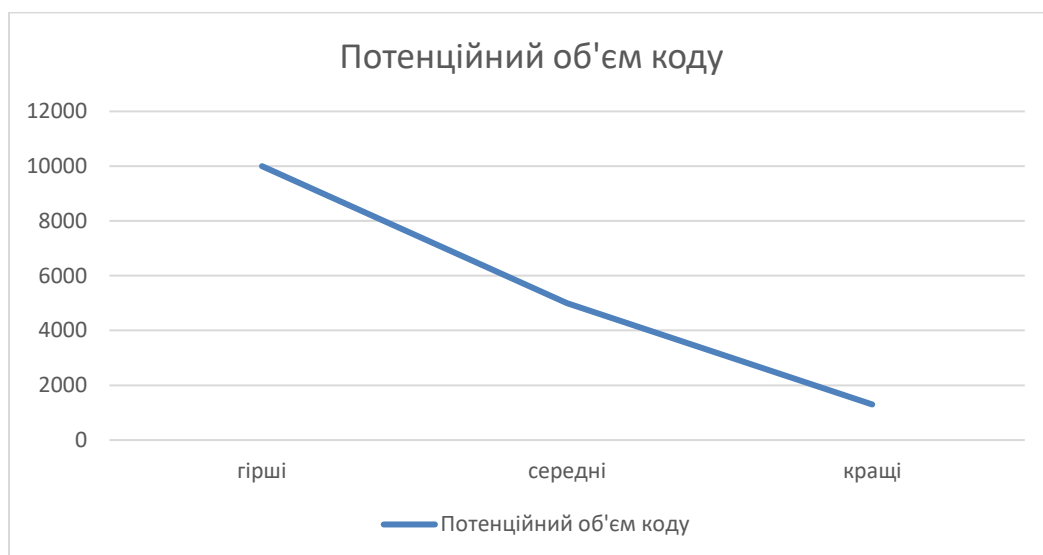


Рисунок 4.5 – Х4, потенційний об'єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 5.4.

Таблиця 4.4 - Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Швидкодія мови програмування	Оп/мс	1	1	1	1	1	1	2	8	-9,75	95,0625
X2	Об'єм пам'яті	Мб	3	2	3	2	3	2	1	16	-1,75	3,0625
X3	Час попередньої обробки даних	мс	2	3	2	3	2	3	6	22	4,25	18,0625
X4	Потенційний об'єм програмного коду	Кількість рядків коду	4	4	4	4	4	4	1	25	7,25	52,5625
	Разом		10	10	10	10	10	10	10	71	0	168,75

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 98, \quad (4.1)$$

де N – число експертів,

n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,75 \quad (4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (4.3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 168,75. \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 211}{7^2(4^3 - 4)} = 0,69 > W_k = 0,67. \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.5.

Таблиця 4.5 - Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	<	>	<	>	>	>	>	1,5
X1 і X3	=	<	=	=	>	=	=	=	1
X1 і X4	=	<	<	<	<	=	>	<	0,5
X2 і X3	=	=	=	=	>	>	=	=	1
X2 і X4	>	>	>	>	>	>	>	>	1,5
X3 і X4	>	=	>	=	=	>	>	>	1,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{ei} за наступними формулами:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.7)$$

$$b_i = \sum_{j=1}^N a_{ij} \quad (4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (4.9)$$

$$b'_i = \sum_{j=1}^N a_{ij} b_j \quad (4.10)$$

.Як видно з таблиці 5.4, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.6 - Розрахунок вагомості параметрів

Параметрих _i	Параметрих _j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
X1	1	1,5	1	0,5	4,00	0,25	16,00	0,25	64,00	0,24
X2	0,5	1,5	1	1,5	4,50	0,28	20,25	0,31	91,13	0,35
X3	0,5	1	1	1,5	4,00	0,25	16,00	0,25	64,00	0,24
X4	1,5	0,5	0,5	1	3,50	0,22	12,25	0,19	42,88	0,16
Всього:					16,00	1,00	64,50	1,00	262,00	1,00

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2 (Об'єм пам'яті), X3 (час попередньої обробки даних) та X4 (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X1 (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 5.7):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j}, \quad (4.11)$$

де n – кількість параметрів;

K_{ei} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.7 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	9000	6	0,24	1,44
F2	A	X2	32	3	0,35	1,05
	Б	X2	64	5	0,35	1,75
F3	A	X3	1000	9	0,24	2,16

За даними з таблиці 4.7 за формулою:

$$K_K = K_{Ty}[F_{1k}] + K_{Ty}[F_{2k}] + \dots + K_{Ty}[F_{zk}], \quad (4.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1,44 + 1,05 + 2,16 = 4,65,$$

$$K_{K2} = 1,44 + 1,75 + 2,16 = 5,35.$$

Як видно з розрахунків, кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.13)$$

де T_P – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{\Pi} = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації

для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.8$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 * 1,7 * 0,8 = 122,4 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_p = 27$ людино-днів, $K_{П} = 0.78$, $K_{СК} = 1$, $K_{СТ} = 0.8$:

$$T_2 = 27 \cdot 0,78 \cdot 0,8 = 16,85 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (122.4 + 16.85 + 4.8 + 16.85) \cdot 8 = 1287,2 \text{ людино-годин.}$$

$$T_{II} = (122.4 + 16.85 + 6.91 + 16.85) \cdot 8 = 1304,08 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 21325,54 грн., один аналітик в області даних з окладом 23698,71 . Визначимо середню зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (4.14)$$

де M – місячний оклад працівників;

T_m – кількість робочих днів тижень;

t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{21325,54 + 21325,54 + 23698,71}{3 \cdot 21 \cdot 8} = 131,7 \text{ грн.} \quad (4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_d, \quad (4.16)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

K_d – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$\text{I.} \quad C_{\text{зп}} = 131,7 \cdot 1287,2 \cdot 1,2 = 203429,1 \text{ грн.}$$

$$\text{II.} \quad C_{\text{зп}} = 131,7 \cdot 1304,08 \cdot 1,2 = 206\,096,8 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I.} \quad C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 203429,1 \cdot 0,22 = 44754,4 \text{ грн.}$$

$$\text{II.} \quad C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 206\,096,8 \cdot 0,22 = 45341,3 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_M)

Так як одна ЕОМ обслуговує одного програміста з окладом 21325,54 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 21325,54 \cdot 0,2 = 51181,3 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{\Gamma} \cdot (1 + K_3) = 51181,3 \cdot (1 + 0,2) = 61417,56 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0,22 = 61417,56 \cdot 0,22 = 13\,511,87 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 65435 грн.

$$C_A = K_{TM} \cdot K_A \cdot Ц_{ПР} = 1,15 \cdot 0,25 \cdot 20000 = 18812,56 \text{ грн.,}$$

де K_{TM} – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{TM} \cdot C_{ПР} \cdot K_P = 1.15 \cdot 20000 \cdot 0.05 = 3\,762,52 \text{ грн.},$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{\text{ЕФ}} &= (D_K - D_B - D_C - D_P) \cdot t_z \cdot K_B (365 - 115 - 16) \cdot 8 \cdot 0.9 = \\ &= 1684,8 \text{ годин}, \end{aligned}$$

де D_K – календарна кількість днів у році;

D_B, D_C – відповідно кількість вихідних та святкових днів;

D_P – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1684,8 \cdot 0,3 \cdot 0,88 \cdot 3,51547 = 1563,64 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 65435 \cdot 0.67 = 43\,841,45 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H, \quad (4.17)$$

$$C_{\text{ЕКС}} = 1684,8 + 13511,87 + 18812,5 + 3762,52 + 1563,64 + 43841,45 = 83\,176,78 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 83\,176,78 / 1684,8 = 49,37 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{\text{М-Г}} \cdot T, \quad (4.18)$$

$$\text{I. } C_M = 49,37 \cdot 1287,2 = 63549,064 \text{ грн.}$$

$$\text{II. } C_M = 49,37 \cdot 1304,08 = 64382,43 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{3П} \cdot 0,67, \quad (4.19)$$

$$\text{I.} \quad C_H = 203429,1 \cdot 0,67 = 136\,297,5 \text{ грн.}$$

$$\text{II.} \quad C_H = 206\,096,8 \cdot 0,67 = 138\,084,85 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{3П} + C_{ВІД} + C_M + C_H, \quad (4.20)$$

$$\text{I.} \quad C_{ПП} = 203429,1 + 44\,754,4 + 63549,064 + 136297,5 = 448029,564 \text{ грн.}$$

$$\text{II.} \quad C_{ПП} = 206\,096,8 + 45341,3 + 64\,382,43 + 138084,85 = 453905,38 \text{ грн.}$$

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}j} = K_{kj} / C_{\Phi j}, \quad (4.21)$$

$$K_{\text{ТЕР}1} = 4,65 / 448029,564 = 1,04 \cdot 10^{-5},$$

$$K_{\text{ТЕР}2} = 5,35 / 453905,38 = 1,18 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{ТЕР}1} = 1,18 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{ТЕР}} = 1,18 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – Python;
- Використання моделей з великою ємністю
- Використання стандартного інтерфейсу візуалізації, швидкість розробки

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, непоганий функціонал і швидкодію.

Висновки до розділу 4

Проведено повний функціонально-вартісний аналіз програмного продукту. Визначено та проведено оцінку основних функцій програмного продукту. Визначено параметри, які характеризують програмний продукт. Проведено експертне оцінювання параметрів та аналіз якості варіантів реалізації функцій.

Проведено економічний аналіз варіантів розробки – трудомісткість, витрати на заробітну плату та інші витрати.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ

Отже, всі поставлені задачі були вирішені та досягнуто мети дослідження: досліджено альтернативні моделі для прогнозу макроекономічних та фінансових процесів які використовуються на ринку та досліджено їх якість на вибраних часових рядах, а саме: ВВП України, ВВП Польщі, курси акцій компаній Facebook та Netflix. Та використано ці моделі для прогнозування заданих процесів.

В розділі 1 було проведено проаналізовано задачі прогнозування та моделювання макроекономічних та фінансових часових рядів.

В розділі 2 було досліджено альтернативні методи прогнозування часових рядів. Та методи підготовки даних.

В розділі 3 проведено моделювання обраних моделей на чотирьох вище перелічених часових рядах. Підведено підсумки на які моделі показали найкращі результати роботи на тренувальному та тестовому наборі.

В розділі 4 було проведено функціонально-вартісний аналіз продукту.

Як результат отримали, що для підготовки такого вигляду даних, найкращими моделями виявилися лінійна та баєсівська регресія. Моделі дерев рішень та методу найблизчих сусідів показали або опосередковані результати, або незадовільні.

В результаті можемо сказати що важливим етапом є підготовка даних та вибір моделей які співвідносяться до цих даних

ПЕРЕЛІК ПОСИЛАНЬ

1. Diebold. Forecasting in Economics, Business, Finance and Beyond. Francis X. 2017.
2. Robert H. Sgumway, David S. Stoffer. Time Series Analysis and Its Applications. 2011. Springer. С. 316–339.
3. Демківський Є.О., Демківська Т.І. Прогнозування нестационарних фінансово-економічних процесів. IV-а Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг»: Київ, КНУТД, 22 жовтня 2020 року, с. 39-40
4. Time Series Analysis Tutorial Using Financial Data URL: <https://towardsdatascience.com/time-series-analysis-tutorial-using-financial-data-4d1b846489f9> (дата звернення: 04.04.2021).
5. Principles of Predictive Abalytics and the Path to Time-Series Predictions URL <https://medium.com/@ODSC/principles-of-predictive-analytics-and-the-path-to-time-series-predictions-a61fb547efa6> (дата звернення: 24.03.2021).
6. Rolling Window Regression: a Simple Approach for Time Series Next value Predictoins URL: <https://medium.com/making-sense-of-data/time-series-next-value-prediction-using-regression-over-a-rolling-window-228f0acae363> (дата звернення: 31.03.2021).
7. Financial, Economic and Alternative Data. Quandl URL: <https://www.quandl.com/> (дата звернення: 10.2.2021)
8. Бідюк П.І., Коршевніук Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень. Київ, 2010. С. 115–136.
9. Probabilisic data analysis in forecasting nonstationary processes. Kuznietsova N.V., Polozhayenko S.A., Levenchuk L.B.. System Analysis and Information Technologies: IEEE-Xplore, SAIC – 2020; Kyiv, October, 2020, pp. 297 – 302.
10. Spirtes, P., Glymour, C. Scheines, R., Causation, prediction and search. Adaptive computation and machine learning. MIT press, 2001. P. 540–600

11. Бідюк П.І., Меньяйленко О.С., Половцев О. В. Методи прогнозування. Луганськ: Альма-Матер, 2008. С. 583–607.
12. Python Release Python 3.9.2 URL:
<https://www.python.org/downloads/release/python-392/> (дата звернення: 01.04.2021).
13. Facebook Inc Stock Predictions URL: <https://financhill.com/stock-forecast/fb-stock-prediction> (дата звернення: 28.04.2021).
14. Financial Forecasting Preparing a prediction of the future URL:
<https://corporatefinanceinstitute.com/resources/knowledge/modeling/financial-forecasting-guide/> (дата звернення: 28.02.2021).
15. Predicing the Stock Market with Machine Learning.Introduction. URL:
<https://towardsdatascience.com/predicting-the-stock-market-with-machine-learning-introduction-310cd6069ffa> (дата звернення: 20.02.2021).
16. Predicting Stock Returns Using Financial Statement Information URL:
<https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5957.t01-1-00205> (дата звернення: 18.02.2021)
17. Das and M. Chen. Yahoo for Amazon: Extracting market sentiment from stock message boards.2001.
18. Lerman, A. Gilder, M. Dredze, F.Pereira. Reading the markets: Forecasting public opinion of political candidates by news analysis.2008.
19. How to Forecast Financial Statements: Balance Sheets, Income Statements URL: <https://www.freshbooks.com/hub/accounting/how-to-forecast-balance-sheet-and-income-statement> (дата звернення: 18.02.2021).
20. Predicting Stock Market Structure with Social and Financial Networks URL:
<https://medium.com/@souzatharsis/predicting-stock-market-structure-with-social-and-financial-networks-870a0fb1343f> (дата звернення: 18.05.2021).
21. Stock Market Forecasting Using Time Series Analysis URL:
<https://www.kdnuggets.com/2020/01/stock-market-forecasting-time-series-analysis.html> (дата звернення: 20.05.2021).

22. Koenker R. Quantile Regression. Cambridge University Press. 2005. P. 146–147
23. Branham R.L. Alternatives to least squares. *Astronomical Journal* 87. 1982. P.928–937
24. Breiman L., Friedman J.H., Olshen R. A., Stone C. J. Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. 1984. P. 312
25. Kennedy P. A Guide to Econometrics. Cambridge: The MIT Press. 2003. Vol.5, P. 205–206
26. Cook R.D. Sanford Weisberg Criticism and Influence Analysis in Regression. *Sociological Methodology*. 1982. Vol. 13, P. 313–361 c.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics

from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import pacf
from statsmodels.tsa.stattools import acf
from statsmodels.stats.stattools import durbin_watson
import statsmodels.api as sm
from sklearn.metrics import mean_absolute_error

from scipy.stats import jarque_bera
from sklearn.metrics import r2_score
import warnings

from sklearn.metrics import mean_squared_error
import seaborn as sns
```

```

import matplotlib.pyplot as plt
from scipy.stats import boxcox

warnings.filterwarnings('ignore')
get_ipython().run_line_magic('matplotlib', 'inline')
# plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = [20,8]
# lags=60
# precent_train_set = 0.9
# rolling_value = 10

```

```
# In[2]:
```

```

df_1 = pd.read_csv('../data_dm/eugen/FB.csv', header=0, index_col=0,
parse_dates=True)
df_1

```

```
# In[3]:
```

```

df_1 = pd.read_csv('../data_dm/eugen/NFLX.csv', header=0, index_col=0,
parse_dates=True)
df_1

```

```
# In[4]:
```

```
df_1 = pd.read_csv('../data_dm/eugen/gdp_poland.csv', header=0,
index_col=0, parse_dates=True)
```

```
df_1
```

```
# In[5]:
```

```
df = pd.read_csv('../data_dm/eugen/gdp_ukraine.csv', header=0, index_col=0,
parse_dates=True)
```

```
df
```

```
# # GDP Ukraine
```

```
# In[2]:
```

```
df = pd.read_csv('../data_dm/eugen/gdp_ukraine.csv', header=0, index_col=0,
parse_dates=True)
```

```
df = df / 1e+10
```

```
y = df
```

```
df
```

```
# In[7]:
```

```
df.plot()
```

```
# ## Decompose data
```

```
# In[8]:
```

```
def seasonal_decompose(y):
    decomposition = sm.tsa.seasonal_decompose(y,
model='additive',extrapolate_trend='freq')
    fig = decomposition.plot()
    fig.set_size_inches(14,7)
    plt.show()
```

```
# In[9]:
```

```
seasonal_decompose(df)
```

```
# In[10]:
```

```
# ## Checking Stationarity (do not work for gdp)
```

```
# In[11]:
```

```
def test_stationarity(timeseries, title):
```

```
    #Determining rolling statistics
```

```
    rolmean = pd.Series(timeseries).rolling(window=rolling_value).mean()
```

```
    rolstd = pd.Series(timeseries).rolling(window=rolling_value).std()
```

```
    fig, ax = plt.subplots(figsize=(16, 4))
```

```
    ax.plot(timeseries, label= title)
```

```

ax.plot(rolmean, label='rolling mean');
ax.plot(rolstd, label='rolling std (x10)');
ax.legend()

```

```
# In[12]:
```

```
rolling_value = 7
```

```

pd.options.display.float_format = '{:.8f}'.format
test_stationarity(y.gdp,'raw data')

```

```
# ## Augmented Dickey-Fuller Test
```

```
# In[13]:
```

```

def ADF_test(timeseries, dataDesc):
    print(' > Is the {} stationary?'.format(dataDesc))
    dfctest = adfuller(timeseries.dropna(), autolag='AIC')
    print('Test statistic = {:.3f}'.format(dfctest[0]))
    print('P-value = {:.3f}'.format(dfctest[1]))
    print('Critical values :')
    for k, v in dfctest[4].items():
        print('\t{}: {} - The data is {} stationary with {}% confidence'.format(k, v,
'not' if v<dfctest[0] else "", 100-int(k[:-1])))

```

```
# In[14]:
```

```
ADF_test(y,'raw data')
```

```
# In[ ]:
```

```
# ## Detrending
```

```
# In[15]:
```

```
rolling_value = 3
```

```
# In[16]:
```

```
# Detrending
```

```
y_detrend = (y - y.rolling(window=rolling_value).mean())/y.rolling(window=rolling_value).std()
```

```
test_stationarity(y_detrend.gdp,'de-trended data')
```

```
ADF_test(y_detrend,'de-trended data')
```

```
# ## Differencing
```

```
# In[17]:
```

```
# Differencing
```

```
diff_model = 1
```

```
y_10lag = y - y.shift(diff_model)
```

```
test_stationarity(y_10lag.gdp,f'{diff_model} lag differenced data')
```

```
ADF_test(y_10lag,f'{diff_model} lag differenced data')
```

```
# ## Combining Detrending and Differencing
```

```
#
```

```
# In[18]:
```

```
y_10lag_detrend = y_detrend - y_detrend.shift(diff_model)
```

```
test_stationarity(y_10lag_detrend.gdp,f'{diff_model} lag differenced de-  
trended data')
```

```
ADF_test(y_10lag_detrend.gdp,f'{diff_model} lag differenced de-trended data')
```

```
# ## Create Training & Testing Datasets
```

```
#
```

```
# In[19]:
```

```

fig, ax = plt.subplots(figsize=(20, 10))
ax.plot(y_10lag_detrend, marker='.', linestyle='-', linewidth=0.5, label='Year')
ax.plot(y_10lag_detrend.resample('Y').mean(), marker='o', markersize=8,
linestyle='-', label='Year Mean Resample')
ax.set_ylabel('Orders')
ax.legend();

```

```
# In[4]:
```

```
def calculate_metrics_sample(y):
```

```
    print('yes')
```

```
    sample = pd.Series(y)
```

```
    y_dict = dict()
```

```
#    y_dict['series'] = 'Y'
```

```
    y_dict['observations'] = sample.size
```

```
    y_dict['mode'] = sample.mode()[0]
```

```
    y_dict['mean'] = sample.mean()
```

```
    y_dict['median'] = sample.median()
```

```
    y_dict['variation'] = sample.var()
```

```
    y_dict['skewness'] = sample.skew()
```

```
    y_dict['kurtosis'] = sample.kurt()
```

```
    y_dict['jarque-bera'] = jarque_bera(sample).statistic
```

```
    y_dict['probability'] = jarque_bera(sample).pvalue
```

```

return pd.DataFrame(np.array([[metr] for metr in y_dict.values()]).T,
                    columns=[ 'observations', 'mode','mean','median',
                              'variation','skewness','kurtosis','jarque-
bera','probability']).astype('float16')#, 'params_ar', 'params_intercept'])

```

```

# ## Metrics

```

```

# In[7]:

```

```

#                columns                =['name',                'r2',
'RMSE','MAE','sum_squared_resid','AIC','durbin_watson','theil_index']

```

```

# models_metrics_static = pd.DataFrame(columns=columns)

```

```

# models_metrics_static_test = pd.DataFrame(columns=columns)

```

```

calculate_metrics_sample(y.gdp)

```

```

# In[22]:

```

```

y.hist(figsize=(10,10))

```

```

plt.show()

```

```

# In[23]:

```

```

y.plot(kind='kde', figsize=(10,10))

```

```

plt.show()

```

```

# ## Box and Whisker Plots by Interval (for GDP does not work)

```

```
# #### Years
```

```
# In[24]:
```

```
groups = series.groupby(pd.Grouper(freq='Y'))
years = pd.DataFrame()
for name, group in groups:
    # print(name.year)
    # print(np.reshape(group.values, (-1,)))
    # print(name.year)
    years[name.year] = np.reshape(group.values, (-1,))

# years.plot(kind='boxplot', subplots=True, legend=False, figsize=(15, 20))
plt.figure(figsize=(10,5))
years.boxplot()
plt.show()
```

```
# #### Months
```

```
# In[25]:
```

```
one_year = series['1990']
groups = one_year.groupby(pd.Grouper(freq='M'))

months = pd.concat([pd.DataFrame(x[1].values) for x in groups], axis=1)
months = pd.DataFrame(months)
months.columns = range(1,13)
```

```
plt.figure(figsize=(10,5))
months.boxplot()
plt.show()

# #### Box-Cox Transformed Dataset

# In[26]:

from scipy.stats import boxcox
from statsmodels.graphics.gofplots import qqplot

X = y.gdp
transformaed, lam = boxcox(X)

print(f'Lambda:\t{lam}')

plt.figure(figsize=(20,30))
plt.subplot(311)
plt.plot(transformaed)
plt.subplot(312)
plt.hist(transformaed)
#q-q plot
plt.subplot(313)
qqplot(transformaed, line='r', ax=plt.gca())
plt.show()
```

```
# In[27]:
```

```
def show_plots(y):  
    sample = pd.Series(y)  
    # plt.subplot(311)  
    # plt.figure(figsize=(20,6))  
    plt.title('Sample')  
    plt.xlabel('k')  
    plt.ylabel('y')  
    plt.plot(sample, marker='o')  
    plt.grid(True)  
  
    # plt.subplot(312)  
    plot_acf(sample, lags=lags);  
  
    # plt.subplot(313)  
    plot_pacf(sample, lags=lags);  
  
    plt.show()
```

```
# In[28]:
```

```
lags=10  
show_plots(y.gdp)
```

```
# ## Preparing data
```

```
#
```

```
# In[29]:
```

```
## Preparing data
```

```
p = 2
```

```
q = 7
```

```
# In[3]:
```

```
from sklearn.pipeline import Pipeline
```

```
from sklearn.base import BaseEstimator, RegressorMixin, TransformerMixin,
```

```
clone
```

```
from sklearn.preprocessing import PolynomialFeatures
```

```
import seaborn as sns
```

```
def barplot(data, title):
```

```
#    fig = plt.figure(figsize=(18,6))
```

```
    bar_plot = sns.barplot(x=data['feature'], y=data['value'])
```

```
    for item in bar_plot.get_xticklabels():
```

```
        item.set_rotation(90)
```

```
    plt.title(title)
```

```
    plt.show()
```

```
class DiffData(BaseEstimator, TransformerMixin):
```

```

def __init__(self, diff_model=0, box_cox=False, ln=False):
    self.diff_model = diff_model
    self.box_cox = box_cox
    self.ln = ln

def fit(self, X, y=None):

    index = X.index
    y_value = X['gdp'].values

    #    print(X_copy)
    if self.box_cox:
        transformed, self.lmbd = boxcox(y_value)

    if not self.box_cox and self.ln:
        transformed = np.log(y_value)

    transformed = pd.DataFrame({'value' : y_value}, index=index)

    return self

def transform(self, X, y=None):
    X_copy = X.copy()

    index = X.index
    y_value = X['gdp'].values

    if self.box_cox:

```

```

transformed = boxcox(y_value, lmbda=self.lmbd)

if not self.box_cox and self.ln:
    transformed = np.log(y_value)

transformed = pd.DataFrame({'value' : y_value}, index=index)

#    transformed = (transformed - transformed.shift(self.diff_model)).dropna()
if self.diff_model:
    transformed = transformed - transformed.shift(self.diff_model)

return transformed

class PrepareData(BaseEstimator, TransformerMixin):
    def __init__(self, lags=0, window=2):
#        self.box_cox = box_cox
        self.lags = lags
        self.window = 2
#        self.diff = diff_model

    def fit(self, X, y=None):
#        X_copy = X.copy()

#        transformed = pd.DataFrame()

#        if self.box_Cox:
#            transformed, self.lmbd = boxcox(X_copy)

#        transformed = transformed - transformed.shift(self.diff_model)

```

```

#     for i in range(1, self.lags + 1):
#         transformed[f'y-{i}'] = transformed['value'].shift(i)

#     for i in range(2, self.window + 1):
#         transformed[f'window_mean_{i}'] =
transformed['value'].rolling(window=i).mean()
#         transformed[f'window_std_{i}'] =
transformed['value'].rolling(window=i).std()

    return self

def transform(self, X, y=None):
    X_copy = X.copy()
    index = X_copy.index
    y_value = X['value'].values

    df_temp = pd.DataFrame({'value':y_value}, index=index)

    transformed = pd.DataFrame()

    for i in range(1, self.lags + 1):
        transformed[f'y-{i}'] = df_temp['value'].shift(i)

    for i in range(2, self.window + 1):
        transformed[f'window_mean_{i}'] =
df_temp['value'].rolling(window=i).mean()
        transformed[f'window_std_{i}'] =
df_temp['value'].rolling(window=i).std()

```

```

transformed['value'] = y_value
return transformed.dropna()

# prepera = Pipeline([('diff_box_cox', DiffData(diff_model=1, box_cox=True)),
#                     ('prepare', PrepareData(lags=2, window=2))])

# prepera.fit(y, y)

# ## Fiting Models

# In[4]:

from sklearn.linear_model import LinearRegression, Lasso, BayesianRidge,
SGDRegressor, Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import AdaBoostRegressor, RandomForestRegressor,
GradientBoostingRegressor
from sklearn.model_selection import cross_val_score, GridSearchCV, KFold,
StratifiedKFold

from xgboost import XGBRegressor, XGBRFRegressor

from modules import define_metrics, get_score_for_model

# In[5]:

def show_pred(y_train, y_test, y, title):
    y_train_copy = pd.Series(y_train, index=y.index[:len(y_train)])

```

```

y_test_copy = pd.Series(y_test, index=y.index[-len(y_test):])

# print(y_test_copy)
# print(f'y_train_copy shape:\t{y_train_copy.shape}')
# print(f'y_test_copy shape:\t{y_test_copy.shape}')
# print(f'y shape:\t{y.shape}')

plt.plot(y_train_copy, label='y_train_predicted')
plt.plot(y_test_copy, label='y_test_predicted')
plt.plot(y, label='y_real')
plt.title(title)
plt.legend()
plt.show()

### Without preparations (diff and ln)

# In[6]:

pipeline_repare_without = Pipeline([('diff_box_cox', DiffData(diff_model=0,
box_cox=False)),
                                   ('prepare', PrepareData(lags=2, window=2))])

pipeline_repare_without.fit(y, y)

data_prepare_without = pipeline_repare_without.transform(y)
data_prepare_without

#### Creating train and test sets

# In[7]:

```

```
data_prepare_without.to_csv('prepared data/gdp_Ukraine.csv')
```

```
# In[10]:
```

```
features = data_prepare_without.columns[:-1]
```

```
X = data_prepare_without.drop(['value'], axis=1)
```

```
y_real = data_prepare_without['value']
```

```
count_test = 3
```

```
X_train = X.iloc[:-count_test, :]
```

```
y_real_train = y_real.iloc[:-count_test]
```

```
X_test = X.iloc[-count_test:, :]
```

```
y_real_test = y_real.iloc[-count_test:]
```

```
print(f'X_train shape:\t{X_train.shape}')
```

```
print(f'y_real_train shape:\t{y_real_train.shape}')
```

```
print()
```

```
print(f'X_test shape:\t{X_test.shape}')
```

```
print(f'y_real_test shape:\t{y_real_test.shape}')
```

```
# In[ ]:
```

```
# Loking how work models
```

```
# In[16]:
```

```
get_ipython().run_cell_magic('time', '', "models =\n\nmodels.append(('LinearRegression', LinearRegression()))\n#\nmodels.append(('Lasso', Lasso()))\n#\nmodels.append(('Ridge', Ridge()))\nmodels.append(('BayesianRidge', BayesianRidge()))\nmodels.append(('KNeighborsRegressor', KNeighborsRegressor()))\nmodels.append(('DecisionTreeRegressor', DecisionTreeRegressor()))\n\nmodels.append(('GradientBoostingRegressor', GradientBoostingRegressor()))\nmodels.append(('RandomForestRegressor', RandomForestRegressor()))\n\n# models.append(('LinearSVR', LinearSVR()))\n#\nmodels.append(('XGBRegressor', XGBRegressor()))\n#\nmodels.append(('XGBRFRegressor', XGBRFRegressor()))\n#\nmodels.append(('SVR', SVR()))\n\nbox_plot, df_means = get_score_for_model(models, X_train, y_real_train, n_splits=10, scoring='neg_mean_absolute_error', print_res=True)")
```

```
# In[17]:
```

```
plt.figure(figsize=(12,7))
plt.title("neg_mean_absolute_error")
sns.boxplot(data = box_plot, x = box_plot.columns[1],y=box_plot.columns[0]);
```

```
# In[18]:
```

```
box_plot_new = box_plot.copy()
```



```

importances = decTree_sep.feature_importances_
feat_imp = dict((value, name) for value, name in zip(features, importances))

feat_imp = sorted(feat_imp.items(), key=lambda x: np.abs(x[1]), reverse=True)
feat_imp = pd.DataFrame(feat_imp, columns=['feature', 'value'])

```

```
# In[23]:
```

```
feat_imp
```

```
# In[24]:
```

```
barplot(feat_imp, title='dec')
```

```
# # LinearRegression
```

```
# In[25]:
```

```

get_ipython().run_cell_magic('time', '', "lin_reg =
LinearRegression()\nlin_reg.fit(X_train, y_real_train)\nny_train_predict_linReg =
lin_reg.predict(X_train)\nny_test_predict_linReg = lin_reg.predict(X_test)\nname =
'LinearRegression'")

```

```
# ### Show predictions
```

```
# In[26]:
```

```

show_pred(y_train=y_train_predict_linReg,
          y_test=y_test_predict_linReg,
          y=y_real,
          title=name)

```

```

# #### Show Features importance

```

```

# In[27]:

```

```

importances = lin_reg.coef_
feat_imp = dict((value, name) for value, name in zip(features, importances))

feat_imp = sorted(feat_imp.items(), key=lambda x: np.abs(x[1]), reverse=True)
feat_imp = pd.DataFrame(feat_imp, columns=['feature', 'value'])

```

```

# In[28]:

```

```

feat_imp

```

```

# In[29]:

```

```

barplot(feat_imp, title='dec')

```

```

# # KNeighborsRegressor

```

```

# In[30]:

```

```

get_ipython().run_cell_magic('time', "", "knn = KNeighborsRegressor()\ngrid =
{'n_neighbors': np.linspace(1,5, dtype=int),\n      'weights': ['uniform', 'distance'],\n
'algorithm': ['ball_tree', 'kd_tree', 'brute', 'auto']\n}\n\nn_splits = 10\nclf2 =
GridSearchCV(knn, grid, n_jobs=-1,\n              cv=KFold(n_splits=n_splits,
shuffle=True),\n              scoring='neg_mean_absolute_error',\n              verbose=5,
refit=True)\n\nclf2.fit(X_train,
y_real_train)\n\nprint()\nprint(clf2.best_params_)\nprint(clf2.best_score_)\nprint(clf2
.best_estimator_)\n\nname='KNeighborsRegressor'\nmodel = clf2\nscore_model =
define_metrics(model,      X_train,      X_test,      y_real_train,      y_real_test,
name)\nscore_model")
# In[31]:

```

```

get_ipython().run_cell_magic('time', "", "knn_sep = clf2.best_estimator_ #
DecisionTreeRegressor(criterion='mae', max_depth=4)\n# decTree_sep.fit(X_train,
y_real_train)\n\nny_train_predict_knn= knn_sep.predict(X_train)\nny_test_predict_knn
= knn_sep.predict(X_test)")

```

```

# ### Show predictions

```

```

# In[32]:

```

```

show_pred(y_train=y_train_predict_knn,
          y_test=y_test_predict_knn,
          y=y_real,
          title=name)

```

```

# # BayesianRidge

```

```

# In[33]:

```

```

get_ipython().run_cell_magic('time', "", "bayes = BayesianRidge()\ngrid =
{'n_iter': np.linspace(100,500, 5,dtype=int)\n}\n\nn_splits = 10\ncf3 =
GridSearchCV(bayes, grid, n_jobs=-1,\n                cv=KFold(n_splits=n_splits,
shuffle=True),\n                scoring='neg_mean_absolute_error',\n                verbose=5,
refit=True)\nncf3.fit(X_train,
y_real_train)\n\nprint()\nprint(cf3.best_params_)\nprint(cf3.best_score_)\nprint(cf3
.best_estimator_)\n\nname='BayesianRidge'\nmodel = cf3\nscore_model =
define_metrics(model, X_train, X_test, y_real_train, y_real_test,
name)\nscore_model")

```

In[34]:

```

get_ipython().run_cell_magic('time', "", "bayes_sep = cf3.best_estimator_ #
DecisionTreeRegressor(criterion='mae', max_depth=4)\n# decTree_sep.fit(X_train,
y_real_train)\n\nny_train_predict_bayes = bayes_sep.predict(X_train)\nny_test_predict_bayes = bayes_sep.predict(X_test)")

```

Show predictions

In[36]:

```

show_pred(y_train=y_train_predict_bayes,
          y_test=y_test_predict_bayes,
          y=y_real,
          title=name)

```

Show Features importance

```
# In[37]:
```

```
importances = bayes_sep.coef_
```

```
feat_imp = dict((value, name) for value, name in zip(features, importances))
```

```
feat_imp = sorted(feat_imp.items(), key=lambda x: np.abs(x[1]), reverse=True)
```

```
feat_imp = pd.DataFrame(feat_imp, columns=['feature', 'value'])
```

```
# In[38]:
```

```
feat_imp
```

```
# In[39]:
```

```
barplot(feat_imp, title='dec')
```

ДОДАТОК Б ПЕРЕЛІК ГРАФІЧНОГО МАТЕРІАЛУ ДО ЗАХИСТУ

Альтернативні методи прогнозування макроекономічних процесів

ВИКОНАВ: ОРЕЛ ЄВГЕН ВЛАДИСЛАВОВИЧ

НАУКОВИЙ КЕРІВНИК: Д.Т.Н., ПРОФЕСОР

БІДЮК ПЕТРО ІВАНОВИЧ,

Постановка задачі

- ▶ Аналіз інформаційного ринку для отримання даних.
- ▶ Аналіз статистичних моделей для прогнозування процесів
- ▶ Аналіз методів перетворення часових рядів
- ▶ Вибір альтернативних методів
- ▶ Підготовка даних для застосування альтернативних методів
- ▶ Моделювання альтернативних моделей на кожному з часових рядів
- ▶ Порівняння роботи моделей з різними рядами
- ▶ Аналіз ефективності створених моделей

Актуальність роботи

- ▶ Прогнозування має величезне значення для розрахунку цін на електроенергію, скільки потрібно виготовити продукції, очікувані прибутки від бізнесу, очікуване зростання ВВП країни тощо. Зачасту використовують математичні або статистичні моделі для прогнозу на ринку, однак вони не завжди працюють в такому разі використовують альтернативні моделі для прогнозування які дають набагато кращі результати ніж математичні.

- ▶ Об'єкт дослідження:

Нестационарні процеси: ВВП України, ВВП Польщі, курс акцій Netflix, Facebook

- ▶ Предмет дослідження:

Методи прогнозу макроекономічних процесів.

- ▶ Мета роботи:

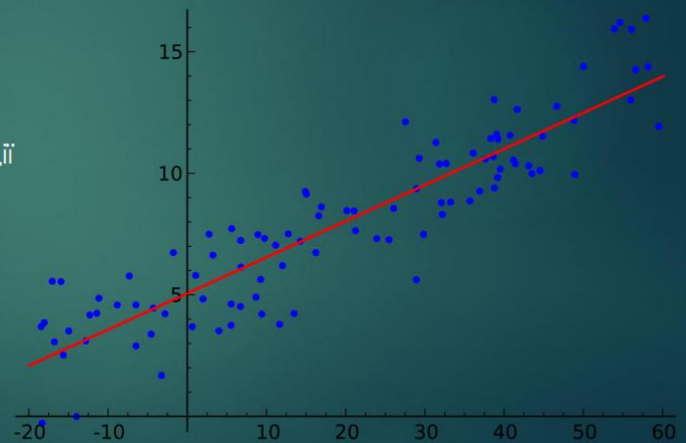
Побудова та порівняльний аналіз альтернативних моделей використовуючи макроекономічні процеси

Етапи вирішення задачі

1. Пошук методів моделювання нестационарних процесів та обрання для подальшого розгляду
2. Пошук даних
3. Підготовка даних як задача регресії для обраних моделей
4. Моделювання обраних моделей на кожному з часових рядів
5. Оптимізація гіперпараметрів моделей
6. Порівняльна характеристика результатів моделей

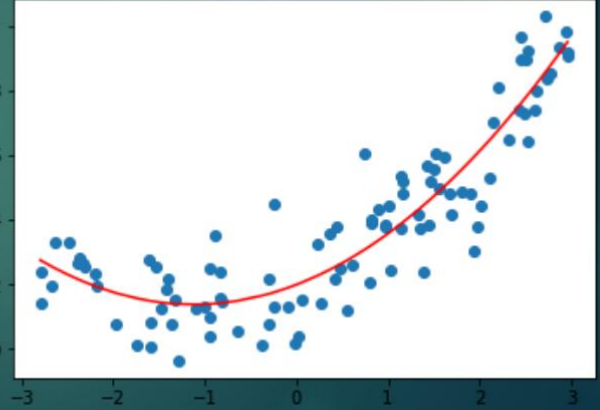
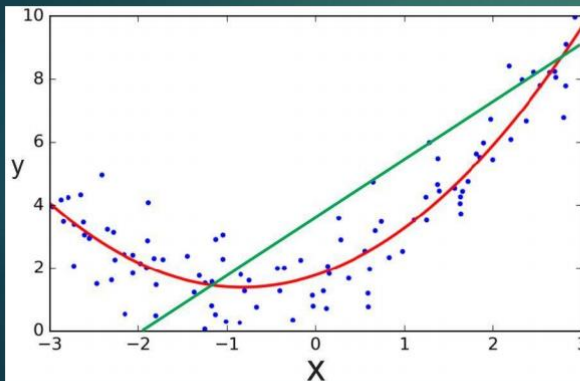
Регресійний аналіз

- Регресійний аналіз – розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої. Займається пошуком моделі зв'язку, вираженої у функції регресії.

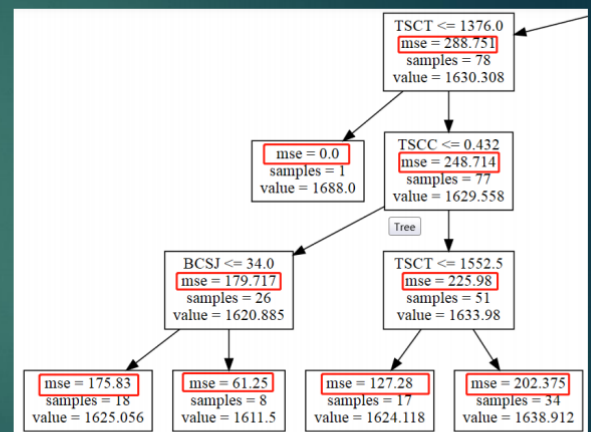
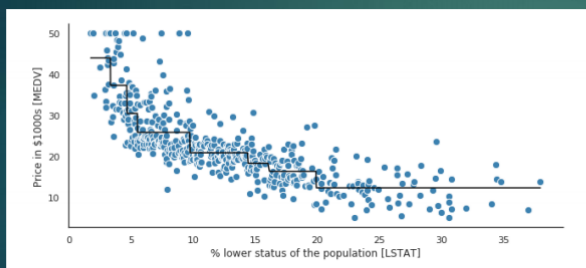


Поліноміальна регресія

$$\triangleright y(k) = a_0 + a_1k + a_2k^2 + \dots a_mk^m + \varepsilon(k)$$



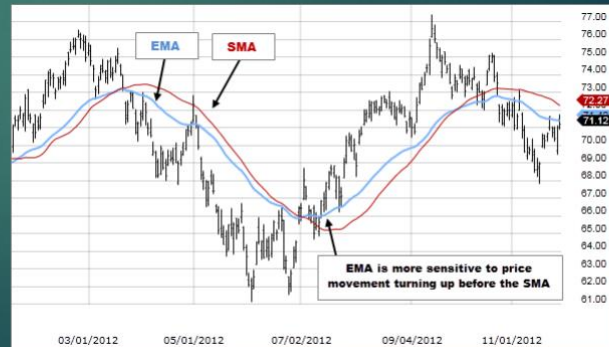
Регресійна модель дерева рішень



Модель прогнозу на основі ковзного середнього

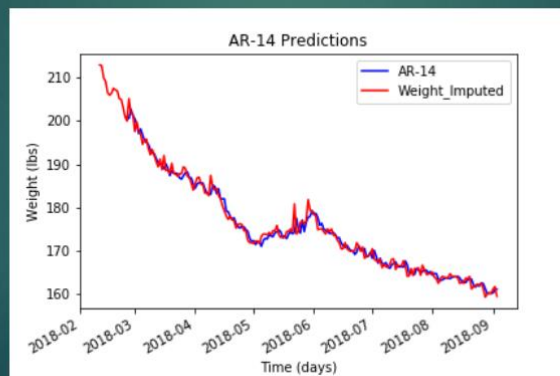
$$MA(k) = \frac{w_1 y(k-1) + w_2 y(k-2) + \dots + w_n y(k-n)}{w_1 + \dots + w_n}$$

n – розмір ковзного вікна, w_i – вагові коефіцієнти, $y(*)$ – часовий ряд

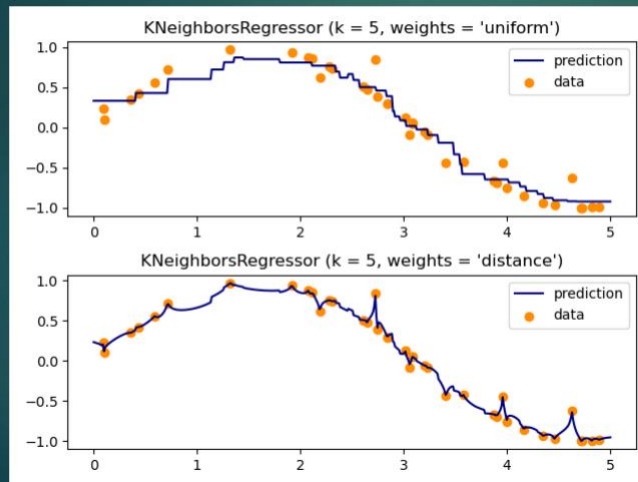


Модель прогнозу на основі авторегресії

$$y(k) = a_0 + a_1 y(k-1) + \dots + a_N y(k-N)$$



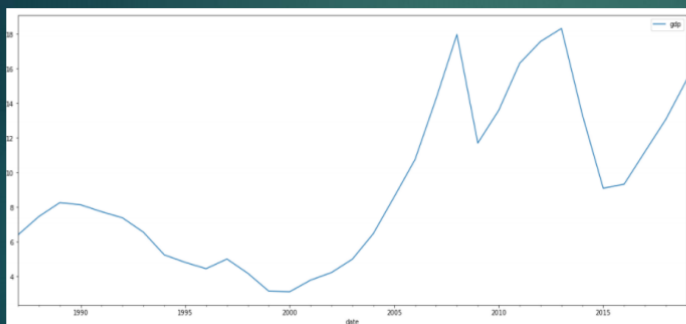
Модель регресії найближчих сусідів



Підготовка даних для прогнозування

Назва змінної	Опис
Value	Значення реальне часового ряду
Y - 1	Лег першого порядку
Y - 2	Лег другого порядку
Y - K	Лег K-того порядку
WINDOW_MEAN_1	Ковзне середнє 1-шого порядку
WINDOW_MEAN_K	Ковзне середнє K-того порядку
WINDOW_STD_1	Ковзна дисперсія 1-того порядку
WINDOW_STD_K	Ковзна дисперсія K-того порядку

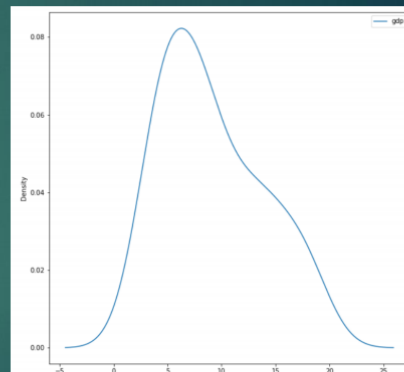
Дослідження та моделювання ВВП України



Часовий ряд

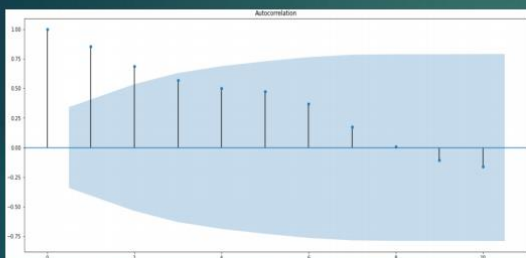
	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	30.00000000	6.59765625	31.92187500	28.06250000	331.25000000	0.11840820	-1.69042969	3.32421875	0.18969727

Описові характеристики

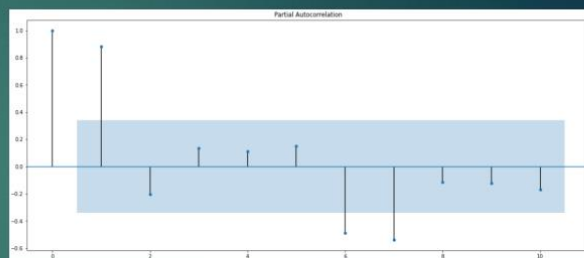


Щільність розподілу

АКФ та ЧАКФ



АКФ для ряду ВВП України в період 1988-2019



ЧАКФ для ряду ВВП України в період 1988-2019

Вид даних

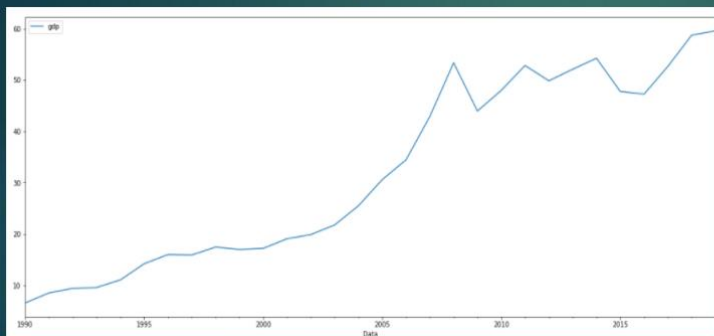
	y-1	y-2	window_mean_2	window_std_2	value
date					
1989-01-01	7.470352	6.408769	7.870634	0.566084	8.270916
1990-01-01	8.270916	7.470352	8.208304	0.088547	8.145692
1991-01-01	8.145692	8.270916	7.946074	0.282302	7.746456
1992-01-01	7.746456	8.145692	7.570340	0.249066	7.394224
1993-01-01	7.394224	7.746456	6.979540	0.586451	6.564856
1994-01-01	6.564856	7.394224	5.909906	0.926240	5.254956
1995-01-01	5.254956	6.564856	5.038171	0.306579	4.821387
1996-01-01	4.821387	5.254956	4.638597	0.258503	4.455808
1997-01-01	4.455808	4.821387	4.735424	0.395437	5.015040
1998-01-01	5.015040	4.455808	4.601682	0.584576	4.188324
1999-01-01	4.188324	5.015040	3.673194	0.728504	3.158064
2000-01-01	3.158064	4.188324	3.142108	0.022565	3.126153
2001-01-01	3.126153	3.158064	3.461691	0.474523	3.797230

Метрики для тренувального та тестового набору даних

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	0.98	12.01	8.029118	0.655776	2.179328
Лінійна Регресія	1	0	0	0	1,745
Найближчі сусіди	0.90	58.236289	12.534785	1.442175	2.014156
Баєсова регресія	1.0	1.66e-15	6.56e-08	7.71e-09	1.739015

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	0.188973	7.037999	10.202293	1.531666	2.669322
Лінійна Регресія	1	0	0	0	0.145
Найближчі сусіди	0.697647	2.623787	6.066802	0.935198	1.481656
Баєсова регресія	1.0	1.26e-16	4.90e-08	6.45e-09	0.130501

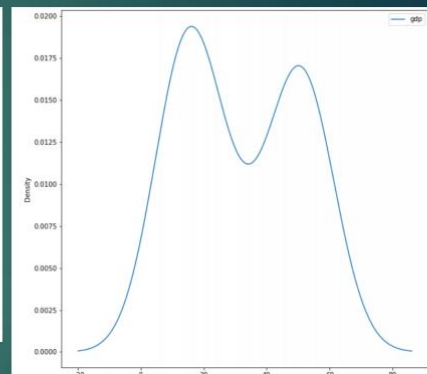
Дослідження та моделювання ВВП Польщі



Часовий ряд

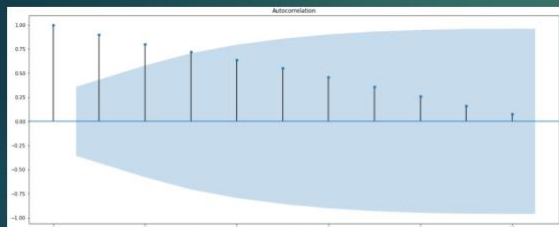
	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	30.00000000	6.59765625	31.92187500	28.06250000	331.25000000	0.11840820	-1.69042969	3.32421875	0.18969727

Описові характеристики

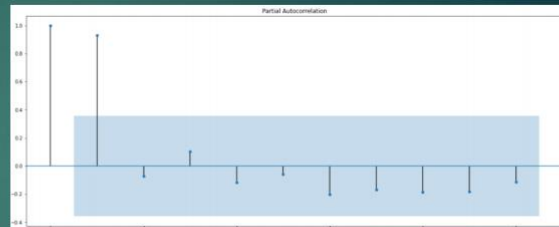


Щільність розподілу

АКФ та ЧАКФ



АКФ для ряду ВВП України в період 1988-2019



ЧАКФ для ряду ВВП України в період 1988-2019

Вид даних

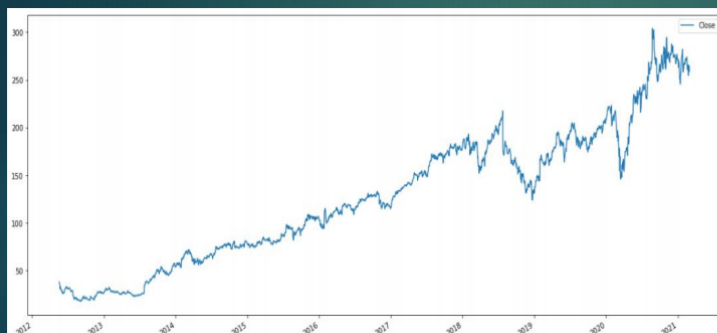
	y-1	y-2	y-3	window_mean_2	window_std_2	window_mean_3	window_std_3	value
Data								
1993-01-01	9.43370507	8.55009359	6.59777490	9.51913479	0.12081586	9.19612106	0.56596099	9.60456450
1994-01-01	9.60456450	9.43370507	8.55009359	10.34245183	1.04353026	10.03953624	0.90540116	11.08033915
1995-01-01	11.08033915	9.60456450	9.43370507	12.65480875	2.22663626	11.63806067	2.36226258	14.22927835
1996-01-01	14.22927835	11.08033915	9.60456450	15.12430128	1.26575357	13.77631390	2.50045515	16.01932421
1997-01-01	16.01932421	14.22927835	11.08033915	15.97755159	0.05907540	15.39479385	1.01023004	15.93577898
1998-01-01	15.93577898	16.01932421	14.22927835	16.70217907	1.08385340	16.47456078	0.86185808	17.46857916
1999-01-01	17.46857916	15.93577898	16.01932421	17.23583983	0.32914311	16.80248621	0.78584583	17.00310050
2000-01-01	17.00310050	17.46857916	15.93577898	17.11252331	0.15474722	17.23120859	0.23287752	17.22194611
2001-01-01	17.22194611	17.00310050	17.46857916	16.15624773	1.32130202	17.77186532	1.14724410	19.09054935
2002-01-01	19.09054935	17.22194611	17.00310050	19.49887762	0.57746337	18.73990045	1.37654313	19.90720588
2003-01-01	19.90720588	19.09054935	17.22194611	20.84496598	1.32619305	20.26016044	1.38035741	21.78272608
2004-01-01	21.78272608	19.90720588	19.09054935	23.64687212	2.63630061	22.40031671	2.85249744	25.51101815
2005-01-01	25.51101815	21.78272608	19.90720588	26.06272589	3.60865969	25.96939262	4.43366044	30.61443363
2006-01-01	30.61443363	25.51101815	21.78272608	32.53831697	2.72078191	30.19588403	4.49024535	34.46220031
2007-01-01	34.46220031	30.61443363	25.51101815	38.68252542	5.96844101	35.99316149	6.28563276	42.90285054
2008-01-01	42.90285054	34.46220031	30.61443363	46.13187936	7.39496348	43.57531968	9.46728318	53.36090819
2009-01-01	53.36090819	42.90285054	34.46220031	46.66732951	6.63772261	46.74583652	5.75378890	43.93735084

Метрики для тренувального та тестового набору даних

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	0.98467331	91.90670008	5.43217798	1.95689699	2.24260626
Лінійна Регресія	1.000	0.00000	0.0000	0.0000	0.36
Найближчі сусіди	1.000	0.00000000	0.00000000	0.00000000	nan
Баєсова регресія	1.00000000	0.00000000	0.00000639	0.00000278	1.32664848

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	-6.39	211.72856462	12.12190704	8.40096353	0.54682912
Лінійна Регресія	1.00000000	0.00000000	0.00000000	0.000002	0.26279
Найближчі сусіди	-3.17	119.62988476	9.07353818	6.31479440	0.49090345
Баєсова регресія	1.00000000	0.00000000	0.00000392	0.00000287	0.71595828

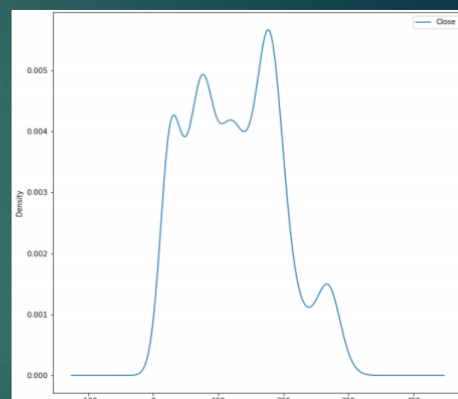
Дослідження та моделювання курсу акцій Facebook



Часовий ряд

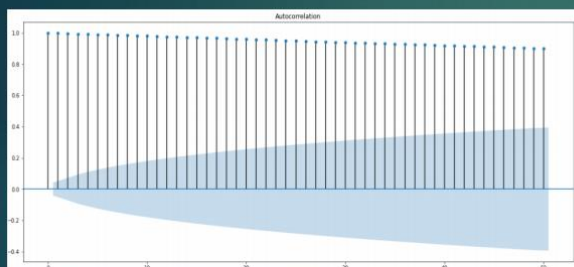
	observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
0	2210.00000000	26.84375000	126.68750000	123.93750000	4912.00000000	0.27124023	-0.75097656	79.12500000	0.00000000

Описові характеристики

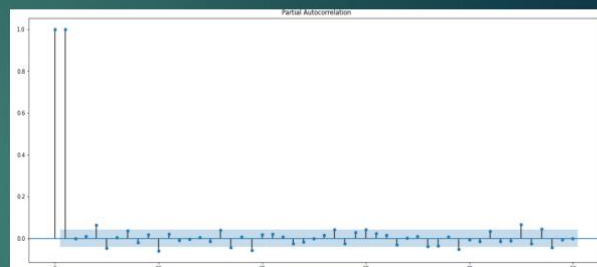


Щільність розподілу

АКФ та ЧАКФ



АКФ для ряду ВВП України в період 1988-2019



ЧАКФ для ряду ВВП України в період 1988-2019

Вигляд даних

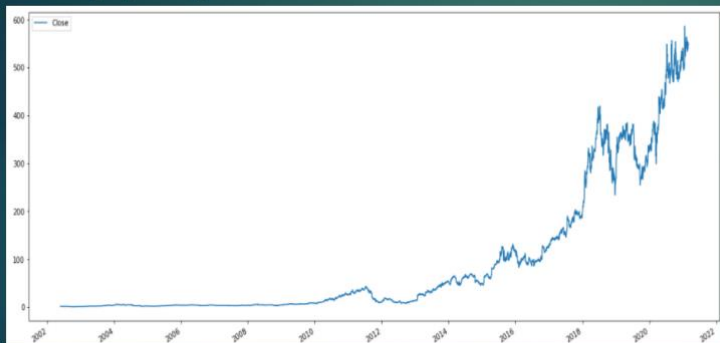
Date	y-1	y-2	y-3	y-4	y-5	y-6	y-7	y-8	y-9	y-10	window_mean_2	window_std_2	window_mean_3	window_std_3	window_mean_4	window_std_4	value
2012-06-04	27.71999931	29.60000036	28.19000053	28.84000015	31.90999985	33.02999878	32.00000000	31.00000000	34.02999878	36.22999954	27.30999947	0.57982734	28.07333310	1.38424514	28.10249996	1.13173579	26.89999962
2012-06-05	26.89999962	27.71999931	29.60000036	28.19000053	28.84000015	31.90999985	33.02999878	32.00000000	31.00000000	34.02999878	26.38500023	0.72831912	26.82999992	0.92689358	27.52250004	1.57831915	25.87000084
2012-06-06	25.87000084	26.89999962	27.71999931	29.60000036	28.19000053	28.84000015	31.90999985	33.02999878	32.00000000	31.00000000	26.34000015	0.66467940	26.52666664	0.57046690	26.82499961	0.75694498	26.80999947
2012-06-07	26.80999947	25.87000084	26.89999962	27.71999931	29.60000036	28.19000053	28.84000015	31.90999985	33.02999878	32.00000000	26.55999947	0.35355339	26.32999992	0.47031837	26.47249965	0.47821664	26.30999947
2012-06-08	26.30999947	26.80999947	25.87000084	26.89999962	27.71999931	29.60000036	28.19000053	28.84000015	31.90999985	33.02999878	26.70499992	0.55861500	26.73999977	0.39962524	26.52250004	0.54377516	27.10000038
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2021-02-24	265.83998535	260.32998657	261.53999756	269.39001465	273.57000732	273.97000122	270.50000000	270.39001465	271.86999512	269.45001221	265.08499146	1.09600688	263.49998983	2.85259636	263.07499176	2.52304641	264.30999756
2021-02-25	264.30999756	265.83998535	260.32998657	261.53999756	269.39001465	273.57000732	273.97000122	270.50000000	270.39001465	271.86999512	259.50000000	6.80236378	261.61999512	6.05138120	261.29749296	4.98285462	254.69000244
2021-02-26	254.69000244	264.30999756	265.83998535	260.32998657	261.53999756	269.39001465	273.57000732	273.97000122	270.50000000	270.39001465	256.15499878	2.07181769	258.87333171	4.93094473	260.61999512	5.33036674	257.61999512
2021-03-01	257.61999512	254.69000244	264.30999756	265.83998535	260.32998657	261.53999756	269.39001465	273.57000732	273.97000122	270.50000000	261.26499939	5.15481446	259.07333374	5.26272278	260.38249969	5.03188130	264.91000366
2021-03-02	264.91000366	257.61999512	254.69000244	264.30999756	265.83998535	260.32998657	261.53999756	269.39001465	273.57000732	273.97000122	261.95500183	4.17900367	260.50999959	3.87248373	259.05500031	4.29715160	259.00000000

Метрики для тренувального та тестового набору даних

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	0.99883598	7071.25227865	1.81102705	1.92917385	1.46768325
Лінійна Регресія	1.00000000	0.00000000	0.00000000	0.00000000	0.00117067
Найближчі сусіди	1.00000000	0.00000000	0.00000007	0.00000077	2.04342649
Баєсова регресія	0.99999987	0.81343660	0.03122976	0.02069117	0.00053057

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	-0.6765	14.9350	48.27990588	0.01546698	0.10951891
Лінійна Регресія	1.00000000	0.00000000	0.00000000	0.00000000	0.00117067
Найближчі сусіди	-0.255	523490.018	12.5606	41.77279	0.02244335
Баєсова регресія	0.999998	0.716531	0.01925711	0.04887179	0.00060406

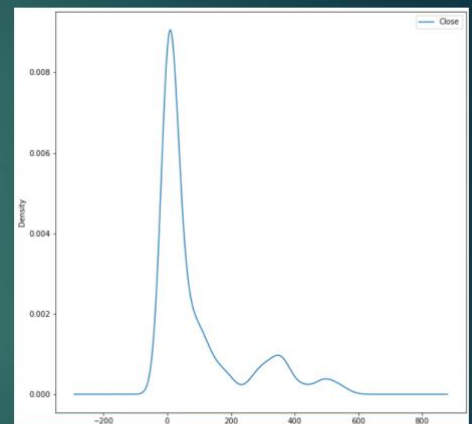
Дослідження та моделювання курсу акцій Facebook



Часовий ряд

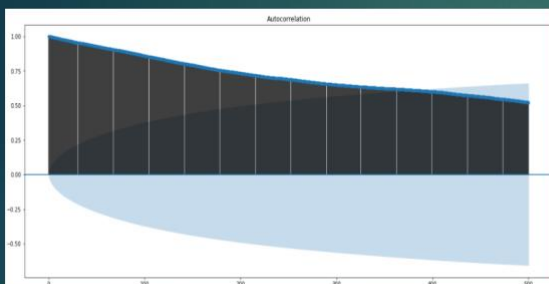
observations	mode	mean	median	variation	skewness	kurtosis	jarque-bera	probability
4728.00000000	1.60742188	91.43750000	16.98437500	19248.00000000	1.70605469	1.78710938	2916.00000000	0.00000000

Описові характеристики

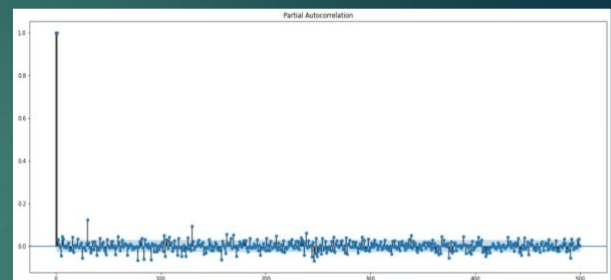


Щільність розподілу

АКФ та ЧАКФ



АКФ для ряду ВВП України в період 1988-2019



ЧАКФ для ряду ВВП України в період 1988-2019

Відповідно було обрано лаг порядку 50 та ковзні статистики порядку 4.

Перейдемо до етапу моделювання моделей.

Вигляд даних

Date	y-1	y-2	y-3	y-4	y-5	y-6	y-7	y-8	y-9	y-10	...	y-48	y-49	y-50	window_mean_2	window_std_2	window_mean_3	window_std_3	window_mean_4	window_std_4
2002-08-05	0.77785701	0.89142901	0.95357102	0.91428602	0.87642902	0.87142903	1.03642905	1.11428595	1.14642894	1.17071402	...	1.15714300	1.21000004	1.19642901	0.77642849	0.00202022	0.81476200	0.06641094	0.84946425	0.08807532
2002-08-06	0.77499998	0.77785701	0.89142901	0.95357102	0.91428602	0.87642902	0.87142903	1.03642905	1.11428595	1.14642894	...	1.10357106	1.15714300	1.21000004	0.76785699	0.01010171	0.77119033	0.00918454	0.80125000	0.06058525
2002-08-07	0.76071399	0.77499998	0.77785701	0.89142901	0.95357102	0.91428602	0.87642902	0.87142903	1.03642905	1.11428595	...	1.07142901	1.10357106	1.15714300	0.77714249	0.02323341	0.77642832	0.01647500	0.77678549	0.01347074
2002-08-08	0.79357100	0.76071399	0.77499998	0.77785701	0.89142901	0.95357102	0.91428602	0.87642902	0.87142903	1.03642905	...	1.07642901	1.07142901	1.10357106	0.79499999	0.00202090	0.78357132	0.01984654	0.78142849	0.01676177
2002-08-09	0.79642898	0.79357100	0.76071399	0.77499998	0.77785701	0.89142901	0.95357102	0.91428602	0.87642902	0.87142903	...	1.12857103	1.07642901	1.07142901	0.79178599	0.00656618	0.79238099	0.00475599	0.78446424	0.01630274
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2021-02-24	546.15002441	533.78002930	540.21997070	548.21997070	551.34002686	557.28002930	556.52001953	557.59002686	563.59002686	559.07000732	...	522.41998291	503.22000122	501.08999634	548.77999878	5.13355888	544.44667562	9.82520638	543.38999939	8.37492725
2021-02-25	553.40997314	546.15002441	533.78002930	540.21997070	548.21997070	551.34002686	557.28002930	556.52001953	557.59002686	563.59002686	...	519.78002930	522.41998291	503.22000122	550.05499268	4.74465888	548.75333659	4.04213058	545.01000977	8.18184143
2021-02-26	546.70001221	553.40997314	546.15002441	533.78002930	540.21997070	548.21997070	551.34002686	557.28002930	556.52001953	557.59002686	...	524.83001709	519.78002930	522.41998291	542.77499390	5.55081413	546.31999898	7.28743416	546.27749634	5.95077190
2021-03-01	538.84997559	546.70001221	553.40997314	546.15002441	533.78002930	540.21997070	548.21997070	551.34002686	557.28002930	556.52001953	...	532.90002441	524.83001709	519.78002930	544.74499512	8.33681657	545.39666748	6.00210679	547.39999390	6.33009670
2021-03-02	550.64001465	538.84997559	546.70001221	553.40997314	546.15002441	533.78002930	540.21997070	548.21997070	551.34002686	557.28002930	...	534.45001221	532.90002441	524.83001709	549.23001099	1.99404630	545.76999919	6.15653585	546.00250244	5.04826715

4676 rows × 57 columns

Метрики для тренувального набору даних

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	0.99996402	1674.09224558	1.42013956	0.61851614	2.09437071
Лінійна Регресія	1.00000000	0.00000000	0.00000000	0.00000000	0.56890375
Найближчі сусіди	1.00000000	0.00000000	0.00000004	0.00000043	1.84689228
Баєсова регресія	1.00000000	0.00801821	0.01578874	0.00135363	0.00027530

Назва моделі	R2	RSS	MAPE	RMSE	Durbin_Wanton
Дерево Рішень	-0.16228630	1755970.37115921	11.78150592	76.50643483	0.02908394
Лінійна Регресія	1.00000000	0.00000000	0.00000000	0.00000000	0.31551639
Найближчі сусіди	-2.08733658	4664316.86228141	20.96599377	124.69050836	0.01363245
Баєсова регресія	0.99999999	0.00806394	0.00111288	0.00518457	0.00023816

Висновки

Розглянувши чотири часові ряди, які є не стаціонарними процесами, два з яких макроекономічні та два фінансовоекономічні. Використовуючи чотири моделі для того щоб спрогнозувати тенденцію процесів можемо зробити висновки, що найкращим з варіантів виявилася багатовимірна лінійна регресія. З іншого боку не сильно їй поступається модель Баєсівської регресії, яка має також хороші результати як на тренувальному так і на тестовому наборі. Моделі дерева рішень та метод найближчих сусідів виявилися слабкими для прогнозу нестационарних ряді, хоча показували гарні результати на тренувальному наборі на тестовому дали жахливі результати.