Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

# Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: **«Комп'ютерна модель аналізу індексів акцій на фондовому ринку»**

Захищено на засіданні
Атестаційної комісії № 38
протокол № __ від __.12.2021 р.
Оцінка _____ / _____
Голова Атестаційної комісії
д.т.н., професор
_____ Мінухін Сергій Володимирович

**Виконав:**
Галузь знань: 12 – Інформаційні технології
за спеціальністю 123 – Комп'ютерна інженерія.

Анжуров Валентин Євгенович _____

**Керівник:**
д.т.н., с.н.с., професор кафедри теоретичної та прикладної системотехніки
Толстолузька О.Г. _____

**Рецензент:**
д.т.н., доцент, завідувач кафедри безпеки інформаційних систем і технологій
Рассомахін С.Г. _____

Харків – 2021

# АНОТАЦІЯ

Кваліфікаційна робота складається зі вступу, трьох розділів, висновку, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 63 сторінки, з яких 46 сторінок основної частини що містять 22 рисунки, 23 найменування списку використаних джерел на 3 сторінках і 4 додатки на 10 сторінках.

Об'єкт дослідження: процес кластеризації даних компаній на фінансовій біржі методом k-середніх

Предмет дослідження: комп'ютерні моделі та методи препроцесінгу в Data Mining

Мета: підвищення ефективності аналізу індексів акцій на фондовому ринку за рахунок використання комп'ютерної моделі кластеризації даних компаній на фінансовій біржі методом k-середніх

Методи дослідження: процес кластеризації даних методом k-середніх

Результати дослідження: комп'ютерна модель в вигляді програмного продукту, що має простий інтерфейс та оптимальний час роботи

В ході даної роботи були розглянуті концепції Data Mining, що стосуються препроцесінгу, а саме його типу – кластеризації. Були розглянуті та досліджені деякі моменти написання моделі препроцесінгу Під час написання моделі були застосовані сучасні технології. В роботі наведені порівняння можливих шляхів побудови даної моделі та дослідження стосовного кожного з них.

**Ключові слова**: Data Mining, пепроцесінг, кластеризація, метод k-середніх, комп'ютерна модель, Python.

# ABSTRACT

The qualifying work consists of an introduction, three sections, a conclusion, a list of sources used and four appendices. The total volume of the work is 63 pages, of which 46 pages of the main part contain 22 figures, 23 names of the list of used sources on 3 pages and 4 appendices on 10 pages.

Object of research: the process of clustering these companies on the financial exchange by the method of k-means

Subject of research: computer models and methods of preprocessing in Data Mining

Objective: to increase the efficiency of the analysis of stock indexes in the stock market through the use of a computer model of clustering of data of companies on the financial exchange by the method of k-means

Research methods: the process of clustering data by k-means

The results of the study: a computer model in the form of a software product that has a simple interface and optimal operating time

In the course of this work, the concepts of Data Mining related to preprocessing, namely its type - clustering, were considered. Some aspects of writing a preprocessing model were considered and investigated. Modern technologies were used during the writing of the model. The paper compares possible ways of constructing this model and studies relevant to each of them.

**Keywords**: Data Mining, reprocessing, clustering, k-means method, computer model, Python.

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application programming interface – Програмний інтерфейс аплікації

ОС – операційна система

БД – база даних

PCA – Principal component analysis – Аналіз головних компонентів

HTTP – Hypertext transfer protocol – Протокол передачі гіпертексту

HTTPS - Hypertext transfer protocol secure – Захищений протокол передачі гіпертексту

IDEF0 – Integrated Definition for Function Modeling – Інтегроване визначення для моделювання функцій

# ВСТУП

Кожна людина шукає шляхи зберегти/збільшити її зароблений тим чи іншим чином фінансовий капітал. Сьогодні тема фондового ринку та його цінних паперів має актуальний характер, тому що це є один з найстаріших засобів інвестування. Фондовий ринок як явище з'явився в Америці в 1970 році, що є роком виникнення першої фондової біржі. Компанії зрозуміли, що капітал можна надбати не лише шляхом продажу власної продукції, а також шляхом вкладів звичайних людей в бізнес компаній. Таким чином, ці люди ставали співвласниками мізерної частини бізнесу та мали відповідний заробіток з цього.

Питання інвестицій вільного капіталу сьогодні хвилює велику кількість людей в багатьох частинах світу. Найцікавіші з точки зору звичайного інвестора перспективи інвестування:

- Збереження та зрощення капіталу, що існує
- Захист від інфляції
- Вихід на ранню пенсію шляхом забезпечення власних потреб за рахунок бонусів, що виплачують компанії своїм вкладникам
- Додатковий прибуток до заробітку від основної діяльності

Враховуючі ці аспекти, можна стверджувати, що інвестиції та дослідження, де інвестиції є основною моделлю, є дуже актуальними в сьогодення.

З причини того, що інформація про акції фондового ринку занадто велика та різноманітна, потрібно вміти швидко і зручно орієнтуватися в ній. Для цього, безумовно, підходить така галузь, як Data Science і, зокрема, її підгалузь Data Preprocessing. Розглядаючи Data Preprocessing, можна виділити такий метод, як кластеризація даних, де вхідні дані групуються в кластери (групи) не за класифікаційною ознакою, а за ознакою належності до певної групи.

Для досягнення успіху в інвестуванні, потрібно мати змогу виявляти більш перспективні для інвестування компанії, де можна було б отримати найбільший прибуток. Саме цю проблему і вирішує комп'ютерна модель, що використовує метод кластеризації k-середніх для групування кластеру, що містить найбільш перспективні компанії. Також, система може проводити кластеризаційний аналіз попередніх років, так як в реальному світі інвестицій потрібно мати змогу аналізувати минулі ситуації на фондовому ринку, щоб розуміти, як діяти сьогодні.

Комп'ютерні моделі мають потужність спроможну відтворювати складні обчислення. Все це марно, коли людина, що використовує цю систему для написання програми, не вміє користуватися цією потужністю. Велика кількість систем працює невдало та повільно саме через це, результатом чого є нестабільний продукт, з яким неможливо працювати. Тому необхідно вміти використовувати ресурси, що видаються машиною для забезпечення швидкої роботи комп'ютерної моделі. Задача розробника полягає в написанні ефективного коду, що оптимально використовує ресурси системи.

На щастя, сучасні методи препроцесінгу та кластеризації працюють таким чином, що їх достатньо впевнено можна використовувати у високонавантажених системах. Тим не менш, потрібно вивчати та досліджувати нові методи роботи з ними задля досягнення більш ефективних результатів у майбутньому.

Робота включає у себе дослідження можливого використання кластеризації та методу k-середніх для аналізу минулого та поточного станів фондового ринку з метою рекомендації покупки акцій з найбільшою вірогідністю росту у майбутньому

**Мета дослідження** – підвищення ефективності аналізу індексів акцій на фондовому ринку за рахунок використання комп'ютерної моделі кластеризації даних компаній на фінансовій біржі методом k-середніх

**Завдання дослідження:**

- дослідити методи препроцесінгу в Data Mining;
- дослідити методи кластеризації в Data Mining;
- розробити комп'ютерну модель аналізу;
- скласти інструкцію для користування.

**Об'єктом дослідження** є процес кластеризації даних компаній на фінансовій біржі методом k-середніх

**Предметом дослідження** є комп'ютерні моделі та методи препроцесінгу в Data Mining

**В роботі виконані наступні завдання:**

1. Проаналізовано існуючі методи препроцесінгу даних у Data Mining.
2. Досліджені формати вхідних та вихідних даних, інтерфейсу користувача.
3. Розроблено комп'ютерну модель аналізу даних фондового ринку
4. Розроблено інструкцію для користування.

# РОЗДІЛ 1.
# DATA MINING ТА КЛАСТЕРИЗАЦІЯ

Дані в комп'ютерних моделях збираються в різних форматах та станах. Можливі втрачені дані, можливі некоректні дані або ж взагалі «шум». Усі подальші етапи обробки інформації залежать від того, наскільки якісно був проведений препроцесінг зібраної інформації.

## 1.1 Data Mining

Data mining (з англ. видобуток даних) - це процес напівавтоматичного аналізу великих баз даних з метою пошуку корисних фактів. Зазвичай, поділяють на задачі класифікації, моделювання та прогнозування [**Error! Reference source not found.**]. На сучасних підприємствах, в дослідницьких проектах або в інтернеті утворюються великі обсяги даних. Глибинний аналіз даних здійснюється автоматично шляхом застосування методів математичної статистики, штучних нейронних мереж, теорії нечітких множин або генетичних алгоритмів. Метою аналізу є виявлення правил та закономірностей, наприклад, статистичних подій. Так, наприклад, можливо виявити зміни у поведінці клієнтів або груп клієнтів для покращення стратегії підприємства.

До Data Mining відносять наступні поняття:

1. Безпосередня обробка великого обсягу даних.
2. Здатність алгоритму до виявлення патерів патернів у даних. При обробці утворюються групи даних, що схожі за відтінками або іншими ознаками.
3. Процесінг може робити прогнози щодо фінального результату. Під час обробки алгоритм може на певному етапі зробити прогноз результату останнього етапу, орієнтуючись на вже оброблену інформацію всіх етапах.

4. Знаходження потрібної інформації. Збір даних алгоритмами виявляє ознаки даних і з цього користувач отримує найбільш релевантну інформацію щодо свого запиту.

Не зважаючи на те, що збір статистики та процес Data Mining виглядають однаковими, можна побачити різні властивості між ними:

1. Data Mining є незалежним процесом від стороннього втручання людини та не вимагає її присутності взагалі.
2. Неможливо порівнювати об'єм даних, що опрацьовує статистичний метод та Data Mining в силу того, що у порівнянні в Data Mining, об'єм даних для метода дуже малий.

Data Mining процес ділиться на 4 етапи:

1. По-перше, визначається основна проблема, яку належить розв'язати процесу.
2. Відбувається збір на попередня підготовка даних для подальшого аналізу. Необхідно обрати найбільш ефективний спосіб збору об'єму даних. Далі зібрані дані передаються до етапу попередньої обробки, де відсікається шум, заповнюються втрачені частини, відбувається фінальна компресія.
3. Будується модель та проводиться оцінка даних. Модель має бути використана для машинної обробки кінцевих даних та отримання певного результату.
4. Фінальні дані мають бути використані для вирішення раніше поставленої проблеми. На їх основі будується висновок щодо роботи моделі в цілому, а також на цьому етапі стає зрозуміло чи вдалося вирішити поставлену задачу.

Необхідно зазначити те, що Data Mining є лише методологічними рекомендаціями щодо збору та обробки великої кількості даних, а не готовим

програмним рішенням. Людина, що використовує Data Mining, має чітко оперувати правилами та поняттями методології, щоб отримати найбільш сприятливий результат та вирішити задачу.

### 1.2 Препроцесінг

Препроцесінг даних – метод збирання інформації, що використовується для конвертування швидко зібраних даних у правильний формат, що може бути ефективно використаним у наступних етапах процесу Data Mining. Препроцесінг складається з наступних кроків [**Error! Reference source not found.**]:

1) Спочатку дані підлягають очищенню. Інформація може мати невідповідності та відсутні частини. Для обробки цієї інформації потрібно замінити шум, втрачені дані, тощо:
    a. Шумні дані – дані, які не відносяться до безпосереднього предмету дослідження або ж не несуть ніякого змістовного навантаження;
    b. Втрачені дані – дані, що не були зібрані з тієї або іншою причини. Як правило заповнюються у відповідності до сусідніх даних.
2) Перетворення інформації. Цей етап полягає в перетворенні очищених даних у потрібний для дослідника формат. Етап передбачає наступні способи:
    a. Дискретизація – заміна необроблених ділянок даних числовими атрибутами;
    b. Покоління ієрархії концепції – атрибути мають бути перетворені за поточного рівня на рівень віще в ієрархії;
    c. Нормалізація – робиться для масштабування значення даних у визначеному діапазоні;

d. Вибірковий метод атрибутів – стратегія будування нових атрибутів із заданого набору;

e. Кластеризація – процедура будування групи даних за схожими ознаками.

3) Компресія даних [**Error! Reference source not found.**]. Зібрані дані, як правило, представлені дуже великим об'ємом, що безумовно робить процес дослідження більш трудомістким, та не завжди ці часові витрати виправдовуються результатом. Спосіб компресії передбачає зменшення об'єму даних із зберіганням цінних властивостей для дослідника. Нижче наведені деякі способи компресії даних:

a. Зменшення чисельності – передбачає зберігання моделі даних замість цілісних даних;

b. Вибирання атрибутної підмножини. Використовувати потрібно лише актуальні дані. Для проведенні вибору атрибутів потрібно використати рівень значущості та p-значення атрибутів. Атрибут, що має p-значення більше за рівень значущості, може бути відкинутим із зібраного об'єму даних;

c. Кубічна агрегація інформації – застосовується до інформації для побудування куба даних;

d. Зменшення розмірності – проводиться шляхом зменшення за допомогою механізмів кодування. Ці механізми можуть бути втратними або безвтратним і, якщо після реконструкції зі зжатої інформації можуть бути отримані вхідні дані, тоді таке зменшення називають безвтратним. Інакше воно вважається втратним. Два найбільш популярні методи зменшення розмірності – перетворення вейвлетів та PCA (аналіз основного компоненту).

### 1.3 Кластеризація

Не можна недооцінити значення кластеризація при роботі с препроцесінгом даних в Data Mining. Утворення груп (кластерів) інформація за схожими ознаками широко застосовується в багатьох галузях, де потрібна обробка великої кількості даних [**Error! Reference source not found.**].

Кластеризація – процес утворення груп з певного набору об'єктів, де групування базується на порівнянні за ознаками, характеристиками тощо. В Data Mining методології розглядаються дані, які реалізують перший алгоритм поєднання, що найбільш підходить до аналізу поточних даних.

Кластеризація має особливість, що об'єкт не має бути належним до того чи іншого кластеру, викликаючи цей тип жорсткого угруповання секціонування. Тим не менш, м'який поділ вказує на те, що кожен об'єкт належним чином належить кластеру. Більш детальний розгляд вказує на те, що можуть бути створені об'єкти з кількома кластерами, щоб змусити об'єкт брати участь лише в одному кластері, і навіть будувати ієрархічні дерева на основі відносин між групами.

Існують наступні доводи того, що кластеризація є ефективним етапом препроцесінгу:

1) Кластеризація має функціонал для роботи з різними типами вхідних об'єктів. Алгоритми мають змогу обробляти будь-який тип даних. Наприклад, можуть бути застосовані до числового діапазону, двійкових даних або категоріальних даних.

2) Ідентифікація груп за формою атрибутів. Процес кластеризації повинен бути універсальним, коли мова іде про довільну ідентифікацію кластеру. Цей процес обмежений лише вимірюванням відстані від значення атрибуту до центроїда кластера. Дана відстань має тенденцію бути виявленою невеликим сферичним кластером [**Error! Reference source not found.**].

3) Кластеризація може обробляти неочищені від шуму на попередніх етапах дані. Бази с інформацією можуть мати галасливі або хибні дані. Інші алгоритми можуть бути чутливими до таких даних. Результатом їх роботи є низькоякісні кластери.

4) Масштабованість. Для роботи з великим обсягом інформації необхідні високо масштабні методи кластерного аналізу.

5) Висока розмірність. Методи кластеризації можуть обробляти як інформацію за малою розмірністю, так дані з високим простором розмірності.

## 1.4 Метод k-середніх

### 1.4.1 Алгоритм

Метод k-середніх – ітеративний алгоритм, що переслідує мету розділення об'єму вхідних даних на окремі кластери, кожен з яких об'єднує об'єкті, схожі за характеристиками. Метод намагається розбити дані на максимально великі за кількістю об'єктів групи шляхом розрахунку квадратів відстаней між значеннями об'єктів та значеннями центроїдів кластерів. Відстань дорівнює середньому арифметичному всіх точок даних, що належить кластеру. Однорідність даних в кластері можна досягнути шляхом збирання даних для цього кластера, які мають однорідні характеристики [**Error! Reference source not found.**].

Алгоритм метода k-середніх описаний нижче:

1) Визначається кількість кластерів.

2) Обираються випадкові значення для кожного кластера.

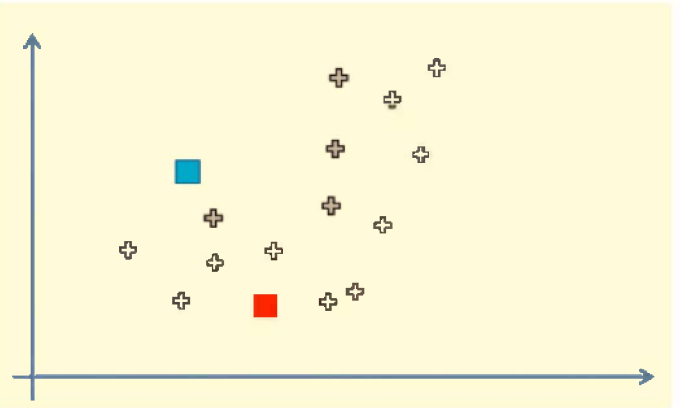


Рисунок 1.1 – випадкові центроїди

3) Кожне значення вхідних даних співвідноситься з центроїдом, до якого квадратична відстань є найменшою.

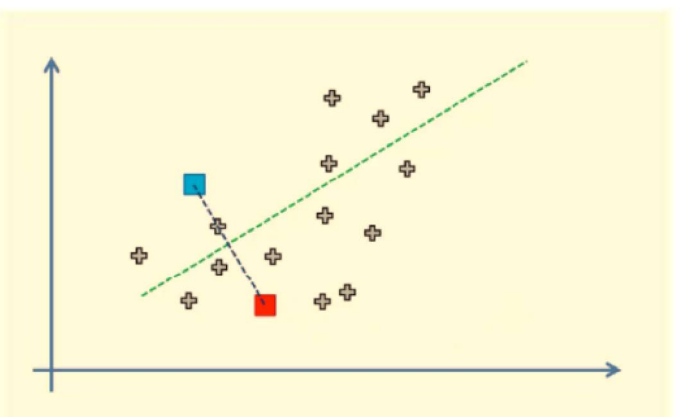


Рисунок 1.2 – формування початкових кластерів

4) Значення центроїдів перераховуються.

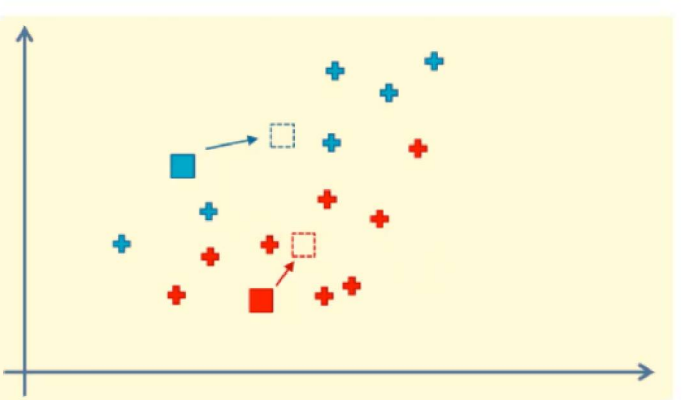


Рисунок 1.3 – перераховані центроїди

5) Відбувається перевизначення кластера для кожної точки з урахуванням нових значень кластерів.

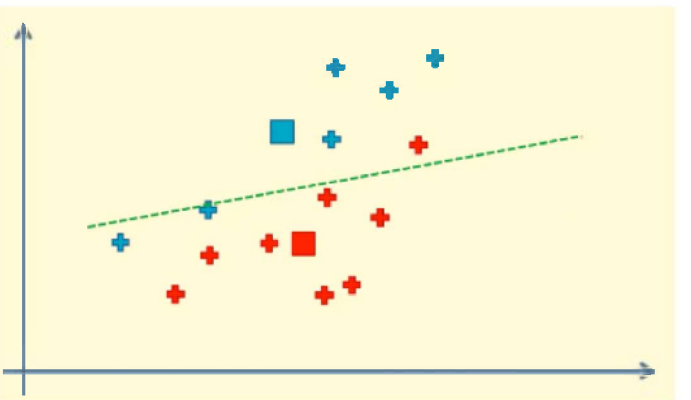


Рисунок 1.4 – перерозподілені точки

6) Кроки 4-5 повторюються, доки перераховані значення центроїдів не зійдуться із попередніми.

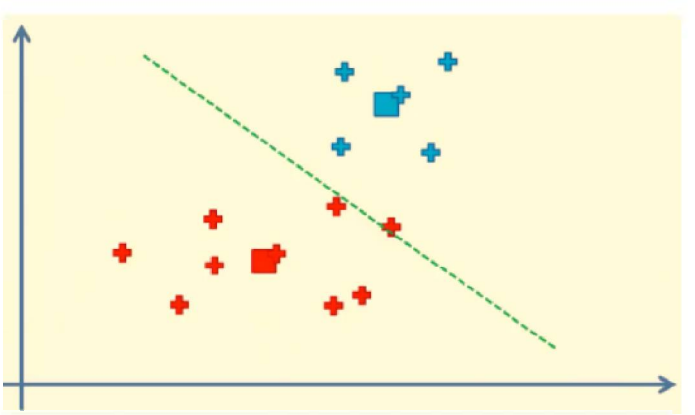


Рисунок 1.5 – результат роботи методу

### 1.4.2 Переваги та недоліки методу

Взявши в роботу будь-який метод або алгоритм кластеризації, можна зрозуміти що він має переваги та недоліки у порівнянні з іншими. Потрібно розуміти, що кожна проблема має певні характеристики, і кожен з методів може бути більш менш ефективним у різних обставинах. Щоб вирішити проблему цієї роботи, було визначено, що метод k-середніх є рішенням, що відповідає поточним обставинам [**Error! Reference source not found.**].

Переваги методу:

1. Спроможність роботи з великим об’ємом інформації.
2. Простий у реалізації.
3. Гарантує конвергенцію – близьке розташування даних за схожими ознаками чи характеристиками.
4. Має здатність узагальнюватись до скупчень різного розміру та форм, таких як еліптичні скупчення. Як веде себе метод, коли кластери мають різний розмір та щільність, зображено на Рис. 1.6. З лівого боку знаходяться інтуїтивні кластери, з правого ті, що є результатом методу k-середніх. Це є доказом того факту, що метод не є орієнтовним на різноманітність форм, а на співвідношення до центроїдів.

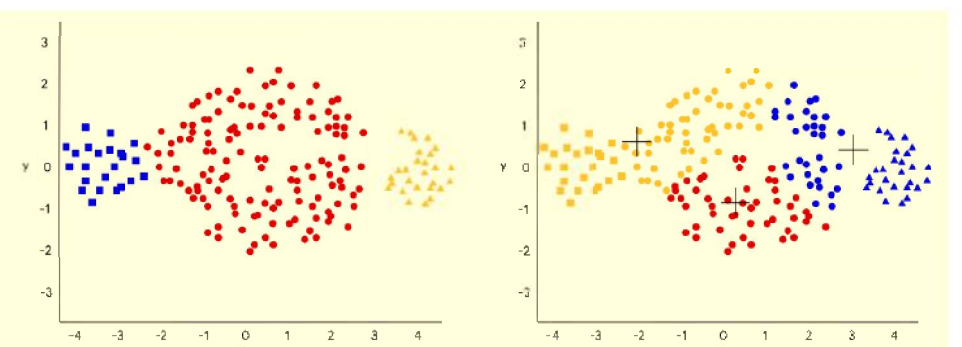


Рисунок 1.6 – Інтуїтивні кластери та кластери із методу k-середніх

5. Легко адаптивний до нових видів вхідних даних. Робота метода не залежить від того якої складності дані він оброблює. Основна задача – сформувати чіткі кластери.
6. Алгоритм піддається модифікації. Для проведення кластеризації кластерів, таких як на Рис. 1.6, можна адаптувати метод k-середніх. На Рис. 1.7 чорна лінія зображує межі кластеру після роботи методу. З лівого боку рисунка

зображена відсутня узагальненість. Це призводить до неінтуїтивної межі кластера. На центральній частині зображене отримання інтуїтивного кластеру різної величини. На правій – результатом є еліптичний кластер

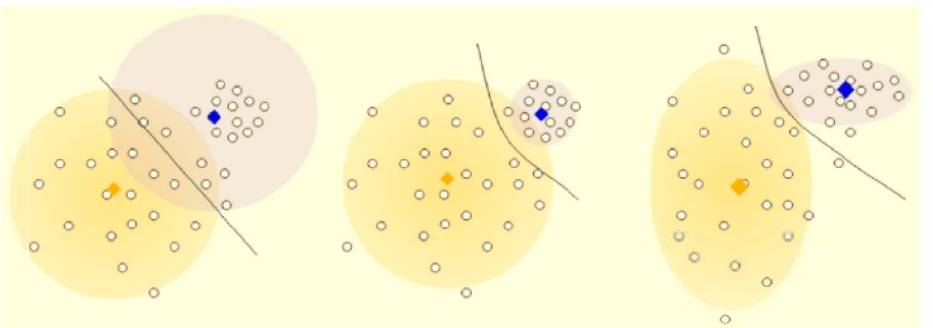


Рисунок 1.7 – Результат адаптації методу

Недоліки методу:

1. Від користувача вимагається кількість початкових центроїдів для початку роботи. Це вимагає присутності людини.
2. Алгоритм має певні проблеми, коли фінальні кластери матимуть різну величину та щільність.
3. Величини центроїдів можуть бути змінені невірно за рахунок шумних даних, або ж шум може взагалі мати власний кластер, якщо він не був видалений раніше. Отже, попередня обробка дуже важлива для коректної роботи методу k-середніх.
4. Чутливість до початкових значень центроїдів. Коли мова йде про маленьку кількість центроїдів (до 4-х), можна зменшити цю залежність, виконуючи метод декілька разів, вказавши різні початкові значення. Зі збільшенням кількості початкових центроїдів потрібно модифікувати метод таким чином, щоб похибка в результатах була мінімальною.

5. Коли обсяг даних збільшується, міра подібності на основі квадратичної відстані переходить у постійну константу між будь-якими вхідними даними. Необхідним в даній ситуації є зменшення розміру шляхом аналізу даних для характеристик, або шляхом спектральної кластеризації. Відношення стандартного відхилення до середньої квадратичної відстані між вхідними даними стрімко зменшується зі збільшенням кількості початкових кластерів. Це зменшення значить те, що метод став менш ефективним [**Error! Reference source not found.**].

## 1.5 Вибір технологій

Для того, щоб розпочати процес розробки комп'ютерної моделі, потрібно визначити технології для реалізації. Далі потрібно обрати найбільш ефективні та зручні засоби серед існуючих, проаналізувати актуальність технології на предмет використання її іншими розробниками. Далі наведені технології, що використані в процесі розробки даної системи.

### 1.5.1 Python

Існує багато мов програмування, що мають можливість інтеграції з процесом Data Mining. Далі наведений список найпоширеніші з цих мов програмування:

- Java;
- C++;
- Python;
- C#;
- JavaScript;
- PHP;

- Ruby;
- Perl;
- Go;
- Swift;
- Groovy.

Після проведеного аналізу було прийняте рішення обрати Python – інтерактивну та об'єктно-орієнтовану мову програмування. Python є розробкою простого для навчання засобу програмування. Цього вдалося досягнути шляхом того, що код, написаний на цій мові, дуже схожий на просту людську мову англійською. Гвідо ван Россум почав розробляти Python наприкінці 80-х років працівником Голландського Інституту математики та інформатики. Через широкий досвід розробників, Python набув багато функціоналу, схожого з мовами Algol, C, C++, Modula-3, SmallTalk.

Починаючі з 2010 року Python набуває більшої популярності у зв'язку с тим, що є універсальним засобом для будь-якої потреби (Див Рис. 1.8)

Завдяки популярності, за допомогою Python часто можна вирішити будь-які проблеми, які можуть виникнути в процесі розробки продукту будь-якої складності. Спільнота Python сильна, і її члени продовжують постійно розробляти покращення та налагодження технічної сторони. Python також має багато корпоративних спонсорів, які успішно популяризують мову та її методи розробки. Серед них такі технологічні гіганти як Google, що також використовую мову при розробці своїх комп'ютерних та наукових моделей [**Error! Reference source not found.**].

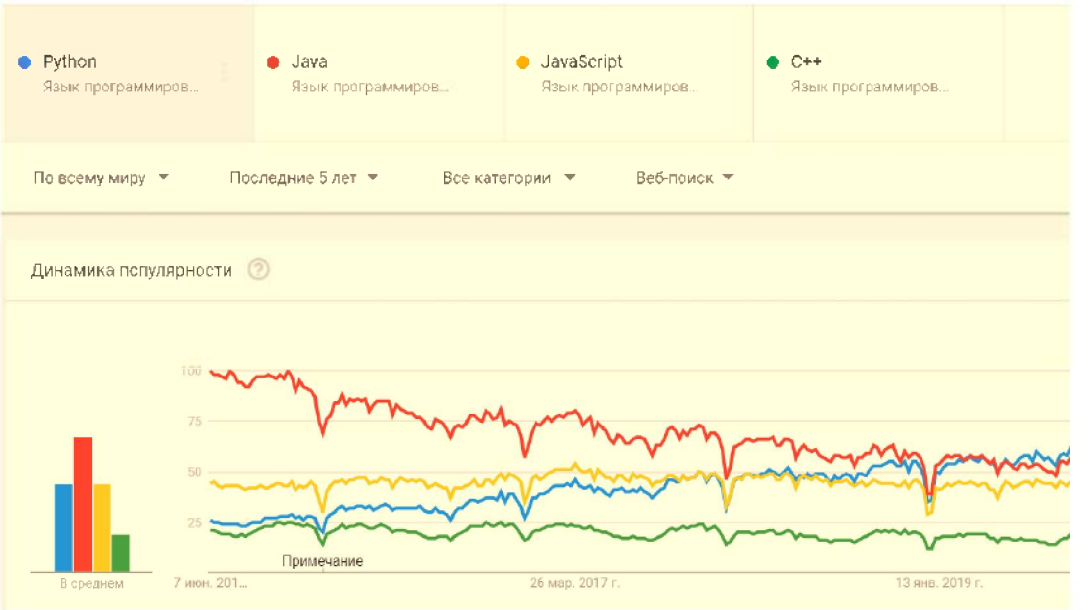



Рисунок 1.8 – Вибірка Google.com

Python був розроблений таким чином, щоб його було легко зрозуміти. Це спрощує написання нового коду Python та прискорює розробку програмного забезпечення Python. Це означає, що менше часу витрачається на боротьбу з проблемами налагодження програм і більше часу створення продукту безпосередньо.

Одна з найвідоміших особливостей Python полягає в тому, що час виконання коду є відносно повільним порівняно з іншими мовами. Однак є рішення цієї проблеми. Коли час виконання програми має високий пріоритет, Python дозволяє інтегрувати інші, більш ефективні мови у ваш код. Це оптимізує вашу швидкість, не змушуючи вас переписувати всю кодову базу з нуля. Слід зазначити, що найцінніший ресурс - це час виконання програми, а час, який розробники витрачають на розробку.

Ще однією перевагою Python є широкий вибір бібліотек та фреймворків, які він пропонує для різних ситуацій розробки. Їхнє використання підвищує ефективність роботи команди розробників, оскільки часто потрібні речі не

потрібно писати з нуля. Мова має засоби для багатьох процесів програмного забезпечення:

- Machine Learning;
- Data Science;
- Візуалізація даних;
- Складний аналіз даних.

Те ж саме стосується фреймворків, що прискорюють розробку проектів та програм [**Error! Reference source not found.**].

Python є інтуїтивно зрозумілим, тому що виглядає як проста англійська мова. Це спрощує розробку та підтримку програмного забезпечення. Крім того, Python має чіткий синтаксис і не вимагає такої кількості рядків коду, як Java або, наприклад, C для досягнення порівнянних результатів. Скорочення часу, що витрачається на перевірку коду, є безцінним, тому що збережений час можна витратити на розробку.

Python - найпопулярніша мова програмування, яку використовують мільйони інженерів для Data Science продуктів. Об'єктно-орієнтована мова програмування широко використовується в організації великих обсягів складних даних. Маючи динамічну семантику і непомірні можливості, Python також широко використовується для написання сценаріїв обробки великих обсягів даних різної складності.

Масштабованість кодової бази непередбачувана. Скільки реальних користувачів ви можете охопити, не можна сказати на початку проекту, тому важливо оцінити, наскільки серйозною має бути база коду. Тому Python є оптимальним вибором через його надійність і масштабованість. Деякі з найбільших продуктів, наприклад Microsoft, обрали Python саме з цієї причини.

### 1.5.2 Pandas

Pandas - це програмна бібліотека, написана для мови програмування Python для маніпуляції та аналізу даних. Зокрема, він пропонує структури даних та операції для маніпулювання числовими таблицями та часовими рядами. Це безкоштовне програмне забезпечення, випущене під ліцензією BSD з трьох пунктів. Назва походить від терміна «панельні дані», економетричного терміна для наборів даних, які включають спостереження протягом кількох періодів часу для одних і тих самих людей. Його назва — це гра на фразі «аналіз даних Python». Вес Маккінні почав створювати те, що стане бібліотекою в AQR Capital, коли працював там дослідником з 2007 по 2010 рік [**Error! Reference source not found.**].

Альтернативи Pandas використовують той самий функціонал та користуються досить великою популярністю серед ком'юніті розробників. Такими альтернативами є Pandasql, NumPy, Scipy, IPy, Matplotlib.

Тим не менш, Pandas має переваги серед наступного функціоналу:

- Об'єкт DataFrame для маніпулювання даними з інтегрованим індексуванням
- Інструменти для читання та запису даних між структурами даних у пам'яті та різними форматами файлів
- Вирівнювання даних та інтегрована обробка відсутніх даних
- Зміна та поворот наборів даних
- Нарізка на основі міток, химерна індексація та піднабір великих наборів даних
- Вставка та видалення стовпців структури даних
- Групування за механізмом, що дозволяє виконувати операції поділу-застосування-об'єднання над наборами даних

- Об'єднання наборів даних
- Ієрархічна вісь індексування для роботи з даними великого розміру в структурі даних меншого розміру
- Функціональність часових рядів: генерація діапазону дат і перетворення частоти, статистика рухомого вікна, лінійна регресія рухомого вікна, зсув і відставання дати.
- Забезпечує фільтрацію даних.

Pandas дозволяє імпортувати дані з різних форматів файлів, таких як значення, розділені комами, JSON, SQL та Microsoft Excel. Бібліотека дозволяє виконувати різноманітні операції маніпулювання даними, такі як об'єднання, зміна форми, виділення, а також очищення даних і функції сперечання даних.

### 1.5.3 Microsoft Excel

Microsoft Excel – це електронна таблиця, розроблена Microsoft для Windows, macOS, Android та iOS. Він містить обчислення, інструменти створення графіків, зведені таблиці та мову програмування макросів під назвою Visual Basic для додатків (VBA). Це була дуже широко застосовувана електронна таблиця для цих платформ, особливо з версії 5 в 1993 році, і вона замінила Lotus 1-2-3 як промисловий стандарт для електронних таблиць. Excel є частиною пакету програмного забезпечення Microsoft Office [**Error! Reference source not found.**].

Microsoft Excel має основні функції всіх електронних таблиць у процесі Data Mining, використовуючи сітку комірок, упорядковану в пронумеровані рядки та стовпці з літерними іменами, щоб організувати маніпуляції з даними, як-от арифметичні операції. Він має батарею функції, які відповідають статистичним, інженерним та фінансовим потребам. Це дозволяє розділяти дані, щоб переглянути їх залежності від різних факторів для різних точок зору

(за допомогою зведених таблиць і менеджера сценаріїв). Зведена таблиця — це потужний інструмент, який може заощадити час на аналізі даних. Він робить це шляхом спрощення великих наборів даних за допомогою полів зведеної таблиці. Він має аспект програмування, Data Mining для додатків, що дозволяє користувачеві використовувати широкий спектр числових методів, наприклад, для розв'язування задач кластерного аналізу, а потім звітувати про результати в електронну таблицю. Він також має різноманітні інтерактивні функції, що дозволяють використовувати інтерфейси, які можуть повністю приховати електронну таблицю від користувача, тому електронна таблиця представляє себе як так звана програма або система підтримки прийняття рішень (DSS) через спеціально розроблений інтерфейс користувача для наприклад, аналізатор запасів або взагалі як інструмент проектування, який ставить запитання користувачам і надає відповіді та звіти.

### 1.5.4 Telegram Bot API

Боти — це сторонні програми, які працюють всередині Telegram месенджеру (найпопулярніший месенджер на території СНД). Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити. Вони керують своїми ботами за допомогою HTTPS-запитів до нашого API бота. Використавши це, можна швидко та якісно реалізувати клієнтську частину комп'ютерної моделі. Більш того, всі, в кого є Telegram, матимуть змогу користуватися системою

Боти працюють наступним чином. Боти Telegram — це спеціальні облікові записи, для налаштування яких не потрібен додатковий номер телефону. Користувачі можуть взаємодіяти з ботами двома способами:

1) Надіславши повідомлення та команди ботам, відкривши з ними чат або додавши їх до груп.

2) Надіславши запити безпосередньо з поля пошуку. Це дозволяє надсилати вміст від вбудованих ботів безпосередньо в будь-який чат, групу чи канал.

Повідомлення, команди та запити, надіслані користувачами, передаються програмному забезпеченню, запущеному на серверах. Сервер-посередник обробляє шифрування та зв'язок із Telegram API. Спілкування з сервером відбувається через простий HTTPS-інтерфейс, який пропонує спрощену версію API Telegram.

### 1.5.5 Heroku хостинг-платформа

Комп'ютерну модель на основі TelegramBotAPI потрібно розмістити в WWW для широкого доступу для користувачів. Для цього необхідні засоби хостинг-платформи. Серед найпопулярніших: Vercel, Netlify, Heroku. Був обраний останній маючи низку переваг перед конкурентами:

- 550 годин/місяць роботи Dynos (1000 після прив'язки банківської картки)
- 512 Мб ОЗП на 1 контейнер
- CI/CD, CLI

Heroku — це хмарна платформа як сервіс (PaaS), що підтримує кілька мов програмування. Одна з перших хмарних платформ, Heroku, розроблялася з червня 2007 року, коли підтримувала лише мову програмування Ruby, але тепер підтримує Java, Node.js, Scala, Clojure, Python, PHP і Go. З цієї причини кажуть, що Heroku є платформою поліглотів, оскільки вона має можливості для розробника створювати, запускати та масштабувати програми подібним чином на більшості мов.

Програми, які запускаються на Heroku, зазвичай мають унікальний домен, який використовується для маршрутизації HTTP-запитів до правильного контейнера програми. Кожен з них розподіллено по "динаміці", яка

складається з кількох серверів. Сервер Git Heroku обробляє завантаження сховища програм від дозволених користувачів. Усі служби Heroku розміщені на платформі хмарних обчислень Amazon EC2.

### 1.5.6 Кросплатформеність

Слід зазначити, що кросплатформеність надто важлива, щоб її ігнорувати. Неможливо передбачити в якій операційній системі буде працювати модель, тому необхідно забезпечити правильну роботу моделі на всіх існуючих платформах. Всі інструменти, що використовуються в цій роботі, кросплатформні і, залежно від операційної системи, не вимагають будь-яких додаткових дій з боку користувача.

Microsoft Excel – це програма, яка працює з форматом XLSX на платформі Windows. Однак аналоги цієї програми є на кожній платформі і є можливість відкрити файл XLSX для роботи.

Python і всі його бібліотеки можуть працювати на будь-якій платформі та відображати один і той самий висновок програми.

## 1.6 Огляд системи

Задачею даної роботи є розробка комп'ютерної моделі препроцесінгу даних фондового ринку з метою аналізу ситуації для інвестора. Методом препроцесінгу даних в даній системі є метод кластеризації k-середніх. Алгоритм роботи системи описаний далі:

1) Вхідними даними програми є лише рік. Це реалізовано з метою можливості аналізу ситуації за поточний та минулі роки. Можливий діапазон років – 1970-поточний рік. Див. Рис. 1.9 та Рис 1.10

2) TelegramBotAPI отримує рік і передає його серверному коду, де далі працює програма

3) Отримавши рік, програма виконує наступні кроки:

    a. Збирає назви крупніших компаній

    b. Збирає індекси цих компаній

    c. Проводить кластеризацію отриманих значень індексів

    d. Формує вихідний файл з рекомендаціями

4) Вихідний файл складається з 3 частин, кожна з яких представлена окремим кластером – результатом процесу кластеризації. Кластери виділені відповідними кольорами для зручності користувача:

    a. Перший зелений – кластер із компаніями, що рекомендовані для інвестування

    b. Другий жовтий – кластер із компаніями, інвестування в які залишені на передбачення інвестора

    c. Третій червоний – не рекомендовані для інвестування компанії

    На Рис.1.11 та Рис.1.12 зображені межі кластерів для більшого розуміння як виглядають дані вихідного файлу

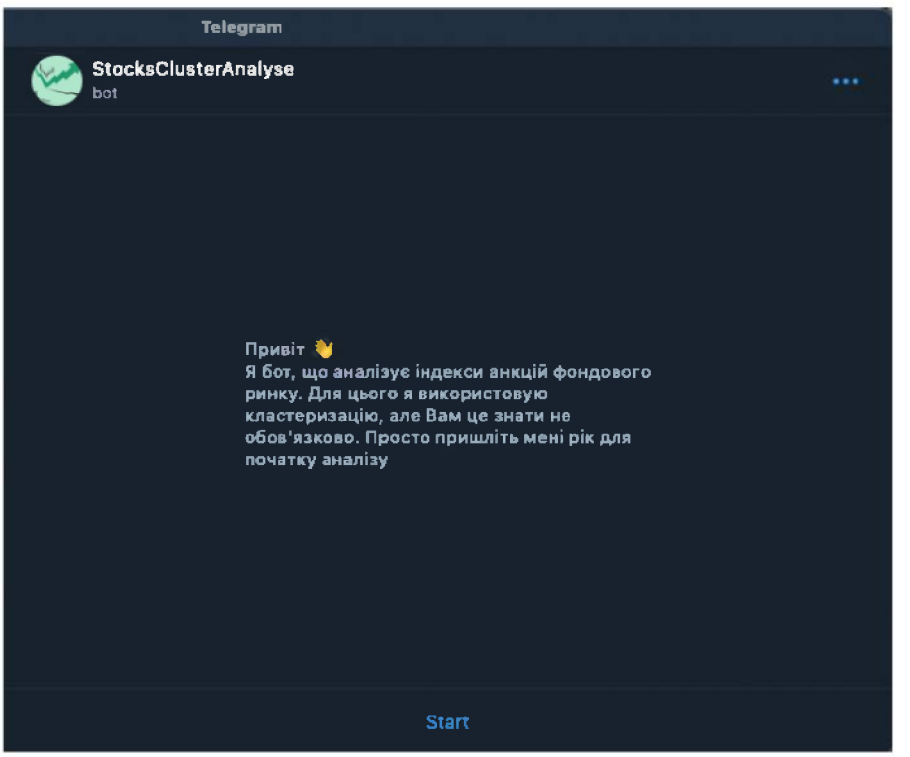



Рисунок 1.9 – Привітання боту

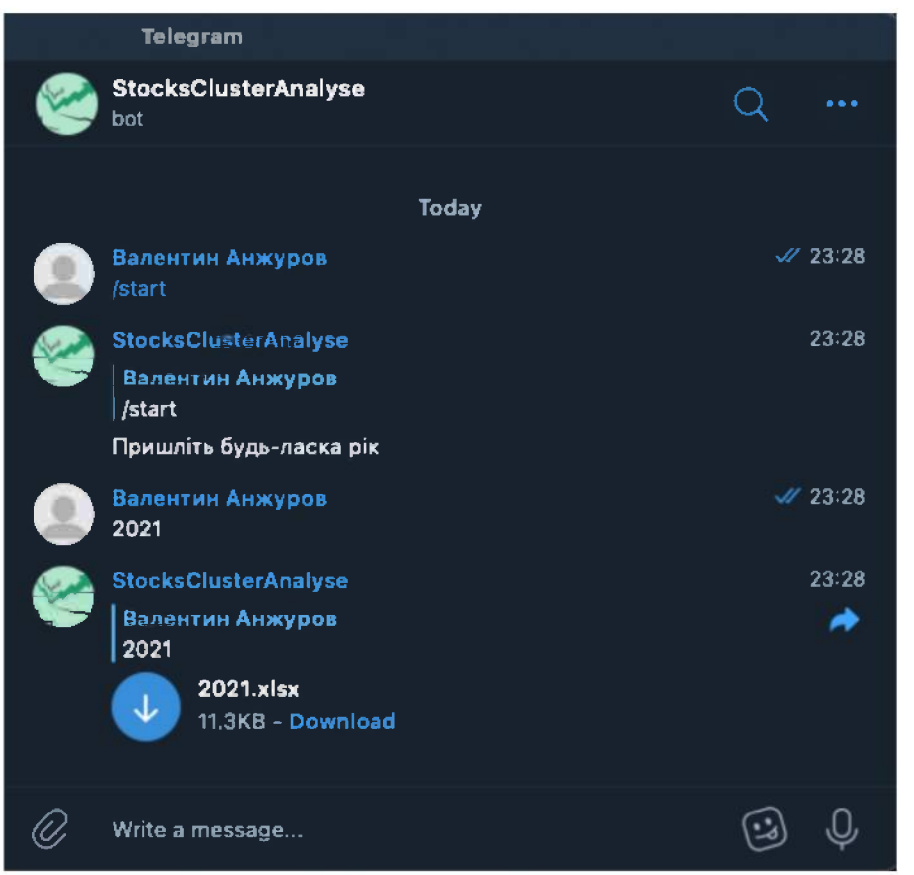



Рисунок 1.10 – Приклад інтерактивної роботи з ботом

| | |
|---|---|
| ABC | 12,8 |
| BWA | 12,87 |
| MS | 12,95 |
| NOC | 13 |
| NWL | 13,16 |
| BK | 13,39 |
| LMT | 13,42 |
| WFC | 13,45 |
| STT | 13,48 |
| GPS | 13,79 |
| CZR | 13,99 |
| TPR | 14,16 |
| BAC | 14,21 |
| SNA | 14,54 |
| ADM | 14,64 |

Рисунок 1.11 – Межа 1-го та 2-го кластерів

| | |
|---|---|
| CMS | 21,36 |
| TER | 21,36 |
| EBAY | 21,41 |
| GPC | 21,46 |
| WDC | 21,48 |
| WEC | 21,59 |
| AMGN | 21,62 |
| CSCO | 21,78 |
| KHC | 21,87 |
| ANTM | 21,9 |
| ORLY | 21,93 |
| ED | 22,03 |
| NSC | 22,17 |
| AEE | 22,2 |
| UNP | 22,3 |

Рисунок 1.12 – Межа 2-го та 3-го кластерів

**Висновки до розділу 1**.

В даному розділі були розглянуті основні концепції Data Mining. Також, був проведений огляд та аналіз методів препроцесінгу даних в Data Mining. Більш детально була звернута увага на метод кластеризаціїї k-середніх.

Були розглянуті технології, використані в процесі розробки комп'ютерної моделі даної роботи.

Більш того, був проведений огляд системи, засобів для створення комп'ютерних моделей

Розглянувши концепції Data Mining та метод k-середніх, можна зробити висновок, що цей метод є дуже зручним для вирішення задані аналізу індексів акцій на фондовому ринку

# РОЗДІЛ 2.
# РОЗРОБКА КОМП'ЮТЕРНОЇ МОДЕЛІ АНАЛІЗУ ІНДЕКСІВ АКЦІЙ НА ФОНДОВОМУ РИНКУ

Кінцевою метою продукту розробки є комп'ютерна модель аналізу індексів акцій фондового ринку, що повинна бути здатна обробити великий обсяг інформації за найкоротший час. Система має бути якомога простішою у використанні користувачем при збереженні обчислювальної здатності. Також, було б добре, щоб система складалася з найменшого обсягу програмного коду, так як менший об'єм коду спрощує майбутнє супроводження продукту.

## 2.1 Структура комп'ютерної моделі

SOLID — це абревіатура складена з перших літер п'яти базових принципів об'єктно-орієнтованого програмування та дизайну, запропонована Робертом Мартіном. Принципи SOLID використовують для дизайну та розробки таких програмних систем, які, з великою ймовірністю, зможуть тривалу годину розвиватися, розширятися та підтримуватися [**Error! Reference source not found.**].

S – Принцип єдиної відповідальності (single responsibility principle). Для кожного класу має бути визначено єдине призначення. Всі ресурси, необхідні для його здійснення, повинні бути інкапсульовані в цей клас і підпорядковані тільки цьому завданню.

O - Принцип відкритості/закритості (open-closed principle). «Програмні сутності мають бути відкриті для розширення, але закриті для модифікації».

L - Принцип підстановки Ліскoв (Liskov substitution principle). «об'єкти у програмі повинні бути замінені на екземпляри їх підтипів без зміни правильності виконання програми». також контрактне програмування. Похідний клас має бути взаємозамінним із батьківським класом.

I - Принцип розподілу інтерфейсу (interface segregation principle). «багато інтерфейсів, спеціально призначених для клієнтів, краще, ніж один інтерфейс загального призначення».

D – Принцип інверсії залежностей (dependency inversion principle). «Залежність на абстракціях. Немає залежності на щось конкретне»

Конструкція системи, що є метою цієї кваліфікаційної роботи, складається з різних частин. Кожна з цих частин виконую одну і тільки одну власну функцію. Незалежно від того, якої складності є система, вона повинна виконувати деякі правила вірного дизайну програмного забезпечення. Більш детально у даному підрозділі можна розглянути принцип S – єдиної відповідальності. Слідуючи цьому правилу, пропонується переслідувати «горизонтальне» розширення систем, де відбувається не розширення існуючого функціоналу, а збільшення кількості маленьких блоків програмного забезпечення. Таким чином, будь-яка проблема, що виникає під час роботи може бути швидко усунена.

На Рис. 2.1 можна розглянути частими даної комп'ютерної моделі. Кожен прямокутник визначає окремий функціональний блок, в середині якого позначена дія, яку він виконує.

- Модуль зчитування вхідних даних. Даний блок створює зчитування року роботи фондового ринку шляхом інтерактивного спілкування з користувачем через консоль месенджеру Telegram. Для цього він використовує засоби Telegram Bot API
- Валідація вхідних даних. Цей модуль здійснює перевірку вхідних параметрів для подальшої обробки системою. Рік роботи фондового ринку повинен бути цілим числом і зі значенням дійсного фондового ринку. Також потрібна перевірка того, що в даний рік біржа працювала (почала роботу з 1970 року)

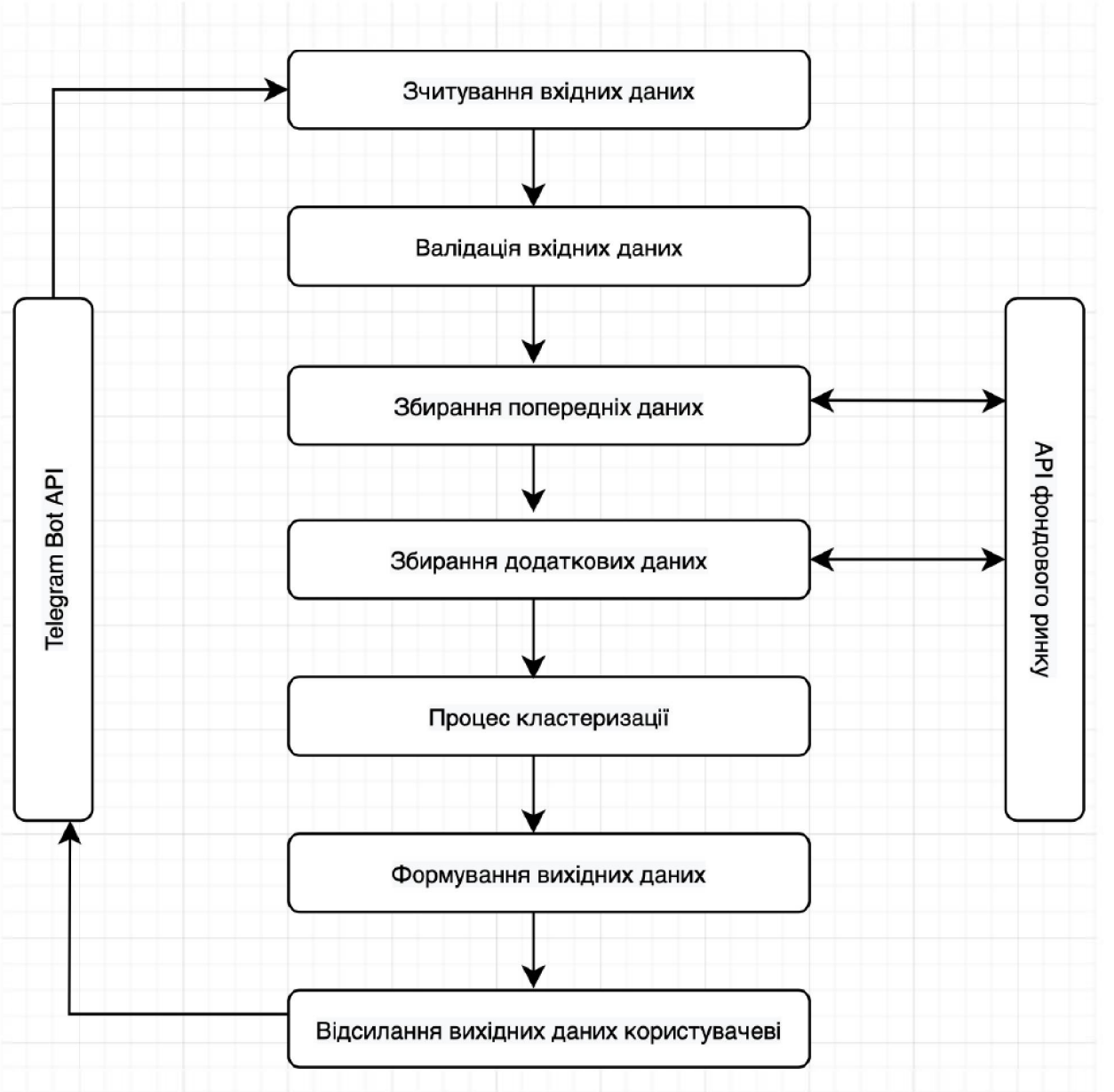



Рисунок 2.1 – Блоки системи

- Збирання попередніх даних. В цьому функціональному блоці необхідно визначити кандидатів для наступних етапів аналізу. Шляхом взаємодії з API фондового ринку система отримує велику кількість тикерів (ідентифікаторів) компаній, починаючі з найбільших і закінчуючи найменшими, так як вони можу бути перспективнішими за гігантів

- Збирання додаткових даних. Даний блок здійснює збір індексів компаній, що були зібрані попереднім блоком. Відбувається робота з іншим API фондової біржі, який повертає значення індексу P/E для кожного емітенту
- Модуль кластеризації даних. Цей функціональний блок є основним в даній системі і він містить алгоритм кластеризації k-середніх, що застосовується до попередньо зібраних значень P/E індексу. Зчитування даних алгоритмом виконується дуже швидко, основною часовою затрато є процес кластеризації. Алгоритм роботи даного модуля більш детально зображений на Рис. 2.2

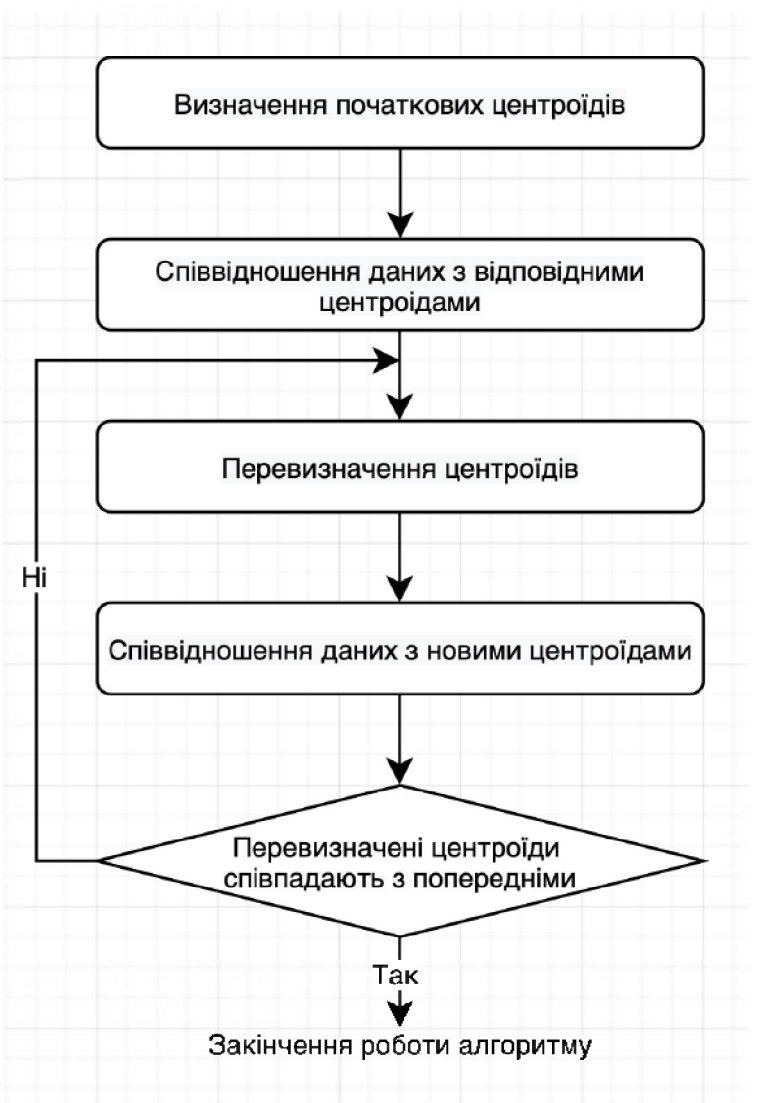



Рисунок 2.2 – Схема модуля кластеризації даних

- Формування вихідних даних. Результат кластеризації представлений Pandas конструкцією і не може бути представлений користувачеві. Даний модель конвертує конструкцію в excel файл, який легко аналізується людиною, що не має досвіду розробника
- Відсилання вихідних даних користувачеві. Цей блок, так само як і перший в даному списку, працює з користувачем через Telegram консоль засобами Telegram Bot API. Блок відсилає excel файл користувачеві шляхом HTTPS протоколу

Нижче на Рис. 2.3 наведена IDEF0 [**Error! Reference source not found.**] діаграма комп'ютерної моделі кластеризації

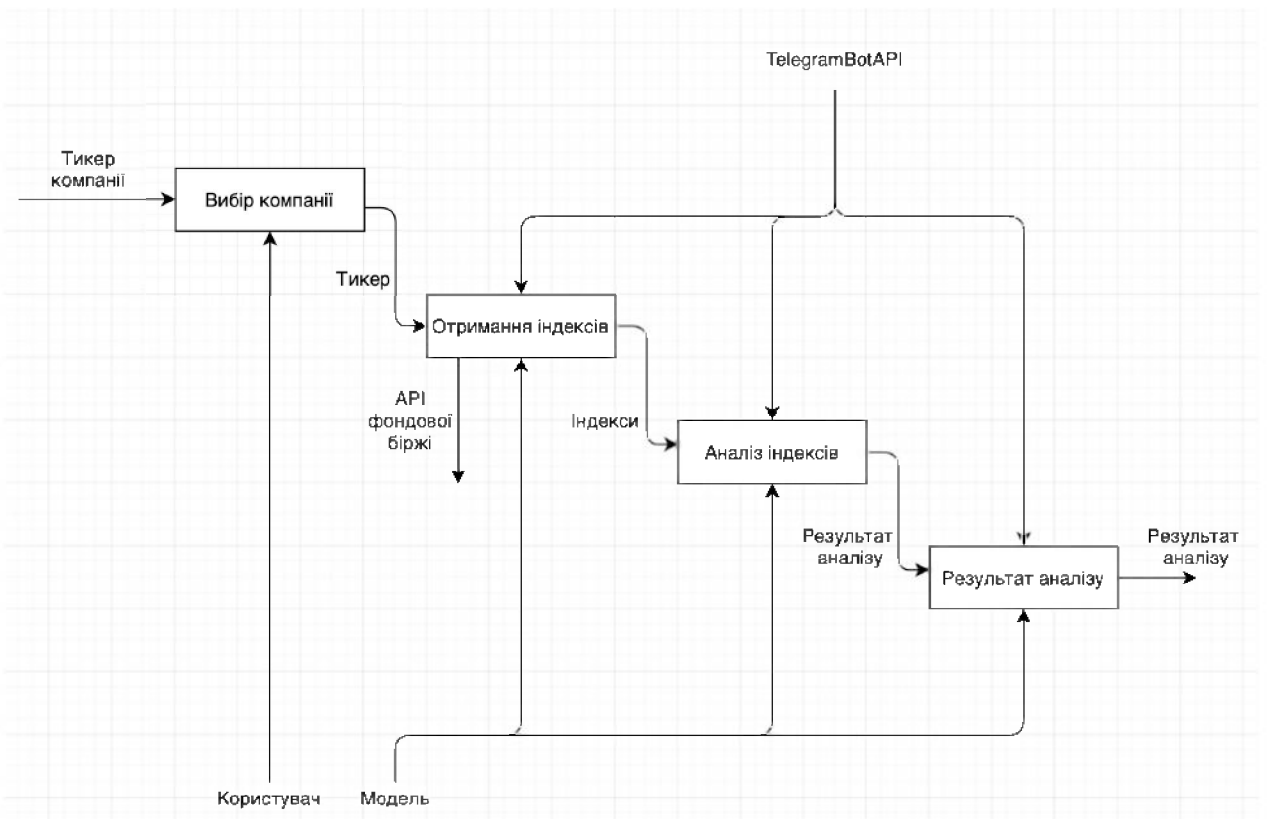



Рисунок 2.3 - модель в нотації IDEF0

Правильно структурована модель складається із частин, що можливо використовувати в інших проектах. Отже, якщо ми маємо модуль, що містить алгоритм зчитування даних користувача, ми можемо використовувати його в інших проектах зі схожою функціональністю. Це стверджує факт того, що модуль спроектований таким чином, що він не залежить від сусідніх або інших модулів програмного забезпечення. Кожен із перелічених вище модулів працює саме таким чином. Розділення системи на перелічені частини достатньо, щоб вважати архітектуру придатною для подальшого зручного розширення. Крім того, набору таких блоків є достатнім для виконання поставленої задачі аналізу даних фондового ринку. Небажано розбивати модель на докладніші блоки, оскільки це може ускладнити навігацію в середині коду.

### 2.2 Вхідні та вихідні дані

Однією з найважливіших етапів проектування комп'ютерної моделі будь-якого призначення є вибір формату вхідних та вихідних даних. Складність структури цих даних прямим чином впливає на те, чи є інтерфейс системи простим і не перенавантаженим для кінцевого користувача. Натомість, надто простий формат даних може спричинити втрату деталей результату опрацювання інформації моделлю.

Існує не так багато популярних форматів вхідних/вихідних даних. Далі проведений аналіз для вибору найоптимальніших з них з урахуванням особливостей роботи даної системи:

1. Звичайний текст. Дані представлені звичайним текстом, що по суті є послідовністю кодів переведених процесом кодування у символи. Даний варіант, як правило, потребує додаткової валідації, так як користувач не має змоги обирати з передбачених варіантів

2. Текстовий файл. Є близьким аналогом варіанту 1, за виключення того, що текст зберігається, безпосередньо, в текстовому файлі, що має своє власне кодування

3. Excel файл. Є близьким аналогом варіанту 2, тільки дані структуровані сіткою, в якій доступ до даних здійснюється методом звернення до комірки. Кожна з комірок є незалежною, або ж залежною від іншим. Даний процес залишається на розсуд користувача того чи іншого excel файлу

4. База даних. Є дуже серйозним варіантом зберігання даних будь-якої складності. Більш того, сервер бази даних є доступним будь-кому (при бажання) з будь-якої точки світу, де є Інтернет та засіб спілкування за HTTP протоколом. Дані в базі даних структуровані таблицями та колонками в цих таблицях. Також, можлива типізація даних для запобігання валідації даних при кожній операції модифікації. Для роботи з базами даних застосовується окремий синтаксис SQL, що потребує додаткового вивчення розробником

Кожен з перелічених типів вхідних/вихідних даних є популярним та визнаним спілкою розробників програмного забезпечення. Проте, кожен з них має переваги і недоліки в залежності від комп'ютерної моделі, в якій вони застосовуються. Розглянемо ці типи в контексті даного програмного забезпечення:

1. Текст є надто важким для сприйняття користувачем форматом для вихідним даних. Важко уявити наскільки складним може бути аналіз «сирого» тексту, що містить складну структуру даних в середині. Також, важко працювати з простим текстом з перспективою майбутнього розширення програмного забезпечення.

Проте, даний формат є дуже вигідним з точки зору використання його для формату вхідних даних. Дана система приймає лише рік роботи

фондового ринку. Інші варіанти для даної мети є занадто ускладненими, та зроблять роботу з системою лише важчою

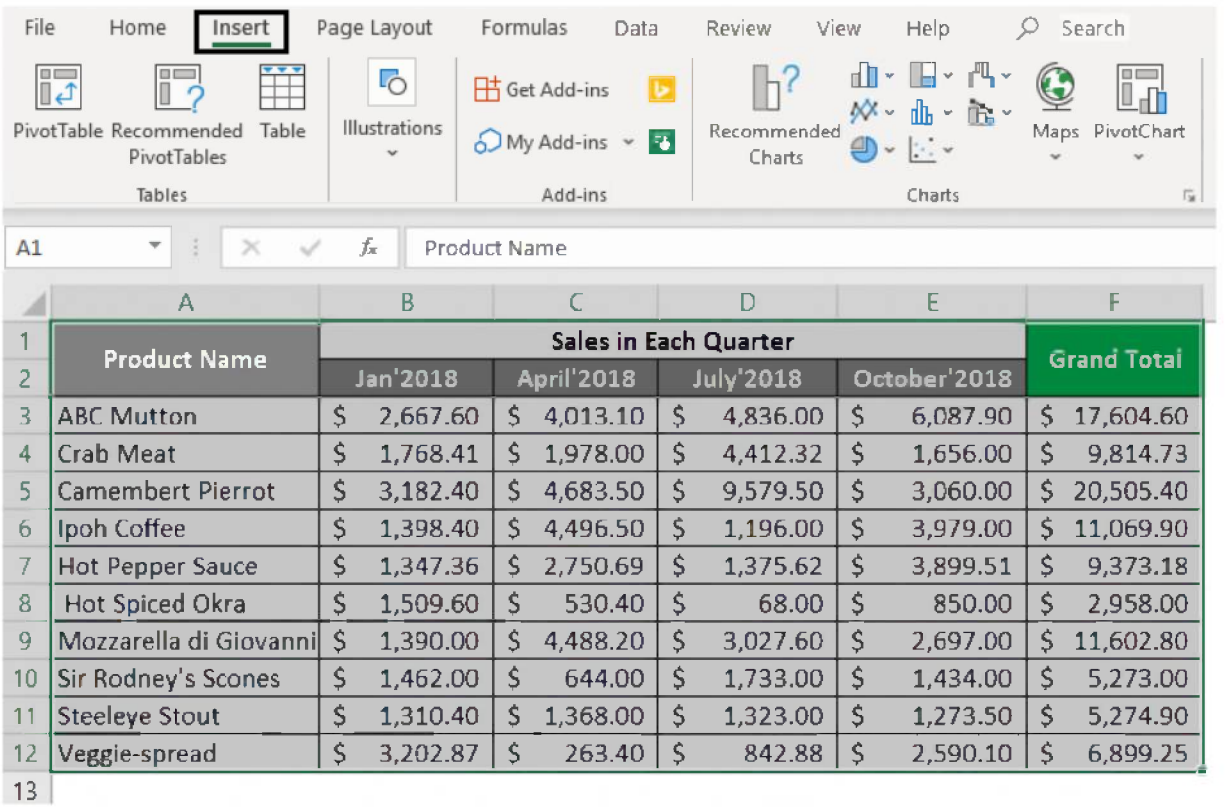
2. Файл з текстом. Так само, як і текст є дуже важким для сприйняття користувачем, враховуючі комплексність вихідних даних в даній комп'ютерній моделі. Також, звичайні текстові файли не мають засобів для виділенні тексту різними кольорами, що може дуже допомогти при аналізі вихідної інформації. Враховуючі вищесказане, можна зробити висновок, що текстовий файл не є оптимальним варіантом ні для вхідних, ні для вихідних даних

3. Excel файл. Після проведення проектування комп'ютерної моделі було прийняте рішення, що excel-файл є найзручнішим типом вихідних даних. Дані у такому файлі можуть бути розділені на аркуші, відсортовані та відфільтровані. Також, є засіб виділення даних різними кольорами. До того ж, Python має велику кількість засобів роботи з такими excel-файлами, що є достатньо оптимізованими та налагодженими. Приклад даних у excel файлі зображено на Рис. 2.3

4. База даних, як правило, використовуються для роботи за даними, що мають дуже складну структуру [**Error! Reference source not found.**][**Error! Reference source not found.**]. Також системи, де застосовані бази даних, зазвичай є високонавантаженими і основну частину спрощення роботи з багатьма потоками бере на себе безпосередньо база даних. Приклад складної структури бази даних наведено на Рис. 2.4

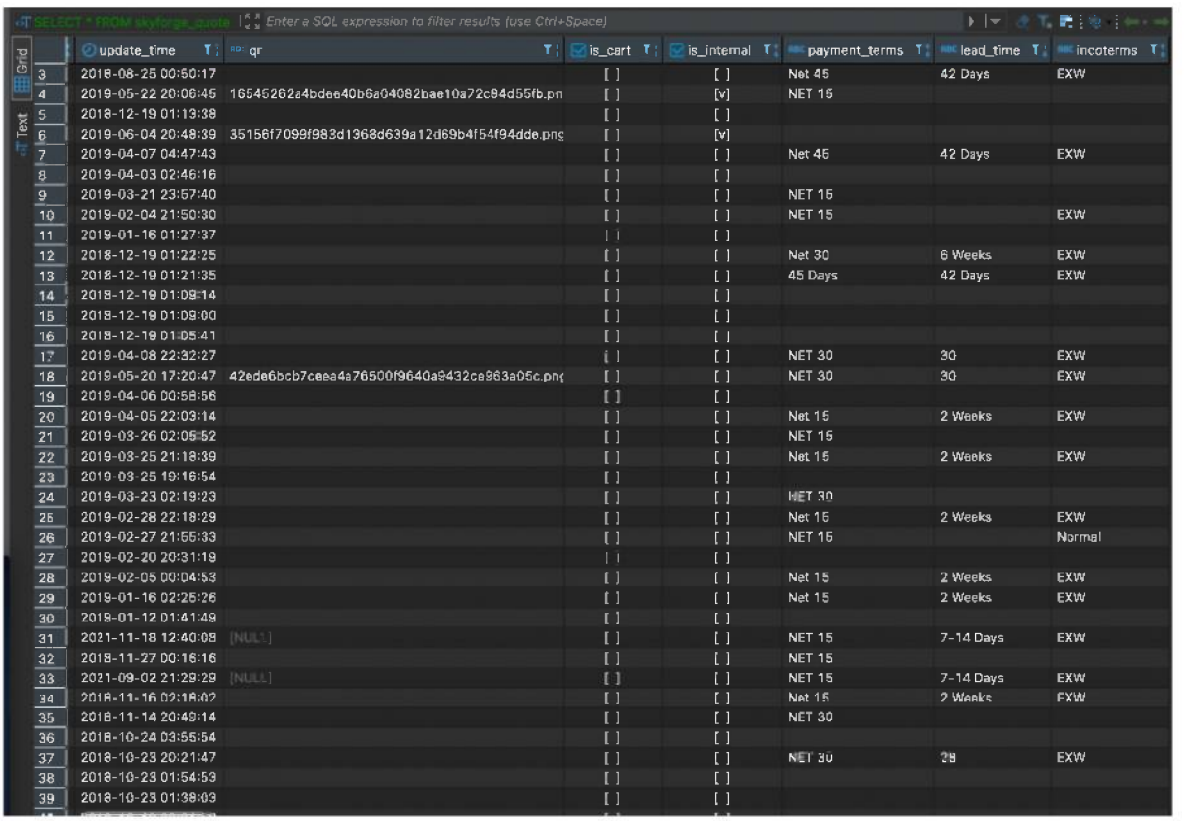
База дана потребує окремого сервера, що потребує окремої роботи, такої як розгортання серверу, його налагодження. Також, потрібно, щоб усі розробники на проекті вміли працювати з діалектом SQL для роботи з даними в базі даних. При проектуванні моделі було прийняте рішення, що даний тип даних є занадто складним для рішення поточної задачі.

Більш того, потрібні додаткові фінансові трати на хостинг серверу бази даних для забезпечення роботи з ним для усіх користувачів




| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Product Name | Sales in Each Quarter | | | | Grand Total |
| 2 | | Jan'2018 | April'2018 | July'2018 | October'2018 | |
| 3 | ABC Mutton | $ 2,667.60 | $ 4,013.10 | $ 4,836.00 | $ 6,087.90 | $ 17,604.60 |
| 4 | Crab Meat | $ 1,768.41 | $ 1,978.00 | $ 4,412.32 | $ 1,656.00 | $ 9,814.73 |
| 5 | Camembert Pierrot | $ 3,182.40 | $ 4,683.50 | $ 9,579.50 | $ 3,060.00 | $ 20,505.40 |
| 6 | Ipoh Coffee | $ 1,398.40 | $ 4,496.50 | $ 1,196.00 | $ 3,979.00 | $ 11,069.90 |
| 7 | Hot Pepper Sauce | $ 1,347.36 | $ 2,750.69 | $ 1,375.62 | $ 3,899.51 | $ 9,373.18 |
| 8 | Hot Spiced Okra | $ 1,509.60 | $ 530.40 | $ 68.00 | $ 850.00 | $ 2,958.00 |
| 9 | Mozzarella di Giovanni | $ 1,390.00 | $ 4,488.20 | $ 3,027.60 | $ 2,697.00 | $ 11,602.80 |
| 10 | Sir Rodney's Scones | $ 1,462.00 | $ 644.00 | $ 1,733.00 | $ 1,434.00 | $ 5,273.00 |
| 11 | Steeleye Stout | $ 1,310.40 | $ 1,368.00 | $ 1,323.00 | $ 1,273.50 | $ 5,274.90 |
| 12 | Veggie-spread | $ 3,202.87 | $ 263.40 | $ 842.88 | $ 2,590.10 | $ 6,899.25 |
| 13 | | | | | | |

Рисунок 2.3 – приклад excel файлу



| | update_time | qr | is_cart | is_internal | payment_terms | lead_time | incoterms |
|---|---|---|---|---|---|---|---|
| 3 | 2018-08-25 00:50:17 | | [ ] | [ ] | Net 45 | 42 Days | EXW |
| 4 | 2019-05-22 20:06:45 | 16545262a4bdee40b6a04082bae10a72c84d55fb.pn | [ ] | [v] | NET 15 | | |
| 5 | 2018-12-19 01:13:38 | | [ ] | [ ] | | | |
| 6 | 2019-06-04 20:48:39 | 35156f7099f983d1368d639a12d69b4f54f94dde.png | [ ] | [v] | | | |
| 7 | 2019-04-07 04:47:43 | | [ ] | [ ] | Net 45 | 42 Days | EXW |
| 8 | 2019-04-03 02:46:16 | | [ ] | [ ] | | | |
| 9 | 2019-03-21 23:57:40 | | [ ] | [ ] | NET 15 | | |
| 10 | 2019-02-04 21:50:30 | | [ ] | [ ] | NET 15 | | EXW |
| 11 | 2019-01-16 01:27:37 | | [ ] | [ ] | | | |
| 12 | 2018-12-19 01:22:25 | | [ ] | [ ] | Net 30 | 6 Weeks | EXW |
| 13 | 2018-12-19 01:21:35 | | [ ] | [ ] | 45 Days | 42 Days | EXW |
| 14 | 2018-12-19 01:09:14 | | [ ] | [ ] | | | |
| 15 | 2018-12-19 01:09:00 | | [ ] | [ ] | | | |
| 16 | 2018-12-19 01:05:41 | | [ ] | [ ] | | | |
| 17 | 2019-04-08 22:32:27 | | [ ] | [ ] | NET 30 | 30 | EXW |
| 18 | 2019-05-20 17:20:47 | 42ede6bcb7ceea4a76500f9640a9432ca963a05c.png | [ ] | [ ] | NET 30 | 30 | EXW |
| 19 | 2019-04-06 00:58:56 | | [ ] | [ ] | | | |
| 20 | 2019-04-05 22:03:14 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 21 | 2019-03-26 02:05:52 | | [ ] | [ ] | NET 15 | | |
| 22 | 2019-03-25 21:18:39 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 23 | 2019-03-25 19:16:54 | | [ ] | [ ] | | | |
| 24 | 2019-03-23 02:19:23 | | [ ] | [ ] | NET 30 | | |
| 25 | 2019-02-28 22:18:29 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 26 | 2019-02-27 21:55:33 | | [ ] | [ ] | NET 15 | | Normal |
| 27 | 2019-02-20 20:31:19 | | [ ] | [ ] | | | |
| 28 | 2019-02-05 00:04:53 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 29 | 2019-01-16 02:25:26 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 30 | 2019-01-12 01:41:49 | | [ ] | [ ] | | | |
| 31 | 2021-11-18 12:40:08 | [NULL] | [ ] | [ ] | NET 15 | 7-14 Days | EXW |
| 32 | 2018-11-27 00:16:16 | | [ ] | [ ] | NET 15 | | |
| 33 | 2021-09-02 21:29:29 | [NULL] | [ ] | [ ] | NET 15 | 7-14 Days | EXW |
| 34 | 2018-11-16 02:16:02 | | [ ] | [ ] | Net 15 | 2 Weeks | EXW |
| 35 | 2018-11-14 20:49:14 | | [ ] | [ ] | NET 30 | | |
| 36 | 2018-10-24 03:55:54 | | [ ] | [ ] | | | |
| 37 | 2018-10-23 20:21:47 | | [ ] | [ ] | NET 30 | 28 | EXW |
| 38 | 2018-10-23 01:54:53 | | [ ] | [ ] | | | |
| 39 | 2018-10-23 01:38:03 | | [ ] | [ ] | | | |

Рисунок 2.4 – приклад бази даних

### 2.3 Реалізація кластеризація на Python

Використавши Python, був отриманий короткий та змістовний код алгоритму k-середніх. Ця мова була створена таким чином, щоб з нею було зручно працювати людям, що не програмували до цього. Велика купа конструкцій має близький до людської мови вигляд. Це зробило даний алгоритм простим і швидким в роботі [**Error! Reference source not found.**].

Код методу k-середніх реалізований наступними власними функціями. Лістинги цих функцій також представлені у Додатку Г:

1. Функція `define_centroids` визначає значення початкових центроїдів набору даних. Програма містить константу `CLUSTERS_COUNT=3`, що дорівнює кількості початкових кластерів, на які буде розділений набір інформації. Дана зміна є константною, тому що потрібно отримати 3 результуючі кластери (рекомендовані, нейтральні та нерекомендовані

акції). Зміна даної змінною не бажана, так як вона може призвести до неочікуваних результатів

2. Алгоритм визначає випадкові значення, що формуються в список, який далі представлений списком перших центроїдів
3. Функція `assignment` відносить дані до найближчого кластеру базуючись на квадратичній відстані. Спочатку функція знаходить відстань між кожним значенням даних та кожним центроїдом, а потім алгоритмічно визначає до якого з них це значення ближче
4. Функція `refresh_centroids` визначає нові значення центроїдів попередньо утворених кластерів. Центроїди кластерів намагаються зміститися якомога ближче до середнього значення відносно усіх значень центроїда. Після виконання цієї функції кожен центроїд опиняється майже в середині власного кластеру. Далі алгоритм буде перевіряти чи продовжують центроїди переміщуватися і, коли вони зупиняються, тоді алгоритм завершує свою роботу
5. Функція `do_clustering` оперує функціями 1-4 порядком, що передбачений методом k-середніх та описаний в розділі 1.2 даної кваліфікаційної роботи. Після виконання функції `do_clustering`, вона повертає об'єкт DataFrame, що є частиною бібліотеки Pandas, що працює з DataMining. Цей об'єкт утримує кластеризовані дані, які далі будуть конвертовані системою у результуючий excel файл

### 2.4 Хостинг системи

Яка б не була корисна та якісна система, всі зусилля не мають сенсу, якщо нею не можуть користуватися кінцеві користувачі. Доступ до системи із різних куточків світу можна забезпечити лише через Інтернет через протокол

HTTP/HTTPS. Для цього систему потрібно перемістити із свого комп'ютеру на так званий хостинг [**Error! Reference source not found.**].

Хостинг (англ. hosting) — послуга надання ресурсів для розміщення інформації на сервері (зазвичай Інтернет). Зазвичай послуга хостингу входить до пакета обслуговування сайту і передбачає, як мінімум, розміщення файлів сайту на сервері, на якому запущено ПЗ, необхідне для обробки запитів до цих файлів (веб-сервер). Як правило, обслуговування вже входить надання місця для поштової кореспонденції, баз даних, DNS, файлового сховища на спеціально виділеному файл-сервері тощо, а також підтримка функціонування відповідних сервісів. Хостинг бази даних, розміщення файлів, хостинг електронної пошти, послуги DNS можуть надаватися окремо як самостійні послуги або входити в комплексну послугу.

Існують рішення для хостингу такі як:

- Back4App
- AWS & Azure & Google Cloud
- Dokku
- Firebase
- OpenShift
- Engine Yard
- Netlify
- Heroku

Всі ці рішення мають переваги та недоліки. Після їх аналізу було прийняте рішення розмістити систему, використавши засоби Heroku, тому що цей хостинг має наступні переваги:

- Пропонує єдиний білінг для всіх проектів із розбивкою по командам Моніторинг та підвищення продуктивності завдяки розширеному моніторингу додатків
- Високопродуктивна інтеграція з Salesforce Data Mining
- Проста горизонтальна та вертикальна масштабованість
- Провідна екосистема інструментів та сервісів платформи
- Швидке управління життєвим циклом додатків та дозволами
- Пропонує потужну панель приладів та інтерфейс командного рядка.
- Інтегрується зі знайомими робочими процесами розробника
- Ряд автоматизованих функцій, включаючи масштабування, конфігурацію, налаштування та інші.
- Проста інтеграція з іншими продуктами AWS
- Гнучкість для налаштування та підтримки унікальних потреб робочого процесу DevOps

Для того, щоб змусити комп'ютерну модель працювати на Heroku хостингу, потрібно виконати наступні кроки:

1) Створити профіль на heroku.com
2) Встановити Heroku CLI на комп'ютер
3) Виконати `heroku login` та `heroku create`, щоб створити аплікацію на хостингу. Результасst команд зображений на Рис. 2.5
4) Далі зв'язуємо систему та git адресу Heroku додатку
5) Виконавши `git add * && git commit -m "First deploy" && git push heroku master` переміщуємо код системи на хостинг та запускаємо її

6) Система зараз на хостингу і до неї можна звернутися через Telegram месенджер. На Рис. 2.6. показана створена аплікація

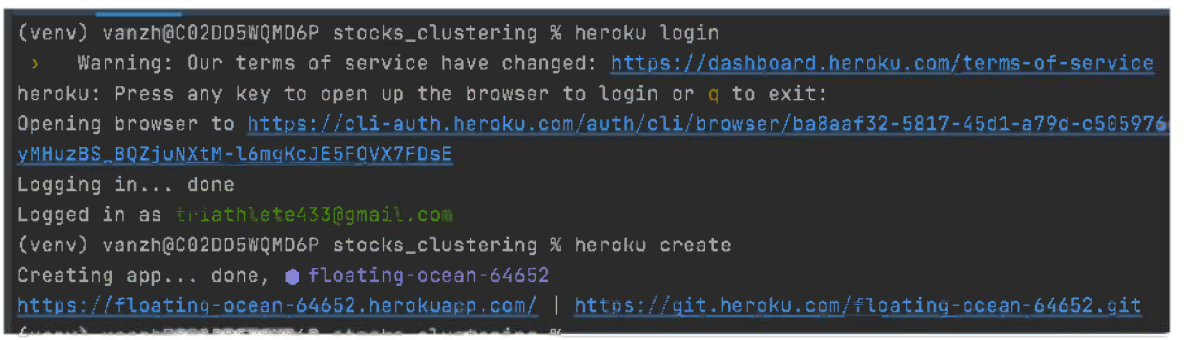



Рисунок 2.5 – початок роботи з Heroku CLI

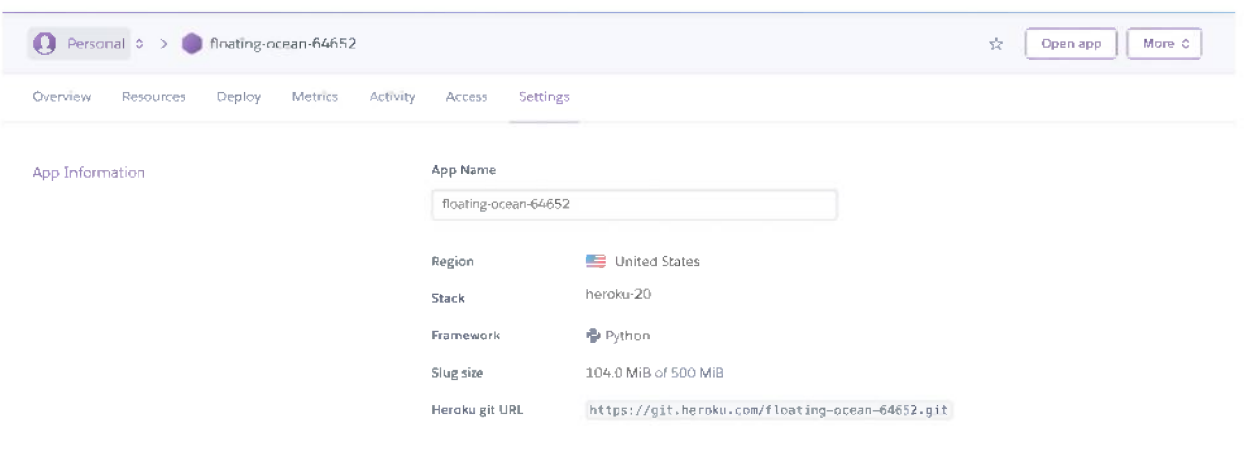



Рисунок 2.6 – аплікація на Heroku

**Висновки до розділу 2.**

В даному розділі були розглянуті та проаналізовані основні поняття та концепції створення комп'ютерної моделі аналізу індексів акцій на фондовому ринку. Огляд торкнувся методології SOLID, діаграми моделі IDEF0, а також окремо кожного функціонального модуля

Можна зробити висновок, що розроблена модель відповідає всім вимогам до комп'ютерних моделей за критеріями дизайну архітектури, розбиття моделі на функціональні модулі, проведення кінцевого тестування

# РОЗДІЛ 3.
# ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАЧА

## 3.1 Мінімальні вимоги до ОС

Мінімальні вимоги до комп'ютерної моделі представлені технічними засобами системи та операційною системою, де вона працює. Python та інші засоби, використані системою, не є занадто вимогливими навіть для слабких машин. Тим паче, будь-яка ОС має вбудований функціонал для роботи з каскадними таблицями, а отже, можна не зважати уваги на розмір вихідного файлу при аналізі його користувачем. Нижче наведені вимоги, що задовольняють технічні засоби даної системи [**Error! Reference source not found.**]:

- Процесор: Intel Celeron
- Вільне місце на жорсткому диску: < 100mb
- Операційна система: Windows 7 або новіша, MacOS, Linux
- Оперативна пам'ять: 1Gb

Проаналізувавши вимоги вище, можна зазначити, що навіть застарілі комп'ютери здатні розгорнути комп'ютерну модель даної кваліфікаційної роботи [**Error! Reference source not found.**].

## 3.2 Робота системи

Основний функціонал, що представляє роботу системи, представлений у файлі app.py (Див. Додаток Г)

Шляхом HTTPS протоколу додаток отримує рік роботи фондового ринку для аналізу ринку саме в цьому році. Далі відбувається валідація року. Він повинен бути:

1. Числом

2. Значенням 1970-поточний рік. Саме в 1970 біржа почала роботу з індексом P/E

Якщо валідація не пройшла, користувач отримує відповідне пояснення і програма завершує свою роботу (Див. Рис. 3.1)

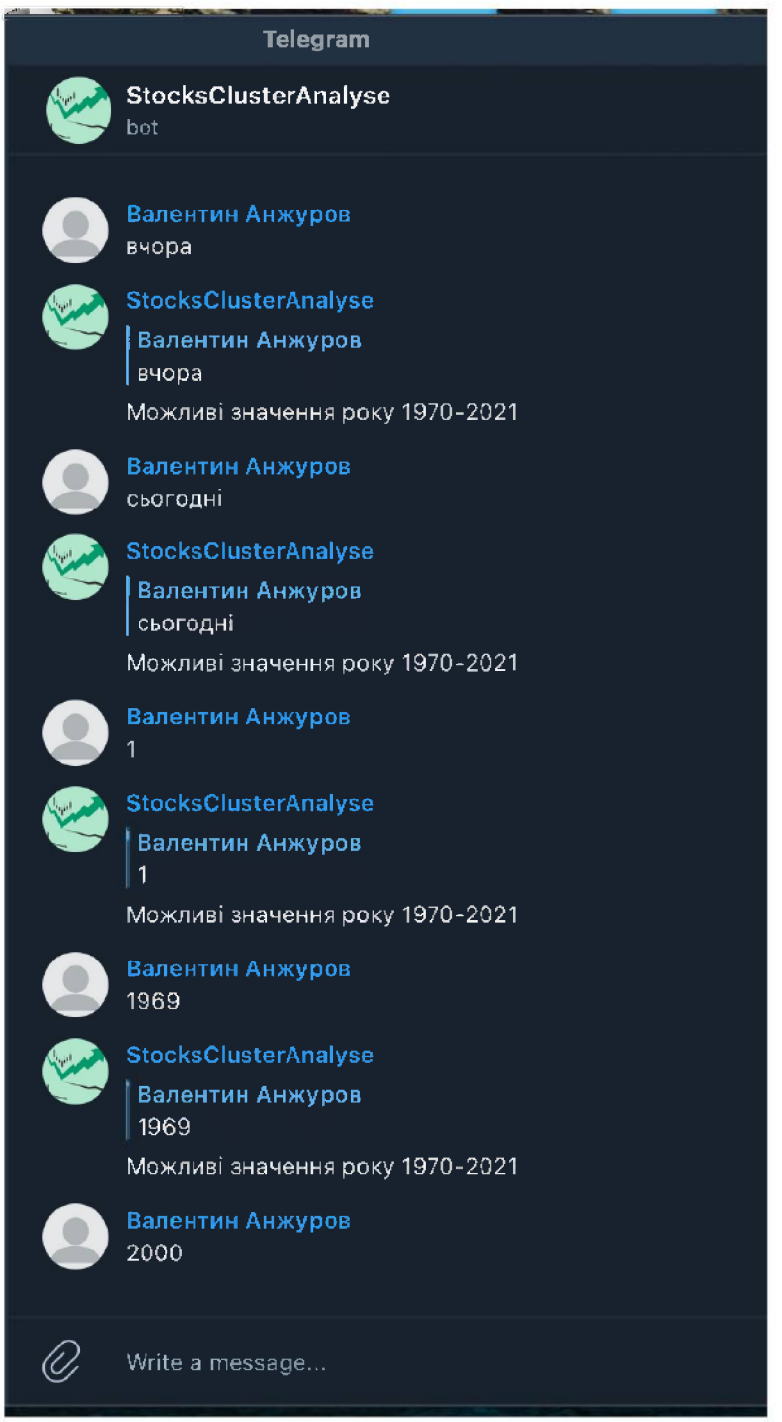



Рисунок 3.1 – помилка валідації

Якщо програма валідує дані вірними, починається збір даних фондового ринку шляхом використання відповідних API. Він розділений на 2 етапи:

1) Збір тикерів компаній (ідентифікаторів)
2) Збір P/E значень компанії

Далі починається процес кластеризації зібраних даних і формування вихідного файлу, після чого він відправляється назад користувачеві на його запит.

Дану модель можна запустити наступними шляхами:

1) Запуск на власному комп'ютері через:
    a. Командну строку. Для цього потрібно мати встановлений Python інтерпретатор на машині. Далі потрібно перейти до директорії з основним файлом програми і виконати відповідну команду запуску скрипту [**Error! Reference source not found.**]. Для забезпечення можливістю користування системи іншими користувачами, необходно передавати цілий проект на інші комп'ютери тим чи іншим чином. Сумуючи це, робимо висновок, що даний спосіб є незручним для користування
    b. Виконавчий файл. Мова Python має засоби для упаковки цілого проекту в один виконавчий файл з розширенням, залежним від ОС користувача. Даний спосіб також не є дуже зручним через те, що повинно узнавати ОС користувача і формувати відповідний файл саме для цього користувача
2) Розмістити систему на хостингу. Даний спосіб показав себе найзручнішим через наступні причини:
    a. Незалежність від типу ОС користувача

b. Незалежність від ресурсів ОС користувача, так як всі процеси працюють на потужній системі хостингу

c. Деякі хостингу не беруть плату за розміщення програми з невеликою кількістю користувачів

### 3.3 Результат роботи системи

В цьому пункті показаний приклад роботи комп'ютерної моделі та результати її роботи. Запустимо систему на дамо їй на вхід поточний рік роботи фондового ринку (Див. Рис. 3.2)

Показниками ефективності роботи моделі були вибрані час та точність даних. Час є основним фактором роботи на фінансовій біржі, але швидкість роботи не повинна компенсуватися неточними або, взагалі, втраченими даними

Так як аналогами моделі цієї кваліфікаційної роботи є Interactive Brokers та Ukraine Finance, порівняння проводяться з ними. Середній час роботи системи складає 3 хвилини, що є оптимальним результатом для складного процесу, що обробляє велику кількість інформації. Через приблизно 3 хвилини користувач отримує вихідний файл (Див. Рис. 3.3)

Вихідний файл містить 3 кластери, підкрашені відповідними кольорами. Зелений – рекомендовані компанії до інвестування, жовті – залишені на розсуд інвестора, червоні - не рекомендовані до розгляду інвестором. Більш детальний зміст вихідного файлу наведено в п. 1.6

Беручі в увагу результати використання аналогів, можна сказати, що модель виявилася більш ефективною, тому що результати аналізу співпадають на 95%, але час виконання дорівнює 3 хвилини, в той час коли результат роботи аналогів складає близько 15 хвилин

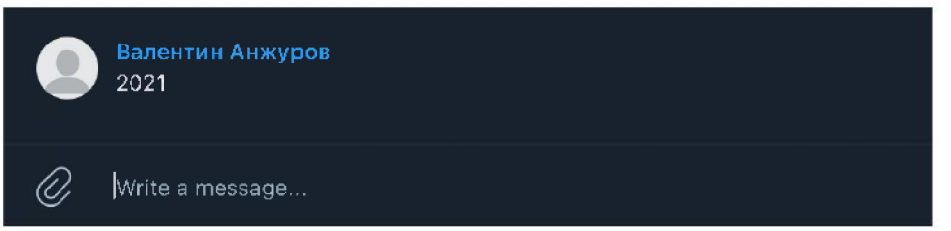



Рисунок 3.2 – передача вхідного значення

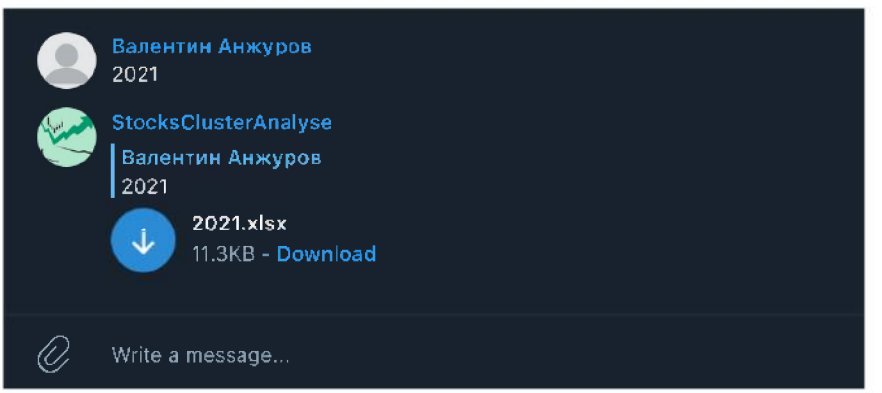



Рисунок 3.3 – отримання вихідного файлу

**Висновки до розділу 3**

В даному розділі розглянуті мінімальні системні вимогу, яким потрібна відповідати машина, на якій працюватиме комп'ютерна моделі аналізу індексів акцій на фондовому ринку.

Також, детально розписана послідовність дій, які виконує моделі для проведення.

Розписана інструкція для користувача, що матиму змогу проводити аналіз індексів

Сумуючі, можна зробити висновок, що моделі працює оптимальним шляхом та є дуже ефективною, беручі в увагу час, за який проводиться аналіз

# ВИСНОВКИ

В першому розділу були розглянуті основні поняття Data Mining, методи препроцесінгу та більше детально розглянутий метод кластериазіції – метод k-середніх. Додатково, були розглянуті використані засоби для створення комп'ютерної моделі

В другому розділу були розглянуті та проаналізовані основні моменти створення комп'ютерної моделі препроцесінгу, а саме її структуру, призначення кожної з частин моделі, структуру вхідних та вихідних даних.

В третьому розділі було описане створеної моделі та інструкція для користувача, наведені мінімальні системні вимоги моделі. Також були розглянуті проміжні та кінцеві результати роботи моделі

В ході даної роботи були розглянуті концепції препроцесінгу Data Mining, була розроблена модель препроцесінгу даних. Вивчаючи теоретичний матеріал, стало зрозуміло, що дана галузь є дуже перспективною та має право на те, щоб приділити їй увагу. Що стосується самої моделі, то вона є достатньо оптимізованою та придатною для розширення. На момент завершення роботи, модель може кластеризувати дані про студентів та виводити проміжну інформацію про стани кластерів. Серед можливостей покращення:

- Розпаралелювання алгоритму обробки даних, що дасть відчутний виграш у часі.
- Переробка моделі в універсальну. Можна зробити модель такою, що вона буде кластеризувати дані незалежно від області цих даних. Вона буде приймати на вхід таблицю з даними та назву колонки, дані з якої будуть оброблені методом кластеризації

Сумуючи, можна сказати, що мета роботи повністю досягнута. Методи були розглянуті, виявлені їх сильні та слабкі сторони.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Что такое Data Mining: [Електронний ресурс] // Mad Data, 2019, URL: https://maddata.agency/mad-blog/chto-takoe-data-mining (Дата звернення: 13.10.2021)
2. Data Preprocessing in Data Mining: [Електронний ресурс] // Geek Forces, 2019, URL: www.geeksforgeeks.org/data-preprocessing-in-data-mining/ (Дата звернення: 10.10.2021)
3. Real-World Machine learning: [Електронний ресурс] // Manning Free Content Center, 2015, URL: https://freecontent.manning.com/real-world-machine-learning-pre-processing-data-for-modeling/ (Дата звернення: 15.10.2021)
4. Data Mining – Cluster Analysis: [Електронний ресурс] // Tutorials Point, 2019, URL: www.tutorialspoint.com/data_mining/dm_cluster_analysis.htm (Дата звернення: 21.10.2021)
5. Clustering methods Data Scientists need to know: [Електронний ресурс] // Toward Data Science, 2018, URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 (Дата звернення: 27.10.2021)
6. Pros and Cons of K-means Clustering: [Електронний ресурс] // Pros And Cons, 2020, URL:https://www.prosancons.com/education/pros-and-cons-of-k-means-clustering/ (Дата звернення: 30.10.2021)
7. K-means Clustering: [Електронний ресурс] // Toward Data Science, 2018, URL: www.towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a (Дата звернення: 07.11.2021)
8. K-means Advantages and Disadvantages: [Електронний ресурс] // Clustering in Machine Learning, 2019, URL: https://developers.google.com/machine-

learning/clustering/algorithm/advantages-disadvantages (Дата звернення: 09.11.2021)

9. Python popularity chart: [Електронний ресурс] // Google Trends, 2020, URL: https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F05z1_,%2Fm%2F07sbkfb,%2Fm%2F02p97,%2Fm%2F0jgqg (Дата звернення: 12.11.2021)
10. Python vs Other programming languages: [Електронний ресурс] // STX Next, 2017, URL: https://stxnext.com/python-vs-other-programming-languages/ (Дата звернення: 22.11.2021)
11. Pandas library: [Електронний ресурс] // DataFlair, 2018, URL: https://data-flair.training/blogs/advantages-of-python-pandas/ (Дата звернення: 25.11.2021)
12. Microsoft Excel для начинающих: [Електронний ресурс] // Office Guru, 2018, URL: http://www.office-guru.ru/excel/microsoft-office-excel-chto-eto-59.html (Дата звернення: 28.11.2021)
13. Мартин Р.,Мартин М. Принципы, паттерны и методики гибкой разработки на языке C#. М. : «ООО Символ Санкт-Петербург Москва», 2011. 139с.
14. Черемных С. В., Семенов И. О., Ручкин В. С. Меделирование и анализ систем. IDEF-технологии. М. : «Финансы и статистика», 2006. 6с.
15. Data Serialization: [Електронний ресурс] // Guide to Python, 2019, URL: https://docs.python-guide.org/scenarios/serialization/(Дата звернення: 30.11.2021)
16. Database server: [Електронний ресурс] // Wikipedia, 2020, URL: https://en.wikipedia.org/wiki/Database_server (Дата звернення: 30.11.2021)
17. McKinney W. Python for Data Analysis. M. : «O`REILLY», 2012. 115c.

18. Parsingin Python: [Електронний ресурс] // Software Architect, 2017, URL: https://tomassetti.me/parsing-in-python/ (Дата звернення: 30.11.2021)

19. Minimum system requirements for Python programming: [Електронний ресурс] // Quora, 2019, URL: https://www.quora.com/What-are-the-minimum-hardware-requirements-for-python-programming (Дата звернення: 20.10.2021)

20. Using Python on Windows: [Електронний ресурс] // Python docs, 2019, URL: https://docs.python.org/3.3/using/windows.html (Дата звернення: 15.10.2021)

21. Create executable from python script: [Електронний ресурс] // Data to Fish, 2020, URL: https://datatofish.com/executable-pyinstaller/ (Дата звернення: 16.10.2021)

22. S&P500 компанії: [Електронний ресурс] // Investopedia, 2019, URL: https://www.investopedia.com/terms/s/sp500.asp (Дата звернення: 17.10.2021)

23. PE index: [Електронний ресурс] // Investopedia, 2019, URL: https://www.investopedia.com/terms/p/price-earningsratio.asp (Дата звернення: 17.10.2021)

# ДОДАТКИ

**Додаток А**

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Магістр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 –Комп'ютерна інженерія

**ЗАТВЕРДЖУЮ**
**Завідувач кафедри теоретичної та прикладної системотехніки**
**_____ д.т.н., проф. Шматков С. І.**

**«____» _____ 20___ року**

**З А В Д А Н Н Я**
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**Анжурова Валентина Євгеновича**
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Комп'ютерна модель аналізу індексів акцій на фондовому ринку»**

керівник роботи **Толстолузька Олена Геннадіївна, д.т.н., с.н.с.**,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "10" 03 2021 року № 0210-05/1804

2. Строк подання студентом роботи 19.11.2021

3. Перелік питань, які потрібно розробити)
   1. Аналіз класичних збірників літератури про фондовий ринок, його процеси та чинники впливу
   2. Дослідження існуючих індексів акцій
   3. Вивчення та практика роботи клієнта та сховища даних про поточні показники акцій ринку
   4. Побудова моделі, що працює на основі мови Python та TelegramBotAPI
   5. Тестування та налагодження роботи побудованої моделі

4. План роботи

| № з/п | Назви етапів роботи | Термін виконання етапів роботи |
|---|---|---|
| 1. | Аналіз класичних збірників літератури про фондовий ринок, його процеси та чинники впливу | 05.10.2020 – 05.03. 2021 |
| 2. | Дослідження існуючих індексів акцій | 09.03.2021– 30.04.2021 |
| 3. | Вивчення та практика роботи клієнта та сховища даних про поточні показники акцій ринку | 03.05.2021– 02.07.2021 |
| 4. | Побудова моделі аналізу індексів | 05.07.2021– 03.09.2021 |
| 5. | Тестування та налагодження роботи побудованої моделі | 06.09.2021– 08.10.2021 |
| 6. | Розробка пояснювальної записки | 11.10.2021– 26.11.2021 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

5. Дата видачі завдання 15.02.2020

Студент Анжуров В. Є. _____
підпис

Керівник роботи Толстолузька О. Г. _____
підпис

**Додаток Б**

Затверджую

_____

«_____»_____2021 р.

**Технічне завдання**
**на розробку програмного виробу «Комп'ютерна модель аналізу індексів акцій на фондовому ринку»**

| | | |
|---|---|---|
| 1. | Введення | Назва: «Комп'ютерна модель аналізу індексів акцій на фондовому ринку»<br>Область застосування: економічні заклади аналізу фондового ринку |
| 2. | Підстава для розробки | Навчальний план ФКН за фахом 123 – Комп'ютерна інженерія.<br>1. Завдання на кваліфікаційну роботу магістра Наказ Харківського національного університету імені В. Н. Каразіна 0210-05/1804 від 10.03.2021; |
| 3. | Призначення розробки | **Мета роботи:** підвищення ефективності аналізу індексів акцій на фондовому ринку за рахунок використання комп'ютерної моделі кластеризації даних компаній на фінансовій біржі методом k-середніх<br>**Призначення розробки:** розробка комп'ютерної моделі аналізу індексів акцій на фондовому ринку.<br>1. Створення додатку на платформі Telegram месенджеру<br>2. Модель призначена для аналізу індексів акцій фондового ринку |

| | | |
|---|---|---|
| | | 3. Заходи аналізу результатів препроцесінгу даних фондового ринку комп'ютерною моделлю |
| 4. | Технічні вимоги до програмного виробу | Програма повинна:<br>1. Представляти з себе Telegram Bot у месенджері Telegram<br>2. Надавати можливість провести аналіз індексів акцій за будь-який рік функціонування ринку<br>3. Забезпечувати пошук на аналіз даних вихідного файлу<br>4. Апаратна сумісність: надати для роботи будь-який пристрій, що підтримує Telegram месенджер<br>5. Вимоги до маркування та упаковки (не висуваються)<br>6. Вимоги до транспортування і зберігання (не висуваються)<br>7. Спеціальні вимоги (не висуваються) |
| 5. | Вимоги до програмної документації | Програмною документацією щодо розроблюваного програмного продукту вважати:<br>1. Справжнє технічне завдання на розробку програми представити в Додатку Б пояснювальної записки до кваліфікаційної роботи.<br>2. Програму і методику випробувань розробленої програми представити в Додатку В пояснювальної записки до кваліфікаційної роботи.<br>3. Опис програмного виробу, представити 3-ім розділом роботи<br>4. Лістінг програми представити у Додатку Г пояснювальної записки до кваліфікаційної роботи. |
| 6. | Техніко-економічні показники | Орієнтовна оцінка ефективності кваліфікаційної роботи: не потрібна. |

| | | | |
|---|---|---|---|
| 7. | Стадії і етапи розробки | Дата | Назва етапу |
| | | 3.09. 2021 – 10.09.2021 | Написання технічного завдання. |
| | | 13.09. 2021 – 24.09.2021 | Аналіз літератури за темою роботи, щодо існуючих засобів програмних рішень |
| | | 27.09. 2021 – 1.10.2021 | Проектування архітектури моделі |
| | | 4.10. 2021 – 15.10.2021 | Розробка моделі аналізу індексів акцій на фондовому ринку |
| | | 18.10. 2021 – 29.10.2021 | Тестування та апробація комп'ютерної моделі. |
| | | 1.11. 2021 – 19.11.2021 | Розробка інструкцій користувача. |
| | | 22.11. 2021 – 26.11.2021 | Розробка пояснювальної записки до роботи. |
| 8. | Порядок контролю імоделі | 1. Перевірка ходу розробки додатку виконувати раз в 2 тижні<br>2. Випробування програмного продукту проводити відповідно до програми та методики випробувань на базі комп'ютерного класу.<br>3. Захист розробленої моделі провести на засіданні Атестаційної комісії.<br>4. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску. | |

Виконавець

студент групи КІ- 61
Анжуров В. Є.

_____

Замовник

д.т.н., с.н.с.
Толстолузька О.Г

_____

**Додаток В**

Затверджую

_____

«_____»_____ 2021р.

# ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ

## програмного виробу «Комп'ютерна модель аналізу індексів акцій на фондовому ринку»

### 1. Об'єкт випробувань

Найменування випробуваного програмного виробу: модель аналізу індексів акцій на фондовому ринку

Область його застосування: економічні заклади аналізу фондового ринку

### 2. Мета випробування

Перевірка працездатності виробу та правильності роботи розробленої моделі. Підтвердження характеристик моделі вимогам, які сформульовані в ТЗ (Додаток Б).

### 3. Вимоги до програми

Програма повинна:

8. Представляти з себе Telegram Bot у месенджері Telegram
9. Надавати можливість провести аналіз індексів акцій за будь-який рік функціонування ринку
10. Забезпечувати пошук на аналіз даних вихідного файлу
11. Апаратна сумісність: надати для роботи будь-який пристрій, що підтримує Telegram месенджер

12.Вимоги до маркування та упаковки (не висуваються)

13.Вимоги до транспортування і зберігання (не висуваються)

14.Спеціальні вимоги (не висуваються)

**4. Вимоги до програмної документації**

Програмною документацією щодо розроблюваного програмного продукту вважати:

1. Справжнє технічне завдання на розробку програми представити в Додатку Б пояснювальної записки до кваліфікаційної роботи.
2. Програму і методику випробувань розробленої програми представити в Додатку В пояснювальної записки до кваліфікаційної роботи.
3. Опис програмного виробу, представити 3-ім розділом роботи
4. Лістінг програми представити у Додатку Г пояснювальної записки до кваліфікаційної роботи.

**5. Засоби випробувань**

Необхідна наявність пристрою з підтримкою Telegram месенджеру

**6. Програма і методика випробувань**

Випробування проведене в два етапи:

- ознайомчий (1-й етап);

- власне випробування програмного вироби (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірка відповідності та актуальності програмної документації за критерієм наявності зазначеної в ТЗ документації.

2) Перевірка якості документації на відповідність до стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає:

1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
2) Перевірку ступеня виконання функціональних вимог до програми.
3) Методику проведення перевірок:

3.1) Встановити програму для роботи з Excel

Даний тест вважається пройденим, якщо серед програм є Excel (Рис. 1)

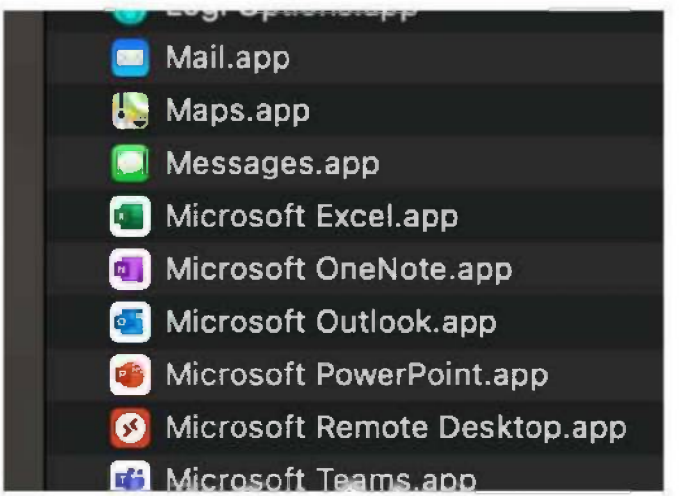



Рис. 1 – наявність Excel серед програм

3.2) Порядок проведення випробувань:

- Виконується запуск моделі. Даний тест вважати пройденим, якщо в Telegram знайдено бота (Рис. 2)

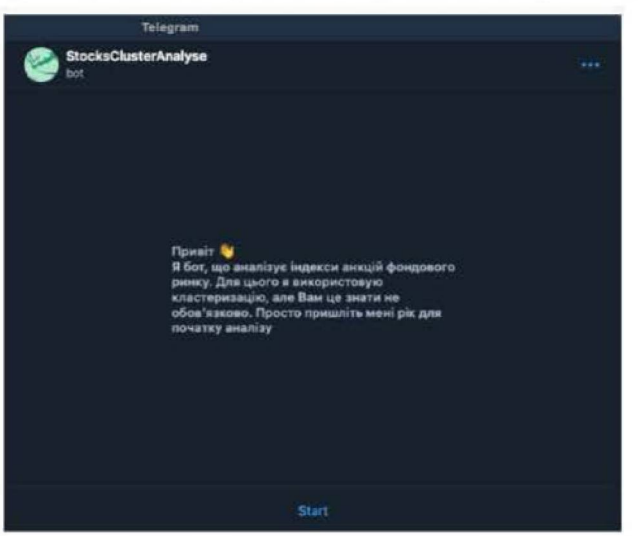



Рис. 2 – привітання бота

- На вхід подається рік роботи. Даний текст вважати пройденим, якщо не виникає помилок в консолі

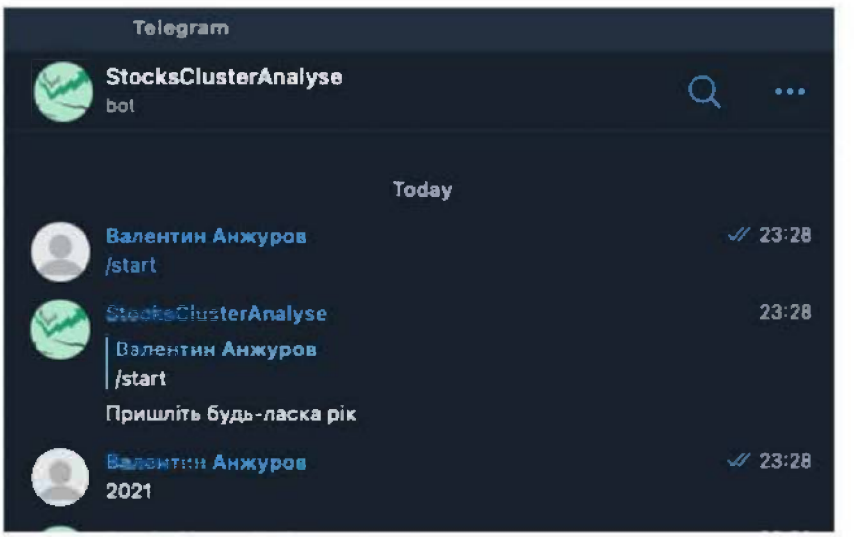



Рис.3. – успішне прийняття даних ботом

- Дані обробляються моделлю

- Створений вихідний файл повинен мати 1 сторінку з проаналізованими даними фондової біржі. Тест пройдений, якщо файл має наступний контент на Рис. 4

| | |
|---|---|
| ABC | 12,8 |
| BWA | 12,87 |
| MS | 12,95 |
| NOC | 13 |
| NWL | 13,16 |
| BK | 13,39 |
| LMT | 13,42 |
| WFC | 13,45 |
| STT | 13,48 |
| GPS | 13,79 |
| CZR | 13,99 |
| TPR | 14,16 |
| BAC | 14,21 |
| SNA | 14,54 |
| ADM | 14,64 |

Рис.4 – контент вихідного файлу

**Додаток Г**

## Лістинги коду

Лістинг Г.1 – функція, формуюча кластери

```
def cluster_data(data: list[dict], out_file_path: str) -> None:
    def _to_paint(row):
        if row.closest == 1:
            return pd.Series('background-color: green', row.index)
        if row.closest == 2:
            return pd.Series('background-color: yellow', row.index)
        else:
            return pd.Series('background-color: red', row.index)

    df = pd.DataFrame(data)
    min_pe = df["pe"].min()
    max_pe = df["pe"].max()

    centroids = _define_centroids(min_pe, max_pe)
    _assignment(centroids, "pe", df)
    while True:
        closest_centroids = df[CLOSEST_COLUMN].copy(deep=True)
        centroids = _refresh_centroids_with_column(centroids, "pe",
df)
        _assignment(centroids, "pe", df)

        if closest_centroids.equals(df[CLOSEST_COLUMN]):
            break

    sorted_pe = df.sort_values(by=["closest", "pe"])
    styled = sorted_pe.style.apply(_to_paint, axis=1)
    styled.to_excel(out_file_path,   engine='openpyxl',   index=False,
columns=["ticker", "pe"])
```

Лістинг Г.2 – функція, що визначає нові цетроїди

```
def _define_centroids(_min, _max):
    new_centroids = list(sorted(
        [random.uniform(_min, _max)
         for i in range(CLUSTERS_COUNT)]
    ))
    return dict(enumerate(new_centroids, 1))
```

Лістинг Г.3 – запуск Telegram Bot

```
server = Flask(__name__)
TOKEN = os.environ["TELEGRAM_BOT_TOKEN"]

bot = telebot.TeleBot(TOKEN, parse_mode=None)


@bot.message_handler(["start"])
def handle_text(message):
    bot.reply_to(message, "Пришліть будь-ласка рік")


@bot.message_handler(func=lambda m: True)
def handle_text(message):
```

```
    def _is_year_valid(text: str) -> bool:
        try:
            _year = int(text)
            return 1970 <= _year <= datetime.now().year
        except ValueError:
            return False

    year = message.text.strip()
    if _is_year_valid(year):
        out_file_path = run()
        with open(out_file_path, "rb") as xlsx:
            content = xlsx.read()
            bot.send_document(
                chat_id=message.chat.id,
                data=content,
                reply_to_message_id=message.id,
                visible_file_name=f"{year}.xlsx"
            )
    else:
        bot.reply_to(message, f"Можливі значення року 1970-
{datetime.now().year}")


bot.remove_webhook()
bot.infinity_polling()
```