

DataWise – ДатаВайс

**ТЕМА:** Приложение свързано с улесняване запознанството и задълбочаване в сферата на структурите от данни и алгоритми

**1. Автори**

**1.1. Алекс Ивайлов Стефанов**

ЕГН: 0745161445

Адрес: гр.Казанлък, ул. „Добри Кехайов“ №13

Телефон: 0889475177

Email: [rlgalexbgto@gmail.com](mailto:rlgalexbgto@gmail.com)

Училище: ППМГ „Никола Обрешков“

Клас: 11б

**2. Ръководител: Здравка Стефанова Димитрова**

Телефон: 0893422519

Email: [dimitrova@pmgkk.com](mailto:dimitrova@pmgkk.com)

Длъжност: Учител по информатика и информационни технологии

**3. Резюме**

**3.1. Цели:** Проектът "DataWise" е насочен към хора, които се стремят към кариера в програмирането. Той предоставя платформа за усъвършенстване на знанията по структури от данни и алгоритми – умения, които са от критична важност при интервюта в големи технологични компании (FANG: Facebook, Amazon, Netflix, Google).  
**\*Контекст:**  
Интервютата във FANG компаниите често изискват задълбочено познаване на структури от данни и алгоритми. Този проект цели да помогне на кандидатите да се подготвят по най-добрия начин, като им предостави достъп до модерни AI технологии в комбинация с традиционни образователни ресурси.

**3.2. Основни етапи в реализирането на проекта:**

**- Основни дейности:**

**\* Формулиране на идеята**

**\* Изграждане на архитектура**

**\* Конфигуриране на екосистемата**

**\* Изучаване на изкуствения интелект (AI)**

**\* Изграждане архитектурата на AI модела**

**\* Обучение по TypeScript и Front-End технологии**

**\* Обучение по Python**

**\* Изучаване на математическите основи**

**\* Правене на DataSet с тестовите задачи (~10 000 примера)**

**\* Имплементация на AI модела**

**\* Разработка на лого и цялостен дизайн**

**\* Тестване и оптимизация**

**\* Събиране на обратна връзка**

**\* Документация и публикуване**

**3.3. Ниво на сложност на проекта**

**Проектът се характеризира с множество слоеве на сложност – както от гледна точка на математическите изчисления, така и от гледна точка на интеграцията на различни технологични екосистеми. Трудността идва от това, че всяко друго приложение ползва библиотеки за разработка на AI продукти или са wrapper-и на съществуващи AI, с “DataWise” искам да покажа, че това не винаги трябва да е случеят, “DataWise.AI” е изцяло създаден от нищото(без библиотеки за AI), на чист Python. Когато всичко ти е дадено на готово е лесно и можеш да се разминеш дори и да не разбираш кода си, когато не ти е дадедно нищо обаче минаваш пред всички трудни стъпки (като програмният език, математиката зад обработването на данните и най-важното събирането на въпросните данни), но в крайна сметка се запознаш с целият процес и наистина разбираш как се работи с „Голям обем от данни“, защото човек трябва да е напълно запознат с това как изобщо да се сдобие с тях и как да ги превърне в нещо, което ще е от полза на хора. По-долу са изброени основните аспекти, като са дадени и подробни примери за математическите предизвикателства:**

* **Математически сложности:**
  + **Линейна алгебра и матрични операции:  
    За изграждането и обучението на AI модела са от съществено значение операции като умножение на матрици, транспониране и изчисляване на инверсии. Например, при предаването на данните през конволюционните слоеве се извършва тензорна операция (тензор-дот продукт), която комбинира информация от няколко измерения и осигурява правилното изчисляване на активациите.**
  + **Функция на активация и нейната диференциируемост:  
    Моделът използва функцията ReLU (Rectified Linear Unit). Тя не само преобразува входните данни, но и трябва да бъде диференцируема, за да може обратното разпространение на грешката (backpropagation) да изчисли производната и. Това включва и сложни математически концепции, като изчисляване на производни в множество точки и справяне с недефинирани стойности при граници.**
  + **Оптимизация и градиентен спуск:  
    Обучението на невронната мрежа става чрез оптимизационни алгоритми като стохастичния градиентен спуск (SGD). Това изисква постоянно изчисляване на градиенти, актуализиране на теглата и настройване на параметри като learning rate. Пример за математическо предизвикателство е адаптивното настройване на learning rate и използването на техники като momentum, за да се избегнат локални минимуми.**
  + **Обратна разпространение (Backpropagation):  
    При обратното разпространение се изчисляват производните на грешката спрямо всички параметри на мрежата. Това включва сложни вериги от диференциране, където се изисква коректно проследяване на градиентите през всяка невронна връзка. Важно е правилното пресмятане на градиентите за всяка операция, включително тези, извършвани в конволюционните слоеве.**
  + **Нормализация и регуларизация:  
    За да се подобри стабилността на обучението, се използват техники като нормализация (batch normalization) и регуларизация (L2-регуларизация). Тези методи добавят допълнителни математически изчисления, които трябва да бъдат интегрирани във функцията за загуба (loss function) и оптимизацията на модела.**
* **Данни (Data set)**
  + **Източник и авторски права:  
    Всички данни, необходими за работата на AI системата, са събрани лично от мен и са защитени с MIT лиценз.**
  + **Размер и структура:  
    Създадох dataset, съдържащ близо 10 000 примера, които са внимателно подбрани и категоризирани. Тези данни служат за обучение и валидиране на AI модела, осигурявайки богата и разнообразна тренировъчна база.**
  + **Категории:  
    Данните са разделени на следните 10 категории:**
    - **BFS (Обхождане в ширина):**
    - **DFS (Обхождане в дълбочина):**
    - **Two Pointers (Два указателя):**
    - **Dynamic Programming (Динамично програмиране):**
    - **Greedy Algorithm (Жадни алгоритми):**
    - **Backtracking (Обратна проследяемост):**
    - **Binary Search (Двоично търсене):**
    - **Disjoint Set (Несъвместими множества):**
    - **Game Theory (Теория на игрите):**
    - **N/A (Неопределено/Други**
* **Екосистеми и интеграция:**
  + **Angular (Front-End):  
    Фронтендът е разработен с Angular, който предоставя модерен, интерактивен потребителски интерфейс. Angular комуникира с бекенд услугите чрез RESTful API, като извиква endpoint-и за получаване на резултати от AI модела. Това осигурява динамично обновяване на данните и интерактивност за потребителя.**
  + **Flask (Python):  
    AI моделът и основната логика за обработка на данни са имплементирани с Python и Flask. Тук се извършват сложни математически изчисления, необходими за функционирането на AI модела, като обработка на текстови данни, конволюционни операции и изчисление на грейдиенти.**
  + **ASP.NET:  
    Допълнителни услуги и бизнес логика могат да бъдат реализирани чрез ASP.NET, като се интегрират с останалата част от системата чрез стандартизирани API endpoint-и. Тази част от екосистемата осигурява мащабируемост и стабилност при изпълнение на корпоративни приложения.**
  + **Свързаност на технологичните слоеве:  
    Проектът интегрира различни технологии – Angular, Flask и ASP.NET – които работят синхронно. Angular изпраща заявки към Flask, който обработва данните и изчислява резултатите на базата на сложни математически модели, а ASP.NET може да допълва системата с допълнителни услуги. Тази мулти-платформена интеграция изисква координация между различните езици и framework-и, което добавя допълнителна степен на сложност.**

**3.4 Логическо и функционално описание на решението**

**- Архитектура**

**Решението е разделено на различни проектни папки, като всяка от тях съдържа специфична функционалност, отговаряща за отделни аспекти на цялостната система. Този модулен подход осигурява гъвкавост, мащабируемост и лесна поддръжка. Ето структурата на проектните папки:**

* **DataWise.AI:  
  Този компонент съдържа AI модула, разработен на Python, който използва сложни математически изчисления за класификация на текстови задачи. Чрез Flask endpoint-и се обслужват заявките от останалите компоненти на системата.**
* **DataWise.Api:  
  Този проект представлява API слой, базиран на ASP.NET, който осигурява мащабируемост и интеграция на различни услуги. Той служи като централен комуникационен възел между фронтенда, AI слоя и останалата инфраструктура.**
* **DataWise.Client:  
  Фронтендът е разработен с Angular и предоставя модерен и интуитивен потребителски интерфейс. Чрез него потребителите могат да взаимодействат с AI модула и да получават незабавна обратна връзка за своите решения по задачи, свързани със структури от данни и алгоритми.**
* **DataWise.Core:  
  Този слой осигурява основните услуги и бизнес логика, реализирани на .NET. Той е отговорен за сървърната логика, координацията на процесите и интеграцията между различните компоненти в системата.**
* **DataWise.Data:  
  Слойът за данни се грижи за съхранението, управлението и достъпа до критичните данни на системата. Той предоставя надежден достъп до бази данни и други информационни източници, необходими за функционирането на цялостната платформа.**
* **DataWise.Tests:  
  Този проект съдържа тестови сценарии и скриптове за валидиране на функционалността и качеството на всички компоненти в системата. Чрез автоматизирани тестове се гарантира стабилността и надеждността на решението.**

**Тази архитектура осигурява цялостно решение, в което всеки слой играе ключова роля:**

* **Модулността позволява независима разработка и оптимизация на отделните компоненти.**
* **Мащабируемостта е постигната чрез интеграция на различни технологии, като AI моделът, API услугите и фронтенд интерфейсът работят синхронно.**
* **Лесната поддръжка и възможността за бъдещо разширяване се гарантират чрез ясното разделение на отговорностите между проектните папки.**
* **Функционалности**

**Следната част е направена така заради това, че AI модулът представлява най-сложната и ценна част от проекта, което го и категоризира в „Големият обем от данни“.. Благодарение на добре изградената архитектура и интегрираната екосистема, останалите функционалности са само въпрос на време. Системата е проектирана да бъде изключително мащабируема, което гарантира, че всяка следваща функционалност може лесно да бъде интегрирана и разширена в бъдеще.**

* **Настоящи функционалности:**
  + **Локално изпълнение на модела:  
    В момента моделът може да бъде стартиран локално, което позволява на потребителите да го използват за лични цели. Той обработва входните данни и предоставя автоматична категоризация според предварително обучените категории.**
  + **Уеб интерфейс с категоризация:  
    Потребителите могат да посетят уебсайта, където е интегрирана функционалността за категоризация. Чрез сайта могат да въведат текст, който се обработва от AI модела, и да получат незабавна обратна връзка.**
* **Бъдещи функционалности:**
  + **Разделяне на проекта на три основни категории:**
    1. **Knowledge Nexus:  
       Ще бъде добавена лекционна част, насочена към обучение по алгоритми и структури от данни. Този модул ще помага на потребителите да се подготвят за интервюта или изпити, предоставяйки теоретични и практически примери.**
    2. **Visionary Vault:  
       Този компонент ще визуализира всички данни, алгоритми и математически модели чрез интерактивни симулации и визуализации. Така потребителите ще могат да видят динамиката и връзките между различните концепции в реално време.**
    3. **Solution Beacon (AI Модул-а):  
       Развитието на AI модела ще продължи, за да осигури все по-прецизни и сложни алгоритми за класификация.**

**3.5. Реализация**

**3.5.1 AI Модул – Архитектура на модела**

**Моделът, наречен TextCNN, представлява конволюционна невронна мрежа (CNN), предназначена за класифициране на текстови задачи. Подробното описание на параметрите и функционалностите на модела е следното:**

* **Входни параметри и начална инициализация:**
  + **vocab\_size: Размерът на речника, използван за векторизиране на думите.**
  + **embedding\_dim: Измерение на векторното представяне на всяка дума. В началната стъпка се инициализират случайни вектори, умножени с малък коефициент (0.01), за да се избегнат прекалено големи стойности.**
  + **max\_len: Максимална дължина на входното изречение (брой думи), до която се обработват данните.**
  + **num\_filters: Броят на филтрите, използвани в конволюционните слоеве. Тези филтри извличат характеристики от вградените представяния.**
  + **filter\_sizes: Списък от размери на филтрите (например 2, 3, 4), които определят колко думи се разглеждат едновременно по време на конволюцията.**
  + **num\_classes: Броят на категориите, в които моделът трябва да класифицира входното изречение.**
  + **learning\_rate: Стъпковата скорост, с която се актуализират параметрите при обучение.**
* **Вградени представяния (Embeddings):**
  + **Използва се матрица с размер [vocab\_size×embedding\_dim][vocab\\_size \times embedding\\_dim][vocab\_size×embedding\_dim], инициализирана с малки случайни стойности, която трансформира думите в непрекъснати вектори.**
* **Конволюционни слоеве:**
  + **За всеки размер на филтър от списъка filter\_sizes се създава отделна група конволюционни филтри с размер [fs×embedding\_dim×num\_filters][fs \times embedding\\_dim \times num\\_filters][fs×embedding\_dim×num\_filters]. Използва се He инициализация (через множител 2.0/(fs×embedding\_dim)\sqrt{2.0 / (fs \times embedding\\_dim)}2.0/(fs×embedding\_dim)**
* **Процес на конволюция и максимално обединяване (Pooling):**
  + **При метода forward() входното изречение се вгражда чрез матрицата с embeddings.**
  + **Прилага се конволюция върху "прозорец" (window) от fs думи, като се извличат характеристики чрез филтрите.**
  + **Получените резултати се подават през функцията за активация ReLU, след което се извършва max-pooling по времевата ос, за да се извадят най-силните характеристики за всяка група филтри.**
  + **Резултатите от всички конволюционни слоеве се конкатенират, за да се получи вход за последния, напълно свързан слой (fully connected layer).**
* **Финален слой (Fully Connected):**
  + **Използва се матрично умножение с теглата [fc\_input\_dim×num\_classes][fc\\_input\\_dim \times num\\_classes][fc\_input\_dim×num\_classes] и добавяне на bias, за да се получат логитите, от които чрез softmax функцията се извличат вероятностите за всяка категория.**
* **Обратна разпространение (Backpropagation):**
  + **Изчислява се градиентът на крос-ентропийната загуба спрямо логитите, който се предава през напълно свързания слой.**
  + **Градиентите се разделят за всеки конволюционен слой, като при max-pooling се проследява кой елемент е дал максималната стойност (backpropagation през max pooling).**
  + **Градиентите се пресмятат за конволюционните филтри и embeddings, след което всички параметри се актуализират с използване на learning rate.**
    1. **Програмни приложения**
* **За разработката и поддръжката на проекта са използвани няколко съвременни инструмента:**
  + **Rider:  
    Отлично IDE за разработка на .NET приложения. Осигурява стабилна поддръжка за C# и ASP.NET, улеснявайки управлението на сложната бизнес логика.**
  + **PyCharm:  
    Популярен IDE за разработка с Python, който е използван за разработка и оптимизация на AI модела и Flask endpoint-ите.**
  + **Visual Studio Code:  
    Лек и гъвкав редактор, използван за работа с Angular и TypeScript, както и за обща редакция на кода. Поддържа множество разширения, които помагат при разработката на уеб приложения.**

**3.5.3 Технологични средства**

* **Проектът използва разнообразни технологични средства, които осигуряват цялостната функционалност и мащабируемост:**
  + **ASP.NET, C#, EF Core:  
    Осигуряват стабилна и мащабируема среда за изграждане на API услугите и бизнес логиката. EF Core позволява лесно управление и достъп до базите данни.**
  + **Angular, TypeScript, Node.js:  
    Създават модерен, интерактивен фронтенд, който комуникира чрез RESTful API с Back-End услугите. Node.js се използва за управлението на зависимостите и стартиране на процесите по разработка.**
  + **Flask, Python:  
    Основата за AI модула, който обработва входните данни, извършва сложни математически изчисления и връща резултатите от класификацията.**
  + **HTML, CSS, Bootstrap:  
    Използват се за оформяне и стилизиране на потребителския интерфейс, осигурявайки отзивчив и модерен дизайн.**

**3.6. Описание на приложението**

**За да се стартира приложението, е необходимо да се изпълнят следните стъпки:**

* **DataWise.AI (Python):  
  Отидете в папката DataWise.AI и изпълнете:**
  + **python app.py – за стартиране на Python endpoint-а, който обслужва заявките към AI модела.**
  + **Ако желаете да обучите собствен модел, използвайте:  
    python train.py**
* **DataWise.Api (.NET):  
  Отидете в папката DataWise.Api и стартирайте .NET API проекта. Обикновено това се извършва чрез командата:**
  + **dotnet run  
    (Уверете се, че имате инсталирана подходящата версия на .NET и че проектът е конфигуриран правилно.)**
* **DataWise.Client (Angular):  
  Отидете в папката DataWise.Client и изпълнете:**
  + **npm start  
    Тази команда стартира Angular клиента, който предоставя потребителския интерфейс на приложението.**

**3.7. Заключение**

**DataWise представлява цялостно и иновативно решение за подготовка на кандидати за технически интервюта, предоставяйки модерни AI технологии и образователни ресурси за усвояване на структурите от данни и алгоритмите. AI моделът, базиран на сложни математически осигурява решение под формата на категория алгоритъм/стратегия, която трябва да се ползва за решаване на задачата.**

**Приложението е насочено към кандидати за програмиране – както студенти, така и млади професионалисти – които се подготвят за интервюта във водещите технологични компании (например FANG). DataWise подпомага потребителите да затвърдят своите знания по структури от данни и алгоритми и да се справят успешно с техническите предизвикателства на интервютата.**

**В заключение, DataWise демонстрира висок технически потенциал и практическа стойност за целевата аудитория. Чрез съчетаването на сложни математически алгоритми и интеграцията на различни технологични екосистеми, проектът осигурява мащабируемост, надеждност и гъвкавост – ключови фактори за успешното обучение и професионално развитие на бъдещите кандидати за технически интервюта.**