**Міністерство освіти і науки України**

**Чернівецький національний університет імені Юрія Федьковича**

**Навчально-науковий інститут фізико-технічних та комп’ютерних наук**

(повна назва інституту/факультету)

**Кафедра комп’ютерних наук**

(повна назва кафедри)

**ЗВІТ З ПРАКТИКИ**

**“Науково-дослідна робота на тему дипломної роботи”**

**Рівень вищої освіти - другий (магістерський)**

Студента 2-го курсу, групи 644

Спеціальність: 122 Комп'ютерні науки

Освітня програма: Інтелектуальний аналіз

даних в комп’ютерних інформаційних системах

(шифр і назва спеціальності)

Практикант: Максимович Микола Юрійович

(прізвище та ініціали)

Керівник від кафедри: Талах М.В. .

(прізвище та ініціали)

Керівник від ВНЗ: к.т.н., асистент Дворжак В.В.

(прізвище та ініціали)

Національна шкала\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів Оцінка: ECTS

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Чернівці – 2024

**ЗМІСТ**

[Календарний план проходження практики 3](#_Toc182994938)

[ВСТУП 2](#_Toc182994939)

[1. МЕТА І ІНДИВІДУАЛЬНЕ ЗАВДАННЯ ПРАКТИКИ 3](#_Toc182994940)

[2. ЗВІТ З НАУКОВО-ДОСЛІДНИЦЬКОЇ ЧАСТИНИ ПЕРЕДДИПЛОМНОЇ ПРАКТИКи 4](#_Toc182994941)

[2.1. Візія та межі проекту 4](#_Toc182994942)

[2.1.1. Бізнес-вимоги 4](#_Toc182994943)

[2.1.2. Перетворення бізнес-цілей на цілі машинного навчання. Визначення шаблону та підходів для застосування моделі машинного навчання 4](#_Toc182994944)

[2.2. Відбір даних. Збір та перевірка даних. 6](#_Toc182994945)

[2.3 Виділення особливостей (Feature extraction) 7](#_Toc182994946)

[2.4 Вибір алгоритму ML 8](#_Toc182994947)

[3. ЗВІТ З НАУКОВО-ПРАКТИЧНОЇ ЧАСТИНИ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ 9](#_Toc182994948)

[3.1 Моделювання предметної області 9](#_Toc182994949)

[Функціональні компоненти та процеси системи 12](#_Toc182994950)

[3.2. Проектування логічної структури системи 14](#_Toc182994951)

[Загальні принципи проектування та декомпозиція системи 14](#_Toc182994952)

[3.3 Інженерія даних 16](#_Toc182994953)

[3.4 Інтерфейс користувача 16](#_Toc182994954)

[ВИСНОВКИ 19](#_Toc182994955)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 20](#_Toc182994956)

# Календарний план проходження практики

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № з/п | Назви робіт |  | | | | | | Відмітки про виконання |
| **1** | **2** | **3** | **4** | **5** | **6** |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | Прибуття на підприємство. | + |  |  |  |  |  | Виконано |
|  | Вступна нарада та постановка цілей. | + |  |  |  |  |  | Виконано |
|  | Вивчення вимог і ресурсів для виконання дипломного проекту. | + | + |  |  |  |  | Виконано |
|  | Аналіз наукової літератури та аналогічних рішень. |  | + |  |  |  |  | Виконано |
|  | Підготовка технічного завдання на виконання проекту. |  | + |  |  |  |  | Виконано |
|  | Збір та обробка даних для дипломного проекту. |  | + | + |  |  |  | Виконано |
|  | Створення базової моделі для зниження розмірності даних. |  |  | + | + |  |  | Виконано |
|  | Оптимізація та налагодження розроблених алгоритмів. |  |  | + | + | + |  | Виконано |
|  | Тестування та порівняння результатів. |  |  |  | + | + |  | Виконано |
|  | Внесення корективів у проект. |  |  |  | + | + |  | Виконано |
|  | Оформлення проміжного звіту про хід виконання. |  |  |  | + | + |  | Виконано |
|  | Завершення практичної частини дипломного проекту. |  |  |  |  | + |  | Виконано |
|  | Підготовка звіту за результатами практики. |  |  |  |  | + | + | Виконано |
|  | Захист звіту та отримання індивідуальної оцінки. |  |  |  |  |  | + | Виконано |

# ВСТУП

Підготовка до переддипломної практики є важливим етапом навчання, що дозволяє систематизувати набуті знання, поглибити навички та впровадити наукові досягнення, отримані в рамках курсової і дипломної роботи. Під час практики я мав змогу зосередитися на поглибленому вивченні та реалізації теоретичних і практичних аспектів теми магістерської роботи, яка присвячена зниженню вимірності даних для підвищення ефективності алгоритмів машинного навчання. Основний акцент було зроблено на реалізації алгоритмів скорочення даних і оцінці їхнього впливу на продуктивність моделей.

Моя курсова робота заклала основу для поглибленого розуміння методів зменшення вимірності, таких як Principal Component Analysis (PCA), Truncated Singular Value Decomposition (SVD), Autoencoder, FastICA, TSNE та інші. У рамках магістерської роботи ці знання були розширені й адаптовані з метою створення комплексної системи для аналізу та зменшення вимірності зображень, з урахуванням специфіки машинного навчання.

Практичний аспект полягав у створенні функціонального застосунку, який об’єднує методи зменшення вимірності з оцінкою їхнього впливу на якість і швидкодію моделей машинного навчання. У процесі роботи було розроблено модулі для обробки даних, забезпечено багатофункціональний інтерфейс користувача та інтегровано інструменти для візуалізації результатів, що робить програму зручною для практичного застосування та досліджень.

Таким чином, переддипломна практика стала важливим кроком у підготовці до магістерської роботи, дозволивши не лише закріпити теоретичні знання, але й реалізувати їх у вигляді практичного інструменту для вирішення актуальних задач машинного навчання.

# 1. МЕТА І ІНДИВІДУАЛЬНЕ ЗАВДАННЯ ПРАКТИКИ

Метою переддипломної практики є поглиблення знань, отриманих в процесі навчання, розвиток навичок у застосуванні методів скорочення даних для підвищення ефективності машинного навчання, а також перевірка теоретичних концепцій у реальних умовах розробки. Практика передбачала розробку функціональних компонентів, які базуються на застосуванні алгоритмів скорочення вимірності, оцінці їхньої ефективності та забезпеченні інтеграції даних у практичний застосунок.

Індивідуальне завдання полягало у детальному аналізі і порівнянні моделей скорочення даних, таких як PCA, SVD, Autoencoder та інших, а також у розробці програмного забезпечення, що дозволяє проводити зменшення вимірності зображень і надавати оцінку якості класифікаційних моделей, використовуючи такі алгоритми, як MLPClassifier, SVC, RandomForestClassifier та DecisionTreeClassifier.

Особливий акцент робився на реалізації можливості налаштування параметрів як для алгоритмів скорочення вимірності, так і для моделей оцінки, що дозволяє гнучко підходити до аналізу різних наборів даних і тестувати вплив параметрів на кінцевий результат. Крім того, завдання включало розробку функціональних елементів для візуалізації результатів роботи алгоритмів, таких як графіки залежності точності класифікації від зменшення вимірності або часу обчислень.

Таким чином, мета практики й індивідуальне завдання були орієнтовані на створення інструменту, який поєднує сучасні алгоритми скорочення вимірності з ефективними засобами їхньої оцінки, що має значний потенціал для застосування у наукових дослідженнях і практичній діяльності.

# 2. ЗВІТ З НАУКОВО-ДОСЛІДНИЦЬКОЇ ЧАСТИНИ ПЕРЕДДИПЛОМНОЇ ПРАКТИКи

## 2.1. Візія та межі проекту

### 2.1.1. Бізнес-вимоги

В сучасному світі, де обсяг даних зростає експоненційно, ефективне зменшення розміру зображень без значної втрати якості є важливим завданням для багатьох галузей, таких як телекомунікації, мультимедіа та Інтернет речей. Редукція розміру зображень дозволяє ефективніше зберігати, передавати та обробляти візуальні дані, заощаджуючи місце на диску, пропускну здатність мережі та обчислювальні ресурси. Особливо важливою є редукція зображень для подальшого використання у навчанні моделей машинного навчання, оскільки вона дозволяє зменшити обсяг даних, що підлягають обробці. Це прискорює процес навчання моделей, забезпечуючи ефективніше використання обчислювальних ресурсів. Крім того, даний підхід сприяє більш швидкому та ефективному розвитку систем машинного навчання, підвищуючи їх продуктивність і здатність до обробки великих обсягів даних.

### 2.1.2. Перетворення бізнес-цілей на цілі машинного навчання. Визначення шаблону та підходів для застосування моделі машинного навчання

Основною метою проекту є створення ковеєру для редукції зображень з використанням алгоритмів SVD (Singular Value Decomposition) та автоенкодерів. SVD є потужним інструментом для зниження розмірності даних, в той час як автоенкодери - це нейронні мережі, які навчаються кодувати вхідні дані в компактне представлення з подальшим відновленням оригінальних даних з цього представлення.

Для розробки конвеєру редукції зображень доцільно обрати підхід розробки на основі моделі (Model-First Development). Це зумовлено наступними міркуваннями:

1. В даному випадку ключовим компонентом є саме модель машинного навчання, навколо якої буде розроблятися система. Тому має сенс зосередитися спочатку на розробці та оптимізації самої моделі.
2. Стратегія редукції даних безпосередньо стосується моделі машинного навчання та впливає на її продуктивність і точність. Спочатку необхідно дослідити та визначити ефективну стратегію редукції даних для моделі.
3. На цьому етапі може бути складно повністю спланувати системні вимоги та архітектуру, оскільки невідомо, якої точності та продуктивності можна досягти з оптимізованою моделлю після застосування стратегії редукції даних.
4. Зосередження на моделі на ранньому етапі дозволить уникнути значних інвестицій у проєкт, який може виявитися нездійсненним, якщо цілі щодо продуктивності та точності не будуть досягнуті.

Після того, як буде розроблена та протестована ефективна стратегія редукції даних для моделі, команда зможе перейти до системного проєктування та врахувати ширші вимоги щодо користувацького досвіду, безпеки, справедливості тощо під час розробки системи навколо оптимізованої моделі.

Для реалізації розробки пайплайну багатоетапної стратегії множинної редукції даних планується використання наступних шаблонів:

Вкладення (Representation Learning). Цей шаблон дозволить отримати компактні векторні представлення високорозмірних вхідних даних, зберігаючи при цьому їхні ключові характеристики та семантичну інформацію. Зокрема, буде використано згорткову автоенкодерну нейронну мережу для навчання на вхідних зображеннях. Енкодер цієї мережі виконуватиме стиснення зображень у компактні вектори вкладень (embeddings) нижчої розмірності, зберігаючи найважливіші візуальні ознаки. Вектори вкладень можна отримати з передостаннього (або іншого прихованого) шару енкодера, який міститиме стисле представлення кожного зображення у вигляді низьковимірного векторного коду. Такий підхід дозволить значно зменшити обчислювальну складність та вимоги до пам'яті для наступних етапів обробки даних. Під час редукції даних часто виникає проблема великої кількості унікальних категоріальних значень, що збільшує розмірність даних. Використання вкладень допоможе зберегти важливу семантичну інформацію в стислому векторному форматі, спростивши подальшу обробку категоріальних ознак [1].

Налаштування гіперпараметрів (Hyperparameter Tuning). Оскільки стратегія редукції даних може включати декілька етапів та методів, зокрема автоенкодерну мережу для отримання вкладень, важливо ретельно підібрати оптимальні гіперпараметри для кожного з них. Це допоможе досягти балансу між ефективністю, точністю та відтворюваністю результатів на різних наборах даних. Налаштування гіперпараметрів стане критичним для забезпечення високої якості отриманих вкладень та подальшої обробки даних.

Такий набір шаблонів, що поєднує вкладення через автоенкодери для редукції розмірності, налаштування гіперпараметрів та оцінку якості вкладень, допоможе впоратися з типовими проблемами, що виникають при роботі з великими наборами даних високої розмірності, і забезпечити ефективну реалізацію багатоетапної стратегії редукції даних.

## 2.2. Відбір даних. Збір та перевірка даних.

Для демонстрації та тестування підходів до редукції зображень в рамках даного проекту буде використано відомий набір даних MNIST, який містить 70000 зображень рукописних цифр розміром 28х28 пікселів у градаціях сірого. Цей набір є класичним та широко використовуваним для задач комп'ютерного зору і машинного навчання, що забезпечує можливість порівняння з іншими дослідженнями.

Статистичні властивості даних MNIST:

* Роздільна здатність зображень: 28x28 пікселів
* Колірна модель: Градації сірого (від 0 до 255)
* Формат файлів: Вбудований бінарний формат набору MNIST
* Обсяг навчальних даних: 60000 зображень
* Обсяг тестових даних: 10000 зображень
* Загальна кількість класів (цифр): 10 (від 0 до 9)

Набір MNIST є добре сформованим, повним та репрезентативним для завдання розпізнавання рукописних цифр. Він широко використовується в дослідженнях машинного навчання як базовий набір для тестування та порівняння різних алгоритмів та підходів.

Вимоги до даних для редукції зображень:

* Дані повинні бути у форматі масиву або тензора, сумісному з бібліотеками глибокого навчання (TensorFlow, PyTorch тощо)
* Бажано мати окремі набори даних для навчання, валідації та тестування моделі
* Дані мають бути належним чином нормалізовані та стандартизовані
* Важливо мати метадані та документацію щодо походження даних та способу їх збору/генерації
* Дані повинні відповідати вимогам конфіденційності та безпеки для запобігання витоку чутливої інформації

Оцінка якості даних буде проводитись за наступними метриками:

* Повнота - відсутність відсутніх значень або пошкоджених зображень
* Точність - відповідність зображень змісту, що очікується (рукописні цифри)
* Послідовність - однакові роздільна здатність, розміри та формат зображень
* Репрезентативність - наявність у наборі усіх класів (цифр від 0 до 9)
* Незміщеність - відсутність значного дисбалансу класів

Якщо під час розробки моделі виникне потреба у використанні додаткових наборів даних, зокрема для трансферного навчання чи фіндтюнінгу. В такому разі буде проведено аналогічний аналіз та оцінку якості цих наборів даних.

## 2.3 Виділення особливостей (Feature extraction)

Для завдання редукції та відновлення зображень основними вхідними даними є самі зображення. Тому головним кроком при вилученні ознак буде перетворення зображень у матриці пікселів або тензори.

Більшість бібліотек машинного навчання (TensorFlow, PyTorch тощо) мають вбудовані функції для завантаження зображень та перетворення їх у числові формати, придатні для подальшої обробки. Наприклад, в TensorFlow є функція tf.io.decode\_image для декодування зображень з різних форматів у тензори.

Також було виконано попередню обробку зображень, таку як нормалізація пікселів до діапазону [0, 1] для полегшення процесу навчання моделей. Та перетворення розмірності зображень до векторів.

Лістинг 2.1 - Попередня обробка даних

# Нормалізація

x\_train = x\_train.astype('float32') / 255.0

x\_test = x\_test.astype('float32') / 255.0

# reshape to flat

x\_train\_flat = x\_train.reshape((len(x\_train), -1))

x\_test\_flat = x\_test.reshape((len(x\_test), -1))

Оскільки ми працюємо безпосередньо з пікселями зображень, немає необхідності в розробці спеціальних технік вилучення ознак. Матриці пікселів вже містять усю необхідну інформацію для навчання моделей редукції та відновлення зображень.

## 2.4 Вибір алгоритму ML

Для реалізації багатоетапної стратегії редукції зображень без значної втрати якості роботи основної моделі було використано комбінацію двох основних алгоритмів машинного навчання:

1. Сингулярне розкладання (SVD, Singular Value Decomposition): SVD є потужним лінійним методом для виявлення найбільш значущих компонент у даних. Застосування SVD до матриці пікселів зображення дозволяє знайти найбільш інформативні сингулярні вектори, які представляють основні патерни та структури в даних. Зберігаючи лише найбільш значущі сингулярні вектори, можна досягти ефективного стиснення зображення [5].
2. Автоенкодери: Автоенкодери є типом нейронних мереж, які навчаються кодувати вхідні дані (зображення) у компактне проміжне представлення (латентний вектор), а потім декодувати це представлення для відновлення наближення вхідних даних. Використання автоенкодерів дозволяє знаходити ефективні нелінійні представлення даних, які зберігають найбільш важливі ознаки.

# 3. ЗВІТ З НАУКОВО-ПРАКТИЧНОЇ ЧАСТИНИ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

## 3.1 Моделювання предметної області

В рамках розробки інформаційної технології оптимізації ресурсоємності класифікаційних моделей проведено детальний аналіз та моделювання предметної області. Система проектується як багатоетапний процес редукції даних з можливістю конфігурації параметрів та оцінки якості результатів.

Для візуалізації структури та взаємозв'язків компонентів системи розроблено дві ключові діаграми:

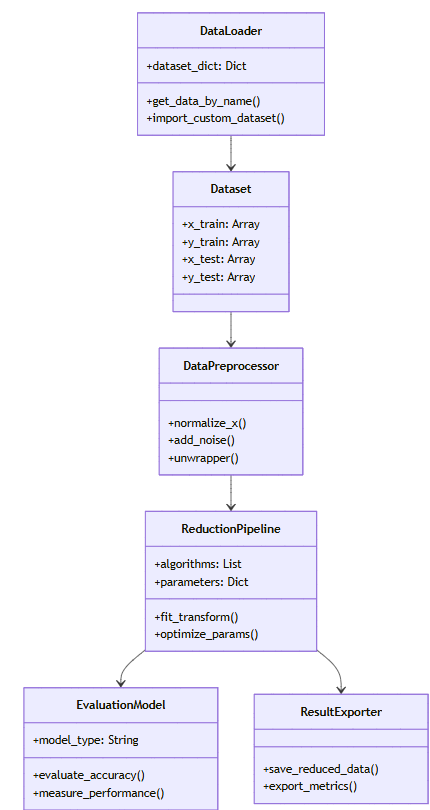


Рисунок 3.1 - Діаграма потоків даних системи

Діаграма відображає повний життєвий цикл обробки даних у системі, який складається з наступних етапів:

1. Підготовчий етап:
   * Завантаження вхідного набору даних (DataLoader).
   * Первинна обробка через Data Loader з формуванням навчальної та тестової вибірок.
   * Попередня обробка даних у Preprocessor, включаючи нормалізацію, шум, розгортання масиву у вектор.
2. Етап конфігурації, що включає три паралельних потоки користувацького введення:
   * Pipeline Configuration: визначення кількості етапів редукції (1-3)
   * Algorithms Selection: вибір конкретних алгоритмів для кожного етапу
   * Parameters Range: встановлення діапазонів параметрів оптимізації (мінімальне/максимальне значення та крок)
3. Етап оптимізації:
   * Формування пайплайну в Pipeline Constructor на основі користувацької конфігурації
   * Виконання процесу оптимізації параметрів (Parameter Optimization Process)
   * Візуалізація результатів оптимізації
   * Вибір користувачем фінальних параметрів на основі запропонованих оптимальних значень
4. Етап фінальної обробки:
   * Застосування Final Reduction Pipeline з обраними параметрами
   * Паралельна передача результатів на оцінку (Evaluation Model) та експорт (CSV Export)

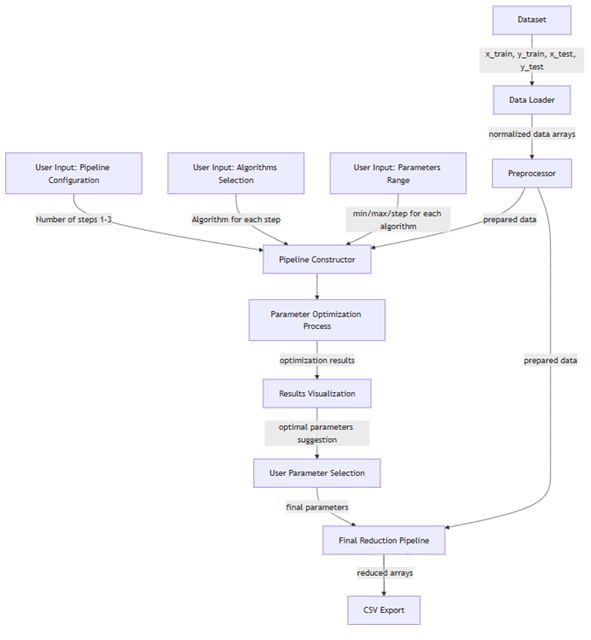


Рисунок 3.2 - UML діаграма класів системи

Діаграма демонструє основні програмні компоненти системи та їх взаємозв'язки:

1. DataLoader:
   * Відповідає за завантаження даних
   * Містить словник доступних датасетів (dataset\_dict)
   * Реалізує методи get\_data\_by\_name() та import\_custom\_dataset()
2. Dataset:
   * Зберігає навчальну та тестову вибірки у вигляді масивів
   * Забезпечує уніфікований формат даних для подальшої обробки
3. DataPreprocessor:
   * Реалізує методи попередньої обробки даних
   * Включає функції нормалізації (normalize\_x), додавання шуму (add\_noise) та перетворення форматів (unwrapper)
4. ReductionPipeline:
   * Центральний компонент системи
   * Зберігає список алгоритмів та їх параметри
   * Забезпечує методи fit\_transform() та optimize\_params()
5. ResultExporter:

Відповідає за збереження результатів у файл.

Розроблена модель предметної області забезпечує чітке розмежування відповідальності між компонентами системи, гнучкість у конфігурації параметрів та можливість розширення функціоналу. Використання патерну "Конвеєр" (Pipeline) дозволяє ефективно організувати послідовну обробку даних та оптимізацію параметрів алгоритмів редукції.

### Функціональні компоненти та процеси системи

Розроблена система реалізує модульний підхід до організації функціональних компонентів, що забезпечує чітке розмежування відповідальності та гнучкість у розширенні функціоналу. Архітектурно система розділена на два основні функціональні рівні: клієнтський та обробний.

Клієнтська частина представлена інтерфейсом користувача, реалізованим на базі бібліотеки Tkinter, що забезпечує всі аспекти взаємодії з системою. Цей рівень включає компоненти для вибору та завантаження зображень, інтерфейс налаштування параметрів редукції даних, модулі конфігурації методів обробки (PCA, TruncatedSVD, TSNE тощо), а також компоненти для візуалізації результатів та метрик якості обробки.

Обробний рівень охоплює всю функціональність, пов'язану з безпосередньою обробкою даних. Він містить модулі зниження розмірності, компоненти для трансформації та обробки даних, підсистему збереження результатів. Важливою особливістю є реалізація модульної структури, яка забезпечує можливість легкого розширення функціоналу та додавання нових методів обробки.

Процес обробки даних в системі організований як послідовність взаємопов'язаних етапів. На етапі передобробки відбувається завантаження даних з Keras datasets або користувацьких джерел, виконується нормалізація значень пікселів до діапазону [0, 1] та здійснюється векторизація зображень для подальшої обробки.

Наступний етап - конструювання ознак - включає перетворення матриць пікселів у вектори фіксованої довжини, формування латентних векторів через алгоритми редукції та масштабування даних відповідно до вимог моделей. За цим слідує етап безпосередньої редукції розмірності, де застосовуються один або декілька алгоритмів (PCA, SVD, t-SNE), виконується їх конфігурація через sklearn.pipeline та послідовне застосування обраних методів.

Завершальний етап включає оцінку якості редукції та експорт результатів. Тут проводиться обчислення метрик якості, візуалізація результатів для порівняння та експорт оброблених даних у CSV формат.

Запропонований підхід має ряд ключових переваг. Перш за все, це ефективне зменшення розмірності при збереженні важливої інформації та виділення значущих патернів в даних. Також важливим є отримання некорельованих ознак, що суттєво покращує подальшу класифікацію, та можливість інтерпретації отриманих латентних векторів.

Реалізація системи характеризується низкою важливих особливостей. Забезпечується гнучкість у виборі джерел даних та можливість детальної конфігурації параметрів на кожному етапі обробки. Система підтримує збереження проміжних результатів, що важливо для аналізу та відлагодження процесу редукції. Особлива увага приділена візуалізації метрик для оцінки якості, що дозволяє користувачам ефективно контролювати та оптимізувати процес обробки даних.

Така організація системи створює надійну основу для ефективної обробки та редукції даних, забезпечуючи при цьому гнучкість у налаштуванні та розширенні функціоналу відповідно до конкретних потреб користувачів.

## 3.2. Проектування логічної структури системи

### Загальні принципи проектування та декомпозиція системи

На етапі проектування логічної структури інформаційної технології оптимізації ресурсоємності класифікаційних моделей було застосовано багаторівневий підхід з чітким розділенням відповідальності між компонентами системи. Проектування здійснювалося з урахуванням вимог до гнучкості конфігурації, можливості розширення функціоналу та оптимізації взаємодії між компонентами.

Для забезпечення ефективної реалізації поставлених задач було розроблено трирівневу архітектуру системи (рисунок 3.4), яка включає:

1. Рівень інтерфейсу (Interface):

* Реалізує взаємодію з користувачем через графічні компоненти
* Забезпечує гнучке налаштування параметрів редукції
* Надає засоби візуалізації результатів оптимізації

1. Рівень обробки даних (Data Handling):

* Відповідає за уніфікований доступ до різних джерел даних
* Забезпечує попередню обробку та валідацію вхідних даних
* Реалізує перетворення форматів даних між компонентами

1. Рівень бізнес-логіки (Core Logic):

* Реалізує алгоритми редукції даних
* Забезпечує оптимізацію параметрів
* Координує процес багатоетапної обробки

1. Рівень оцінки (Evaluation):

* Здійснює оцінку якості редукції
* Забезпечує експорт та збереження результатів
* Формує метрики ефективності

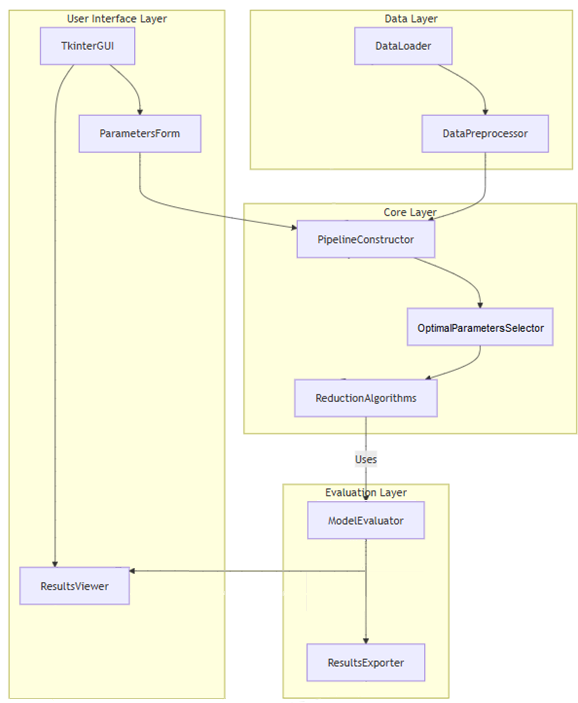


Рис3.3 Компонентна діаграма системи

Для деталізації взаємодії між компонентами системи розроблено компонентну діаграму (рисунок 3.5), яка демонструє:

* Механізми взаємодії через стандартизовані інтерфейси
* Потоки даних між компонентами
* Ієрархію залежностей компонентів

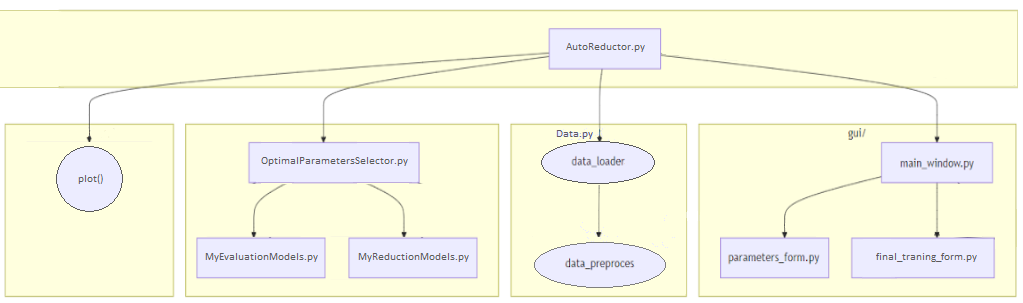


рис 3.4 Діаграма модулів системи

Фізична організація системи представлена через модульну структуру (рисунок 3.6), що забезпечує:

* Оптимальне розділення функціоналу між модулями
* Мінімізацію зв'язків між компонентами
* Можливість незалежної модифікації та тестування
* Спрощення процесу розробки та підтримки

## 3.3 Інженерія даних

В рамках розробки системи редукції зображень реалізовано повний цикл ETL (Extract, Transform, Load) процесу, що забезпечує ефективну обробку та трансформацію даних. Процес включає три послідовні етапи: екстракцію даних з різних джерел, їх трансформацію та завантаження для подальшого використання.

## 3.4 Інтерфейс користувача

Інтерфейс користувача розроблений для забезпечення інтуїтивної роботи з додатком. Основні елементи інтерфейсу включають:

* Головний екран: Пропонує користувачу вибір датасету зображень для обробки та спосіб класифікації, тобто модель оцінки редукції даних.

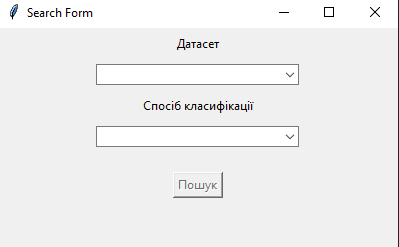


Рисунок 3.5 – Головний екран

* Форма вибору алгоритмів редукції: Дозволяє користувачу налаштовувати розмір конвеєра, від 1 до 3, та обирати методи зниження розмірності з доступного списку (PCA, TSNE, NMF, TruncatedSVD, FastICA, Autoencoder). Після вибору методу відкриваються параметри налаштувань для цього методу. А саме – мінмальний та максимальний розмір стиснення і кількість кроків між цими значеннями.

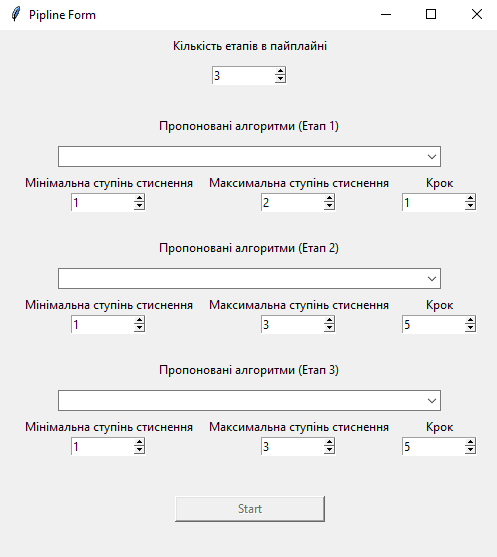


Рисунок 3.6 – Форма вибору алгоритмів редукції

* Панель відображення результатів: Після обробки даних зменшені результати відображаються у вигляді візуалізацій, графіків та/або збереження даних в окремий файл. Візуалізація допомагає користувачу проаналізувати якість зниження розмірності на основі зображення.
* Форма для створення власного пайплайну та отримання даних. В цій форма користувачу пропонується ввести параметри для моделей редукції які він повинен сам обрати базуючись на інформації наданій додатком на ппередньому кроці.

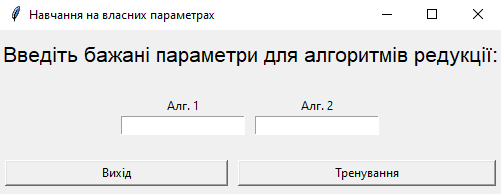


Рисунок 3.7 – Форма для створення власного пайплайну

Дизайн інтерфейсу зосереджений на забезпеченні зручного доступу до основних функцій, щоб користувач міг швидко отримати необхідний результат, не вдаючись до складних налаштувань.

# ВИСНОВКИ

Переддипломна практика дозволила інтегрувати теоретичні знання з практичними навичками розробки сучасного програмного забезпечення. У рамках практики було досягнуто значних результатів, які стали основою магістерської роботи, присвяченої створенню системи зниження вимірності даних для оптимізації моделей машинного навчання.

Було успішно реалізовано алгоритми скорочення вимірності: Principal Component Analysis (PCA), Truncated SVD, Autoencoder, FastICA, TSNE та NMF. Вони були інтегровані у програмний продукт для обробки зображень великої розмірності та підвищення ефективності моделей класифікації. Для оцінки результатів використовувалися моделі MLPClassifier, SVC, RandomForestClassifier та DecisionTreeClassifier. Гнучкість налаштування параметрів дозволила отримати детальну аналітику про вплив зниження вимірності на точність і швидкість класифікації.

Було розроблено інтуїтивний інтерфейс користувача та модулі візуалізації результатів, які дозволяють аналізувати ефективність кожного алгоритму та порівнювати їхні показники. Додано можливості створення графіків точності моделей, часу навчання та часу відповіді, що підвищує наочність застосунку.

Виконана робота поглибила знання у сфері машинного навчання та розширила досвід у розробці програмного забезпечення. Було отримано практичні навички роботи з алгоритмами аналізу даних, розробки модульного застосунку та створення документації.

Переддипломна практика виконала свою основну мету — підготувала основу для завершення магістерської роботи. Набуті знання забезпечують можливість подальшого розвитку та ефективного застосування результатів для вирішення реальних задач машинного навчання.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge: MIT Press, 2016. 775 с.
2. Chollet F. Deep Learning with Python. Shelter Island: Manning Publications Co., 2017. 361 с.
3. Hinton G. E., Salakhutdinov R. R. Reducing the dimensionality of data with neural networks. Science. 2006. Том 313, №. 5786. 504–507 с.
4. Lee, H., & Choi, S. (2018). Encoding convolutional neural networks into recurrent neural networks via autoencoder. IEEE Access, 6, 30097-30108.
5. Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. In Linear Algebra (с. 134-151). Springer, Berlin, Heidelberg.
6. Bourlard H., Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition. Biological cybernetics. 1988. Том 59, №.4-5. 291–294 с.
7. Ke, Q., & Kanade, T. (2005, June). Robust subspace computation using l1 norm. In IEEE Conference on Computer Vision and Pattern Recognition Workshops (с. 175-175).
8. Документація TensorFlow. [Електронний ресурс] - Режим доступу до ресурсу: https://www.tensorflow.org/learn (дата звернення: 10.04.2024).
9. Документація Keras. [Електронний ресурс] - Режим доступу до ресурсу: https://keras.io/guides/ (дата звернення: 10.05.2024).
10. Документація scikit-learn. [Електронний ресурс] - Режим доступу до ресурсу: https://scikit-learn.org/stable/ documentation.html (дата звернення: 10.05.2024).