



УНИВЕРЗИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“ – БИТОЛА

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИСКИ  
ТЕХНОЛОГИИ – БИТОЛА



Студиска програма:  
Информатички науки и компјутерско инженерство

**АДАПТИРАНИ ЈАЗИЧНИ МОДЕЛИ И ВЕКТОРСКИ БАЗИ НА ПОДАТОЦИ ЗА РАЗВОЈ  
НА ИНТЕЛИГЕНТНИ АГЕНТИ СО ПЕРСОНАЛИЗИРАНА ИНТЕРАКЦИЈА**

-Докторски проект-

Кандидат:  
м-р. Анета Трајковска  
бр. на индекс: 20/23/III

Ментор:  
ред. проф. д-р. Костандина Вељановска

Битола, февруари 2026

## СОДРЖИНА

Апстракт .....	3
ОБРАЗЛОЖЕНИЕ НА ТЕМАТА НА ДОКТОРСКИОТ ПРОЕКТ .....	4
1. Вовед .....	5
2. Вештачка интелигенција и Интелигентни агенти, предизвици и проблеми .....	5
3. Методологија и фази на развој .....	6
4. Преглед на архитектура и клучни технологии .....	7
4.1. Дизајн и развој на интелигентен агент и имплементација .....	8
4.2. Големи јазични модели .....	9
4.3 Интеграција со Azure OpenAI моделот .....	9
4.4. Процес на вгнездување на содржина .....	10
4.5. Имплементација на алгоритми за семантичка сличност .....	10
4.6. Процес на фино или прецизно дообучување .....	12
4.7. Retrieval Augmented Generation процес .....	12
4.8. Компаративна анализа помеѓу фино дообучување и употреба на RAG метод .....	13
5. Развој и имплементација на интелигентни агенти со персонализирана интеракција .....	14
5.1. Фаза 1: Агрегација на податоци од различни локации .....	14
5.2. Фаза 2: Подготовка на податоците .....	15
5.3. Фаза 3: Вгнездување на податоците .....	16
5.4. Фаза 4: Запишување на податоците во векторска база .....	16
5.5. Фаза 5: Персонализирана интеракција со интелигентниот агент .....	19
6. Валидација и верификација .....	21
7. Мониторинг и анализа на резултатите .....	21
8. Евалуација на резултатите и предмет на дискусија .....	21
9. Заклучок .....	22
10. Референци .....	22

# АДАПТИРАНИ ЈАЗИЧНИ МОДЕЛИ И ВЕКТОРСКИ БАЗИ НА ПОДАТОЦИ ЗА РАЗВОЈ НА ИНТЕЛИГЕНТНИ АГЕНТИ СО ПЕРСОНАЛИЗИРАНА ИНТЕРАКЦИЈА

Анета Трајковска, Универзитет „Св. Климент Охридски“ – Битола  
Факултет за информатички и комуникациски технологии – Битола  
[aneta.trajkovska@uklo.edu.mk](mailto:aneta.trajkovska@uklo.edu.mk)

Костандина Вељановска, Универзитет „Св. Климент Охридски“ – Битола  
Факултет за информатички и комуникациски технологии – Битола  
<https://orcid.org/0009-0008-6436-2859>  
[kostandina.veljanovska@uklo.edu.mk](mailto:kostandina.veljanovska@uklo.edu.mk)

## Апстракт

Интелигентните агенти претставуваат клучен технички концепт што овозможува ефективна и скалабилна примена на вештачката интелигенција и машинското учење во реални, комплексни системи, а нивната примена претставува фактор од исклучително значење за општествениот развој, со потенцијал суштински да го трансформира начинот на интеракција меѓу човекот и технологијата. Овие системи се карактеризираат со способност за автономно донесување одлуки, анализа на комплексни податоци и адаптација на променливи услови, што резултира со динамични структури способни за континуирано учење и оптимизација на сопствените процеси. Ова истражување се фокусира на развој и имплементација на интелигентен агент за овозможување на персонализирана корисничка интеракција прилагодена на индивидуалните потреби. Техничкиот пристап вклучува генерирање со зголемено пребарување (Retrieval-Augmented Generation), кое овозможува динамично збогатување на содржината преку пребарување на релевантни надворешни податоци, додека адаптацијата на јазичните модели придонесува кон унапредување на процесот на донесување одлуки. Предложениот агент функционира како виртуелен асистент способен за анализа на кориснички барања и извршување на соодветни акции. Потребата за ваков тип на технолошки решенија е особено изразена, бидејќи пристапната и прифатлива комуникација претставува основа за активно учество во дигиталното општество. Дигиталната писменост е од суштинско значење, па со искористување на технолошката експанзија, денес ние имаме можност да изградиме интелигентни системи со користење на адаптирани јазични модели и векторски бази, кои може да бидат користени од целата популација.

**Клучни зборови:** вештачка интелигенција, интелигентни агенти, интеграција, адаптирани јазични модели (анг. fine tuning), принудно учење (reinforcement learning), векторски бази, интеракција во реално време, генерирање со зголемено пребарување (анг. Retrieval Augmented Generation – RAG).

## ОБРАЗЛОЖЕНИЕ НА ТЕМАТА НА ДОКТОРСКИОТ ПРОЕКТ

Тековниот развоен циклус на ВИ означува суштинска трансформација во дизајнот и примената на техничките системи, при што ВИ претставува доминантен фактор за унапредување на комплексни технолошки и организациски процеси. Во тој контекст, се наметнува потребата од развој на интелигентни системи кои не се ограничуваат само на автоматизација на процеси, туку овозможуваат и персонализирана, контекстуално свесна интеракција со корисниците, заснована на анализа на нивното однесување, потреби и тековна состојба.

Ова истражување е насочено кон примена на адаптирани јазични модели и векторски бази како основни технологии за развој на интелигентни агенти способни да поддржат ваков тип на интеракција. Преку прилагодување и интеграција на современи архитектури на големи јазични модели, заедно со техники за вгнездување и векторско претставување на податоци, се воспоставува технолошка основа за системи кои комбинираат разбирање на природен јазик, интелигентно препознавање на контекст и автоматизирано донесување одлуки.

Проектот се темели на холистички и адаптивен концепт, со цел развој на флексибилно решение кое може беспрекорно да се интегрира во различни домени и да придонесе кон унапредување на ефикасноста, комуникацијата и корисничкото искуство. Дополнително, имајќи го предвид динамичниот развој на технологиите во облак, како и зголемените барања за сигурност и скалабилност, во рамките на истражувањето ќе бидат разгледани и механизми за мониторинг, заштита на податоците и етичко функционирање на ВИ системите.

Со тоа, докторскиот проект ја истражува актуелната потреба од интелигентни агенти кои можат да функционираат во динамични, податочно-интензивни средини и да овозможат континуирана, персонализирана и контекстуално свесна интеракција помеѓу човекот и технологијата.

## 1. Вовед

Развојот на интелегентни агенти станува се понапреден и значаен за обезбедување на персонализирана и ефикасна интеракција со корисниците. Интелегентниот агент претставува комбинација од хардвер и софтвер која користи ВИ за да автономно извршува задачи, донесува одлуки или комуницира со луѓето и системите [1],[19]. ВИ се интегрира во различни области здравство, образование, финансиски сектор, земјоделство итн. со цел воведување на нови иновативни решенија. Согласно со зголемената потреба и брзиот напредок на јазичните модели и технологиите за обработка на природен јазик, се отвора можност за нивна адаптација според специфичните потреби на поединци, што овозможува интелегентна и прилагодена комуникација [2], [3]. Адаптивен јазичен модел е систем кој учи од интеракции за да разбира и прилагодува јазик според корисникот и контекстот. Примената на адаптирани јазични модели, во комбинација со векторски бази за складирање и пребарување на информации, овозможува интелегентните агенти да ја разбираат и интерпретираат информацијата на начин кој е во согласност со индивидуалните преференции и контекст на корисникот. Овој пристап не само што ја подобрува точноста и релевантноста на одговорите, туку и го зголемува нивото на доверба и задоволство кај корисниците, овозможувајќи вистински персонализирана интеракција.

## 2. Вештачка интелигенција и интелегентни агенти, предизвици и проблеми

ВИ и интелегентните агенти претставуваат фундаментален сегмент од современите информатички и интелегентни системи, со сè поголемо влијание врз дигиталната трансформација на различни општествени и индустриски домени. Нивната способност за автономно перцепирање, анализирање и интерпретација на податоци, како и за донесување одлуки во динамични средини, создава значителен потенцијал за унапредување на ефикасноста, квалитетот на услугите и интеракцијата помеѓу човекот и технологијата [18]. Сепак, и покрај забрзаниот развој и широка примена, овие технологии се соочуваат со низа суштински предизвици и ограничувања кои бараат систематско научно разгледување. Еден од централните технолошки предизвици е обработката на големи, хетерогени и често неструктурирани податочни множества, кои претставуваат основа за функционирање на интелегентните агенти [4].

Како потенцијален проблем се појавува адаптивноста на интелегентните агенти во контекст на долгорочна интеракција со корисниците и променливи средини. Иако адаптивните механизми овозможуваат прилагодување врз основа на претходни интеракции, постои ограничување во способноста на системите да воспостават стабилна рамнотежа помеѓу персонализацијата и генерализацијата на однесувањето [14]. Ова може да резултира со појава на пристрасност, деградација на перформансите или непредвидливо однесување на агентите.

Безбедносните аспекти и заштитата на податоците претставуваат уште еден критичен сегмент во развојот на интелегентни агенти. Функционирањето на ваквите системи често подразбира обработка на чувствителни и лични информации, што наметнува строги барања за имплементација на механизми за заштита на приватноста, контрола на пристап и усогласеност со постојните регулаторни и етички рамки [3]. Недоволната сигурност и можноста за злоупотреба на ВИ системите претставуваат сериозен ризик за довербата и нивната долгорочна прифатеност.

Покрај техничките и безбедносните аспекти, значајни предизвици се јавуваат и во етичкиот и социјалниот контекст. Автономијата на интелегентните агенти, одговорноста за донесените одлуки и нивното потенцијално влијание врз работните

процеси и општествените структури отвораат комплексни прашања поврзани со коректно однесување, одговорна употреба и објаснивост на ВИ системите. Овие прашања се од особена важност при имплементација на интелегентни агенти во домени со висок степен на критичност.

Интелегентните агенти сè почесто се интегрираат со цел подобрување на ефикасноста, автоматизација на процесите и поддршка во донесувањето одлуки. Сепак, нивната примена не е без значајни предизвици. Практичната имплементација на овие системи открива бројни ограничувања кои ја потенцираат комплексноста на нивната интеракција со техничките, организациските и човечките аспекти на работењето. Недостатоците не произлегуваат единствено од технолошките ограничувања, туку и од начинот на прифаќање, употреба и управување на системите во организациски контекст. Поради тоа, критичката анализа на клучните слабости и потенцијалните ризици поврзани со нивната примена е неопходна за овозможување на поефикасна интеграција, одржливост и долгорочна доверба во вештачката интелигенција како составен дел од деловните процеси. Во продолжение се претставени најчестите пропусти и недостатоци забележани при имплементација на интелегентни агенти:

- Недоволна интеграција со постоечките системи
- Отпор од вработените и страв од автоматизација
- Пристрасност во одлуките
- Недостиг од транспарентност и одговорност
- Проблеми со безбедност и приватност на податоци
- Високи трошоци за одржување и ажурирање
- Ограничена флексибилност и адаптација

Иако ВИ и интелегентните агенти претставуваат моќен инструмент за технолошки напредок, нивниот развој и практична примена се проследени со повеќеслојни предизвици кои бараат интегриран, системски и научно заснован пристап [15]. Надминувањето на овие проблеми е клучно за создавање на доверливи, безбедни и одржливи интелегентни системи, способни да одговорат на комплексните барања на современото дигитално општество.

### **3. Методологија и фази на развој**

Во рамките на овој труд се применува сеопфатен и мултидимензионален методолошки пристап со цел детален преглед на можностите за ефективна имплементација и интеграција на современите ВИ алатки. За потребите на овој истражувачки труд, беше спроведен системски и обемен преглед на литературата со цел да се идентификува и критички оцени актуелноста на алатките за ВИ и нивната применливост во развој на интелегентни агенти. Пребарувањето опфати реномирани академски списанија, трудови од меѓународни конференции, реномирани онлајн бази на податоци и релевантна сива литература, со посебен акцент на публикации објавени во последните пет години, со цел да се осигура интеграција на најновите достигнувања, трендови и иновации во областа. По завршувањето на прегледот на литературата, беше применет систематски пристап за

селекција на алатки кои ќе се применуваат во текот на истражувањето. Критериумите на вклучување се однесуваат според популарноста на алатките, иновативните истражувања како и лавиринтот на можности за имплементација со цел да се добие скалабилно решение кое докажано влијае врз резултатите.



Слика 1: Фази на работен тек

Методологијата на истражувањето вклучува неколку фази, прикажано во Слика 1.

- Анализа и планирање - во оваа почетна фаза се дефинира проблемот, целите на истражувањето и очекуваните резултати.
- Дизајнирање на архитектурата и работниот тек - се креира архитектонски модел на системот, каде се дефинираат компонентите, нивната меѓусебна поврзаност и начинот на комуникација помеѓу модулите.
- Развој и имплементација - интеграција на адаптиран јазичен модел, креирање и поврзување со векторска база, како и дефинирање на алгоритми за персонализирана интеракција.
- Валидација и верификација – спроведување на тестови за точност, перформанси и стабилност на системот, како и оценување на квалитетот на интеракцијата помеѓу агентот и корисникот преку различни сценарија на комуникација.
- Мониторинг и анализа на резултатите – примена на систем за континуирано следење и мерење на перформансите. Исто така, се анализираат собраните податоци за подобрување на моделот и прилагодување на системот кон реални услови.
- Заклучоци и препораки – изведување на конечните заклучоци од истражувањето и предлагање насоки за понатамошен развој и примена во различни области.

#### 4. Преглед на архитектура и клучни технологии

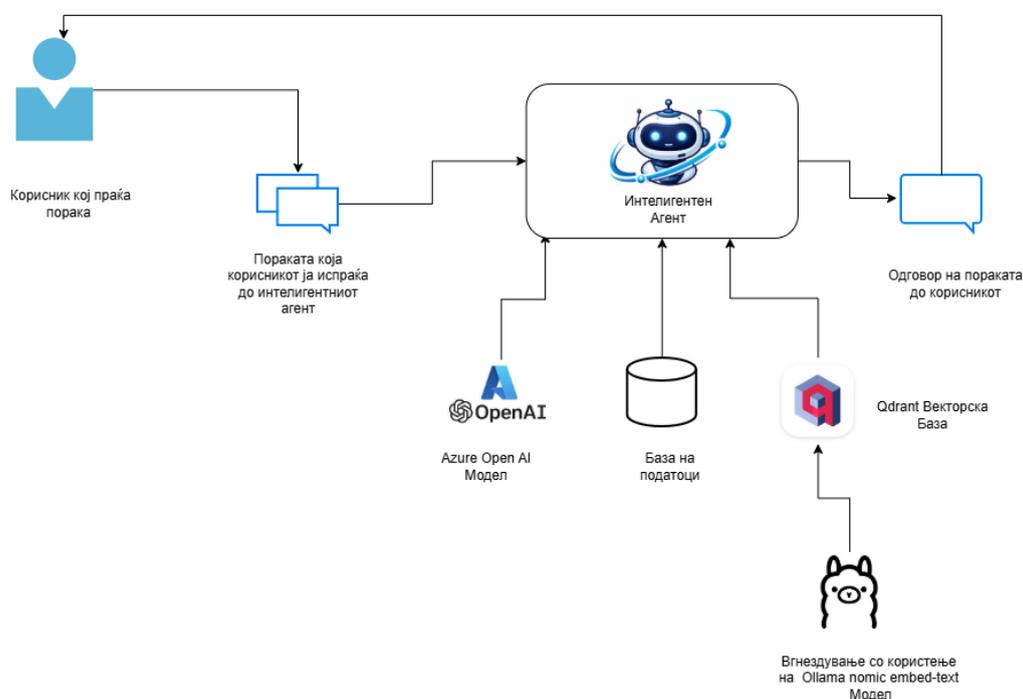
Прегледот на архитектурата обезбедува целосен концептуален увид во системот преку идентификација на основните елементи, нивните функции и начинот на интеракција. Во овој дел се претставува логичката структура и протокот на информации помеѓу слоевите, овозможувајќи проценка на ефикасноста, флексибилноста и сигурноста на решението. Прегледот служи како водилка за донесување одлуки, анализа на ризици и стратески планирање на идни надградби или интеграции, Слика 2.



Слика 2: Преглед на планираните имплементации

#### 4.1. Дизајн и развој на интелигентен агент и имплементација

Развојот на интелигентниот агент ќе се реализира низ повеќе фази, при што ќе се врши континуиран пресек и евалуација помеѓу поединечните фази, следејќи релевантни перформансни индикатори, прикажано на Слика 3. Архитектурата на системот ќе се надградува итеративно сè до фазата на имплементација на целосно функционален и поддржан систем со сопствена инфраструктура. Оваа инфраструктура ќе вклучува примена на технологии во облак како AWS (анг. Amazon Web Services) и ECS (анг. Elastic Container Service), како и имплементација на концепти на дистрибуирани системи преку Apache Kafka за обработка на настани и асинхрона комуникација [16],[20].



Слика 3: Преглед на предложената архитектура при фаза1 на интелигентниот агент

## 4.2. Големи јазични модели

Големите јазични модели (анг. Large Language Models - LLM) претставуваат напредна форма на ВИ способна да обработува, разбира и генерира природен јазик на ниво блиско до човечкото [5]. Тие се базираат на длабоки невронски мрежи кои се обучуваат на огромни количини текстуални податоци, со што стекнуваат способност да предвидуваат зборови, формулираат одговори и креираат смислени текстови. Овие модели се користат во различни области - од автоматизирана поддршка на корисници и преведување, до генерирање на содржина и интелигентни агенти кои комуницираат природно и интуитивно со луѓето.

Обучувањето на големите јазични модели се одвива во две главни фази:

- Претходно обучување (анг. Pre-training) – моделот учи општи правила и структури од големи и разновидни текстуални податоци.
- Фино или така наречено прецизно дообучување: моделот се дообучува на специфични задачи или домени (медицина, финансиски сектор, образование) за да стане покорисен од одреден област.

Меѓу најзначајните современи големи јазични модели се вбројуваат GPT моделите развиени од OpenAI, кои покажуваат висока генерализациска способност и адаптабилност кон различни домен-специфични задачи, Claude од Anthropic, кој е дизајниран со засилен фокус на безбедност и контекстуална конзистентност, како и Gemini од Google DeepMind, кој интегрира мултимодални способности за обработка на хетерогени податоци, [13],[25].

Сеопфатната примена на моделите придонесува кон трансформација на современите интелигентни системи, при што тие сè почесто се интегрираат во архитектури за автономни агенти, системи за поддршка на одлучување низ платформи од различни области. Истовремено, тековните истражувања се насочени кон прашања поврзани со интероперабилност, ефикасност, етички импликации и контрола на однесувањето на овие модели во реални апликативни сценарија.

## 4.3 Интеграција со Azure OpenAI моделот

Интеграцијата со Azure OpenAI моделите најчесто се реализира преку интерфејс за програмирање на апликации (анг. Application Programming Interface - API), кој претставува стандарден и скалабилен механизам за воспоставување комуникација помеѓу локалните апликативни компоненти и моделите распоредени во облак инфраструктура. Овој пристап овозможува јасно дефинирана интеракција преку структуриран формат на барања и одговори, со што се обезбедува контролиран и репродуктивен процес на користење на напредните јазични модели [12].

Преку API повици, системот може да испраќа текстуални или структурирани влезни податоци кон моделот, по што се добиваат генерирани одговори, аналитички интерпретации или препораки, прилагодени на конкретниот апликативен контекст. Дополнително, ваквиот модел на интеграција овозможува централизирано управување со безбедносните политики, автентикација, логирање и мониторинг на повиците, што е особено значајно во системи со зголемени барања за доверливост, скалабилност и перформанси.

Со примена на API-базирана интеграција, Azure OpenAI моделите можат ефективно да

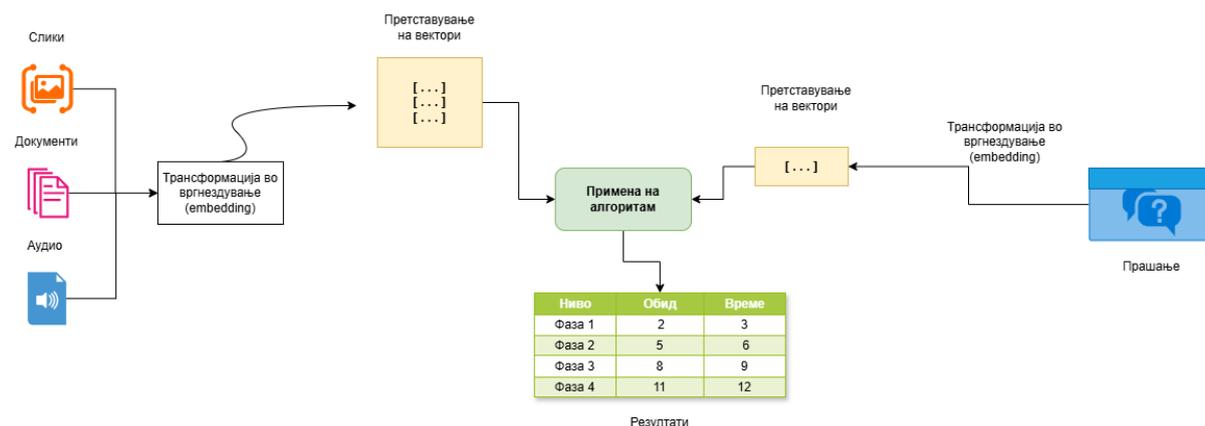
се инкорпорираат во комплексни софтверски архитектури, вклучувајќи микросервисни и дистрибуирани системи, при што се обезбедува флексибилност во развојот и можност за континуирана еволуција на интелегентните функционалности на решението. За интеграцијата со Azure OpenAI моделот се користи, Azure Foundry Service каде што е применет Ollama omic embed-text моделот, [26]. Овозможувањето на конекција со моделот е врз база на генериран API клуч кој се користи за автентикација.

#### 4.4. Процес на вгнездување на содржина

Процесот на вгнездување на содржината претставува клучна фаза во обработката и репрезентацијата на податоци кај системи базирани на ВИ, особено кај архитектури кои се потпираат на векторски претстави (анг. embeddings). Овој процес опфаќа трансформација на текстуална или мултимедијална содржина, вклучувајќи текст, слики и аудио, во нумерички вектори со фиксна димензионалност, кои служат како математичка апстракција на семантичкото значење и контекстуалните релации на податоците, Слика 4.

Генерираните векторски претставувања овозможуваат пресметување на сличност помеѓу различни елементи, кластеризација на содржина и ефикасно пребарување во големи збирки на податоци. Како резултат на тоа, вгнездувањето претставува основа за имплементација на напредни функционалности како семантичко пребарување, системи за препораки и интеграција со големи јазични модели, [6]. Прецизноста и квалитетот на вгнездувањето директно влијаат врз перформансите и релевантноста на резултатите во ваквите интелегентни системи, што го прави овој процес критичен за целокупната ефикасност на решението [11].

Векторските бази претставуваат специјализирани системи за чување и пребарување на податоци кои се претставени како вектори - нумерички низи кои ја кодираат семантичката или контекстуалната информација за одредена содржина, [27].



Слика 4: Процес на вгнездување на содржината

#### 4.5. Имплементација на алгоритми за семантичка сличност

Во современите системи базирани на ВИ, способноста за разбирање на значењето на текстот, наместо само буквите или зборовите, станува клучна [28]. Оваа способност се постигнува преку алгоритми за семантичка сличност, кои овозможуваат модели и апликации да идентификуваат колку различни текстови, документи или информации се значајно блиски [8],[9]. Примената на овие алгоритми е особено важна во системи за

пребарување, препорачувачки системи, интелигентни агенти и анализи на големи текстуални бази.

Алгоритмите за семантичка сличност најчесто функционираат преку векторска претстава на содржината. Текстот се претвора во вектори преку модели за обработка на природен јазик, како што се големи јазични модели или посебни модели за векторски претстави, [17].

Сличноста меѓу два вектори се мери со математички метрики, како што се:

- Hierarchical Navigable Small World (HNSW) - претставува алгоритам за апроксимативно пребарување на најблиските соседи (Approximate Nearest Neighbor Search) во векторски простори со висока димензионалност. HNSW комбинира две клучни концепти: хиерархиска структура на слоеви и „навигациски мал свет“ (анг. navigable small world) графови [33]. Структурата се состои од повеќе слоеви на графови, каде што секој јазол претставува еден податочен вектор, а врските (анг. edges) поврзуваат слични вектори. Алгоритмот гради структура со повеќе слоеви. Највисокиот слој содржи најмалку јазли и овозможува брзо движење низ просторот. Секој понизок слој содржи повеќе јазли со поголема густина на поврзување. При пребарување, алгоритмот почнува од највисокиот слој и постепено се пребарува надолу, секогаш движејќи се кон локацијата најблиску до барањето. Секој јазол е поврзан со фиксен број на најблиски соседи (параметар М), што создава "мал свет" топологија – може да се стигне од било кој јазол до друг за мал број на чекори [34]. Алгоритмот е многу брз при пребарување (логаритамска сложеност), прилично прецизен, и добро скалира со големи бази на податоци. Поддржува инкрементално додавање на нови вектори без потреба од целосна реконструкција. HNSW се користи во семантичко пребарување, системи за препораки, пребарување на слики, и векторски бази на податоци [35].
- Косинусна Сличност (анг. Cosine Similarity) - мери колку два вектори имаат слична насока, Слика 5.

```
Scripts > similarity_search_algo_example.py > ...
1 import numpy as np
2
3 def cosine_similarity(x, y):
4     return np.dot(x, y) / (np.linalg.norm(x) * np.linalg.norm(y))
5
6 def similarity_search(query, embeddings):
7     scores = []
8
9     for idx, vector in enumerate(embeddings):
10        score = cosine_similarity(query, vector)
11        scores.append((idx, score))
12
13        # сортирај по сличност (опаѓачки)
14        scores.sort(key=lambda x: x[1], reverse=True)
15
16        return scores
17
18 # embedding димензија = 3 (само за пример)
19 embeddings = [
20     np.array([1.0, 0.0, 0.0]), # doc 0
21     np.array([0.9, 0.1, 0.0]), # doc 1
22     np.array([0.0, 1.0, 0.0]) # doc 2
23 ]
24
25 query = np.array([1.0, 0.0, 0.0])
26
27 results = similarity_search(query, embeddings)
28
29 for idx, score in results:
30     print(f"Doc {idx} → similarity = {score:.3f}")
31
```

Слика 5: Алгоритам за косинусна сличност

- Евклидово растојание (анг. Euclidean distance) – мери физичката дистанца помеѓу векторите во повеќедимензионален простор.

$$D = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2}$$

- Скаларен производ – се користи секаде каде е потребна мерка на корелација помеѓу векторите.

Преку овие метрики, системите можат да одредат кои документи или информации се најблиски по значење на дадена содржина. Применетата логика во системот се однесува на начинот на кој интелегентниот агентот донесува одлуки, обработува информации и реагира на кориснички барања. Таа претставува основа за рационално однесување на системот, овозможувајќи му да ја анализира влезната содржина, да препознае обрасци и да изврши соодветни активности врз основа на дефинирани правила, модели на учење и контекстуални податоци [22].

#### 4.6. Процес на фино или прецизно дообучување

Финото или прецизното дообучување (анг. fine tuning) е напредна техника во областа на машинското учење и ВИ која овозможува адаптација на веќе обучен модел кон специфични задачи или контексти, без потреба од целосно повторно обучување. Овој процес е особено значаен при работа со големи јазични модели, бидејќи овозможува искористување на постојното општо знаење на моделот, додека се додаваат нови, специјализирани способности, [29].

Моделот ја задржува својата општа јазична компетенција, но дополнително ги „учи“ специфичните термини, контексти и правила што важат во таргетираниот домен. Целта е да се постигне поголема точност, релевантност и контекстуална усогласеност при генерирање на одговори или анализи.

Прецизното дообучување вклучува и фино прилагодување на параметрите и тежините на моделот, што овозможува оптимизација на неговото однесување врз основа на повратни информации или специфични примери. На тој начин, моделот станува посспецифичен, поефикасен и покорисен за конкретни апликации, без потреба од целосно повторно обучување од нула [10]. Овој процес претставува баланс меѓу генералното знаење стекнато во основната фаза и специјализираното знаење потребно за одредена употреба, со што се создаваат модели кои се истовремено интелегентни и контекстуално чувствителни, [30].

#### 4.7. Retrieval Augmented Generation процес

Процесот на генерирање со зголемување на пребарувањето (анг. Retrieval Augmented Generation - RAG) претставува метод дизајниран за подобрување на перформансите на традиционалните големи јазични модели преку интеграција со надворешни системи за пребарување информации. Во RAG поставувањето, системот за пребарување како што е пребарувач или векторска база на податоци, собира релевантни информации од голем корпус на податоци, [7]. Овие надворешни извори на знаење се користат за водство на процесот на генерирање на јазичен модел, што резултира со поточни и контекстуално релевантни одговори [38]. Главната предност на RAG е способноста на моделот да пристапи до ажурирани и специјализирани информации за доменот, кои можеби не биле достапни за време на обуката, комбинирајќи пребарување со генерирање за

поинформативни и прецизни резултати, [31].

RAG процесот се состои од три основни компоненти: пребарување, зголемување и генерирање, при што секоја игра критична улога во обезбедувањето точни и контекстуално свесни излезни податоци, Слика 6.

**Пребарување:** во оваа фаза, системот истражува надворешни извори на знаење за да собере релевантни информации. Ова може да вклучува документи, статии или веб-ресурси. Со користење на техники како споредување на клучни зборови или методи базирани на вгнездување (анг. embeddings), системот идентификува и презема податоци кои се блиски до корисничкото барање. Преземените информации им овозможуваат на моделите да го надминат своето вродено знаење, пристапувајќи до информации во реално време или специјализирани податоци за кои не биле обучени.

**Зголемување:** откако ќе се обезбедат релевантните податоци, започнува фазата на зголемување (анг. augmentation). Преземените информации се интегрираат како дополнителен контекст за моделот, помагајќи му да генерира поинформативни и поточни одговори. Овој чекор гарантира дека одговорите не се базираат исклучиво на вроденото знаење на моделот, туку се збогатени со надворешни податоци, што резултира со одговори кои се и пообемни и релевантни.

**Генерирање:** во последната фаза, јазичниот модел ја обработува зголемената информација и создава кохерентен и контекстуално свесен одговор. Анализирајќи го новиот надворешен контекст со претходното знаење, моделот произведува одговори кои се граматички точни и високо релевантни за корисничкото барање. Овој чекор овозможува RAG системите да создаваат прецизни, поинформативни и контекстуално богати одговори.



Слика 6: Основни компоненти на RAG

#### 4.8. Компаративна анализа помеѓу фино дообучување и употреба на RAG метод

Со цел систематско согледување на клучните карактеристики и разлики помеѓу разгледуваните пристапи, во продолжение е прикажана компаративна анализа, [32]. Анализата ги опфаќа релевантните аспекти од архитектонска, функционална и евалуациска перспектива, овозможувајќи јасен и структуриран увид во нивните предности и ограничувања, Табела број 1.

Споредба врз основа на:	RAG	Фино дообучување
<b>Природа на извршување на задачи</b>	Моделот користи надворешни извори на знаење за да генерира одговори во реално време; комбинира пребарување со генерирање.	Моделот се дообучува со специфични податоци за задача или домен за да ги вгради новите знаења директно во тежините на моделот.
<b>Прилагодливост</b>	Висока, може лесно да се менуваат изворите на знаење без повторно тренирање.	Потребно е повторно тренирање на моделот за нови домени или задачи.
<b>Потреба за податоци</b>	Користи постоечки документи и ресурси, секогаш бара база на знаење за ефикасно пребарување.	Побарува големи количини квалитетни податоци за ефективно дообучување.
<b>Ажурирање на знаење</b>	Може брзо да пристапи до ажурирани информации преку надворешни извори.	Знаењето е статично по дообучувањето. За нови информации потребно е повторно тренирање.
<b>Комплексност</b>	Вклучува заедничка обука и за генеративните и за компонентите за пребарување, додавајќи сложеност на процесот на обука.	Поедноставни процедури за обука, особено кога се користат претходно обучени модели со лесно достапни збирки на податоци специфични за задачата.

Табела број 1: Споредба помеѓу finely дообучување и RAG

## 5. Развој и имплементација на интелигентни агенти со персонализирана интеракција

Подготвените податоци кои ги користевме за оваа фаза на експериментот се податоци кои се автоматски генерирани преку API и користење на интегрирано решение во Kafka, [23]. Податоците се состојат од информации за автомобили во Америка од типот на марка, бренд, боја, цена, година на производство како и дополнителни информации во однос на локациите каде се наоѓаат автосалоните. Истите се централизирани и запишани во база на податоци во Postgres, моменталната големина на базата изнесува 96000 податоци.

Програмскиот јазик за имплементација на решението е Python и користење соодветни библиотеки.

### 5.1. Фаза 1: Агрегација на податоци од различни локации

Првата фаза од предложениот системски модел опфаќа процес на вчитување, собирање и примарна агрегација на податоци, кои се наоѓаат на дефинирани извори и ја сочинуваат основната база за анализа на автомобилски огласи. Податоците се генерираат и прибираат од релевантни веб-платформи за продажба на автомобили, при што се

применуваат механизми за обработка на настани во реално време базирани на Apache Kafka, [24]. Овој пристап овозможува континуирано следење на динамичките промени во податоците, нивна навремена обработка и асинхроно пренесување преку стриминг архитектура. Суровите, неструктурирани податоци се запишуваат во наменска база на податоци, со што се обезбедува стабилна основа за понатамошни фази на прочистување, трансформација и аналитичка обработка. Ваквиот дизајн овозможува висока скалабилност, отпорност на грешки и поддршка за обработка на големи количини податоци во реално време, Слика 7.

	AZ name	AZ state	AZ city	AZ manufacturer	AZ model	123 year	123 price	123 cylinders	AZ cate
1	Koelpin, Hudson and Ledner	Arizona	Mesa	MERCEDES-BENZ	E 350	2,014	48,610	6	Sedan
2	Denesik and Sons	Connecticut	New Haven	HYUNDAI	Sonata	2,016	7,809	4	Sedan
3	Becker Inc	Nebraska	Lincoln	FORD	Transit	1,997	13,485	4	Goods v
4	Sawayn, Corwin and Legros	Texas	Houston	TOYOTA	RAV 4	2,015	28,225	4	Jeep
5	Smitham-Roberts	Nebraska	Omaha	TOYOTA	Prius	2,015	19,130	4	Hatchba
6	Kunde-Dare	Massachusetts	Boston	SSANGYONG	REXTON	2,011	31,659	4	Jeep
7	Waelchi-Boyer	Colorado	Colorado Springs	CHEVROLET	Volt	2,012	22,580	4	Sedan
8	Heaney-Goodwin	Michigan	Lansing	MERCEDES-BENZ	E 500 AMG	2,003	11,917	12	Sedan
9	Koelpin, Hudson and Ledner	Arizona	Mesa	TOYOTA	Prius C Hybrid	2,014	15,994	4	Hatchba
10	Heaney-Goodwin	Michigan	Lansing	VOLKSWAGEN	Jetta	2,015	17,876	4	Sedan
11	Mueller, Fay and Kunde	Illinois	Peoria	MITSUBISHI	CoitPlus Plus	2,006	9,095	4	Hatchba
12	Stanton Group	Arizona	Phoenix	NISSAN	Juke	2,012	20,385	4	Jeep
13	Goyette and Sons	Kansas	Shawnee Mission	RENAULT	Duster	2,017	37,006	4	Jeep
14	Jacobs, Lakin and Crooks	District of Colum	Washington	MERCEDES-BENZ	C 280	1,997	5,331	6	Sedan
15	Kunze-Ondricka	Florida	Miami	HYUNDAI	Sonata	2,015	392	4	Sedan
16	Adams, Dibbert and Hane	Kentucky	Lexington	OPEL	Corsa	1,999	5,018	4	Coupe
17	Skiles LLC	Texas	Houston	HYUNDAI	Elantra	2,012	14,166	4	Sedan
18	Simonis-Will	Texas	Conroe	MERCEDES-BENZ	E 300	2,017	2,038	4	Sedan
19	Jacobi-Lesch	Texas	Houston	SSANGYONG	REXTON	2,011	29,094	4	Jeep
20	Goyette and Sons	Kansas	Shawnee Mission	VOLKSWAGEN	CC	2,013	18,189	4	Sedan
21	Tromp-Beier	Kansas	Topeka	LEXUS	GX 470	2,013	54,568	8	Jeep

Слика 7: Претставување на податоците во база на податоци

## 5.2. Фаза 2: Подготовка на податоците

Втората фаза е фокусирана на систематска обработка, прочистување и обогатување на прибраните податоци со цел зголемување на нивниот квалитет, конзистентност и аналитичка вредност. Во оваа фаза се применуваат техники за отстранување на дупликати, корекција на недостасувачки и неконзистентни вредности, како и нормализација на податоците добиени од хетерогени извори, Слика 8. Дополнително, податоците се обогатуваат со изведени атрибути и релевантни контекстуални информации, што овозможува подлабока семантичка интерпретација и подобрена подготвеност за понатамошна аналитичка обработка. Ваквиот пристап обезбедува сигурна и структурирана основа за точни анализи, предиктивни модели и интелигентни системи за поддршка на одлучување.

```

9 # Функција за генерирање на опис
10 def generate_description(row):
11     availability = (
12         f"{int(row['quantity'])} units available" if int(row['quantity']) > 0 else "Currently out of stock"
13     )
14     serviced = "recently serviced" if str(row["isserviced"]).upper() == "TRUE" else "not recently serviced"
15     return (
16         f"{row['name']} dealership in {row['city']}, {row['state']} offers the {row['manufacturer']} {row['model']} ({int(row['year'])}).
17         f"This {row['category']} comes in {row['color']} with a {row['engine_vol']}L {row['fuel']} engine, "
18         f"{row['gear']} transmission, {row['doors']} doors, and {row['airbags']} airbags. "
19         f"{availability}. The vehicle has {row['mileage']} km mileage and is {serviced}. Price: ${int(row['price'])}."
20     )
21
22 # Генерирање на опис
23 descriptions = df.apply(generate_description, axis=1)
24
25 # Save to text file
26 with open("full_dealership_descriptions.txt", "w", encoding="utf-8") as f:
27     for desc in descriptions:
28         f.write(desc + "\n")
29

```

Слика 8: Функција за генерирање на опис на полињата

### 5.3. Фаза 3: Вгнездување на податоците

Процесот на вгнездување на податоците е следната фаза каде што обработените и обогатени информации се трансформираат во векторска репрезентација со фиксна димензионалност, погодна за семантичко пребарување и интелигентна обработка. Во оваа фаза се врши структурирање и креирање на контекстуални единици, при што содржината се организира во логички сегменти што ја задржуваат семантичката кохерентност и значењето на податоците, Слика 9. Дополнително, се дефинира и управува со метаподаточна структура која вклучува клучни атрибути, извор на податоци, временски ознаки и релевантни доменски карактеристики.

```

30
31 # Ембединг на текстот со помош на Ollama nomic-embed-text
32 def get_embedding(text):
33     try:
34         response = requests.post(OLLAMA_URL, json={
35             "model": MODEL_NAME,
36             "prompt": text
37         })
38         response.raise_for_status()
39         embedding = response.json().get("embedding")
40         if embedding and len(embedding) == VECTOR_SIZE:
41             return embedding
42         else:
43             logging.warning(f"Embedding size mismatch or missing: got {len(embedding)} if embedding else 'None'")
44             return None
45     except Exception as e:
46         logging.error(f"Embedding failed: {e}")
47         return None
48

```

Слика 9: Вгнездување на текст

### 5.4. Фаза 4: Запишување на податоците во векторска база

По извршениот процес на вгнездување, добиените векторски репрезентации, заедно со контекстуалните единици и метаподатоците, се запишуваат во векторска база на податоци базирана на Qdrant, слика 10. Оваа фаза овозможува ефикасна агрегација и индексирање на високо-димензионални вектори, при што се поддржуваат операции за брзо семантичко пребарување и мерење на сличност, како што се косинусна сличност и евклидово растојание.

```

49 # Форматирање на запис за Qdrant
50 def format_for_qdrant(entry):
51     description = entry.get("full_description")
52     if not description:
53         logging.warning("Missing 'full_description'. Skipping entry.")
54         return None
55
56     vector = get_embedding(description)
57     if vector is None:
58         logging.warning("Invalid embedding. Skipping entry.")
59         return None
60
61     metadata = {k: v for k, v in entry.items() if k != "full_description"}
62
63     return PointStruct(
64         id=str(uuid.uuid4()),
65         vector=vector,
66         payload={
67             "full_description": description,
68             "metadata": metadata
69         }
70     )

```

Слика 10: Форматирање на запис во Qdrant

Qdrant обезбедува оптимизирани механизми за управување со векторски индекси, скалабилност и ниска латентност при пребарување, што е од клучно значење за системи кои функционираат во реално време, слика 11. Ваквата организација на податоците претставува основа за ефикасна имплементација на напредни RAG архитектури и интелегентни агенти, овозможувајќи прецизно извлекување на релевантни информации и контекстуално информирано генерирање на одговори.

```

72 # Поставување на записи во групи/батчови
73 def upload_in_batches(entries):
74     formatted = [format_for_qdrant(e) for e in tqdm(entries, desc="Embedding entries")]
75     valid_points = [p for p in formatted if p is not None]
76     faulty_points = [e for i, e in enumerate(entries) if formatted[i] is None]
77
78     if faulty_points:
79         with open(FAULTY_FILE, "w", encoding="utf-8") as f:
80             json.dump(faulty_points, f, indent=2)
81             logging.info(f"Saved {len(faulty_points)} faulty points to {FAULTY_FILE}")
82
83     total = len(valid_points)
84     logging.info(f"Започнување на поставување на {total} валидни записи во групи од по {BATCH_SIZE}")
85
86     for i in tqdm(range(0, total, BATCH_SIZE), desc="Uploading to Qdrant"):
87         batch = valid_points[i:i + BATCH_SIZE]
88         success = False
89         for attempt in range(1, MAX_RETRIES + 1):
90             try:
91                 client.upsert(collection_name=COLLECTION_NAME, points=batch)
92                 logging.info(f"Batch {i // BATCH_SIZE + 1} uploaded successfully.")
93                 success = True
94                 break
95             except Exception as e:
96                 logging.error(f"Batch {i // BATCH_SIZE + 1} failed on attempt {attempt}: {e}")

```

Слика 11: Поставување на записи во групи

Приказот на објект во корисничкиот интерфејс на Qdrant овозможува визуелна и структурирана репрезентација на векторизираниите податоци, при што секој објект е прикажан заедно со неговата содржина, векторот, како и релевантните метаподатоци. Интерфејсот обезбедува увид во клучните атрибути, изворот на податоци и контекстуалната поврзаност, што овозможува ефикасна верификација, анализа и управување со складираните вектори, слика 12.

Point 0006289b-dfe1-442a-8c1b-0cc65637c217

Payload:

**full\_description** Becker Inc dealership in Lincoln, Nebraska offers the TOYOTA Camry 2014 at the price of 26657. 15 units available at this location.

**metadata**

```
{ 7 Items
  "dealership_name": "Becker Inc"
  "location": "Lincoln, Nebraska"
  "car_make": "TOYOTA"
  "car_model": "Camry"
  "year": 2014
  "price_usd": 26657
  "availability": 15
}
```

Слика 12. Приказ на објект во интерфејс на Qdrant

Семантичката сличност помеѓу корисничкото прашање и постојната содржина се пресметува со примена на метрики како косинусна сличност и Евклидово растојание, овозможувајќи ефикасно и прецизно приближно пребарување во високо-димензионален простор. Овој алгоритамски пристап обезбедува скалабилност, ниска латентност и контекстуално релевантни резултати, што го прави Qdrant соодветна основа за системи базирани на RAG архитектура, Слика 13.

```
# Function to search for similarity in Qdrant
def search_similarity(new_embedding, score_threshold=0.88):
    url = f"{QDRANT_API_URL}/collections/{EMBEDDINGS_COLLECTION_NAME}/points/search"
    payload = {
        "vector": new_embedding,
        "score_threshold": score_threshold,
        "limit": 1
    }
    headers = {
        "api-key": f"{QDRANT_API_KEY}"
    }
    response = requests.post(url, json=payload, headers=headers)
    if response.status_code == 200:
        return response.json()["result"]
    else:
        app.logger.error("Failed to search for similarity: %s", response.status_code)
        return []

@app.route('/api/check_and_insert', methods=['POST'])
def check_and_insert():
    data = request.get_json()
    app.logger.info("Received request for the path: %s", request.path)
    input_text = data.get('input')
    if not input_text:
        app.logger.error("Input text is missing")
        return jsonify({'message': 'Input text is missing', 'status_code': 400}), 400

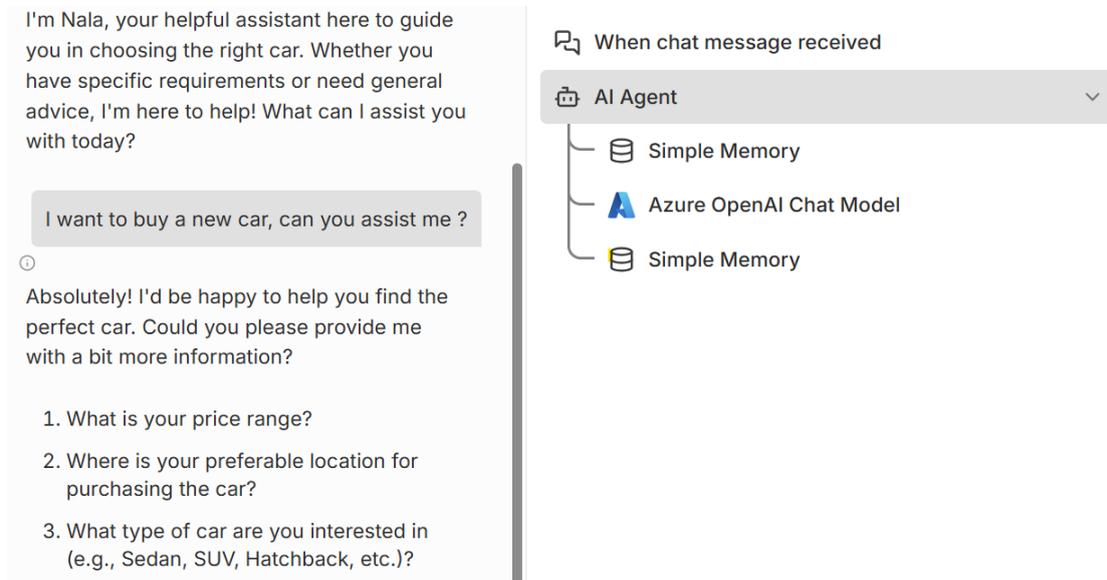
    # Get score_threshold from query parameters, default to 0.88 if not provided
    score_threshold = float(request.args.get('score_threshold', 0.88))

    new_embedding = get_embedding(input_text)
    if new_embedding is not None:
        # Search for similar embeddings
        similarities = search_similarity(new_embedding, score_threshold=score_threshold)
        if similarities:
            app.logger.info("Similar embeddings found")
            return jsonify({'message': 'Similar embeddings found', 'status_code': 200, 'similar_embeddings': similarities}), 200
        else:
            # Insert the new embedding if no similar embeddings found
            success, unique_id = insert_embedding(input_text, new_embedding)
            if success:
                app.logger.info("Embedding inserted successfully")
                return jsonify({'message': 'Embedding inserted successfully', 'status_code': 201, 'id': unique_id}), 201
            else:
                app.logger.error("Failed to insert embedding")
                return jsonify({'message': 'Failed to insert embedding', 'status_code': 500}), 500
    else:
        app.logger.error("Failed to retrieve embedding")
```

Слика 13. Имплементација на алгоритам за семантичка сличност

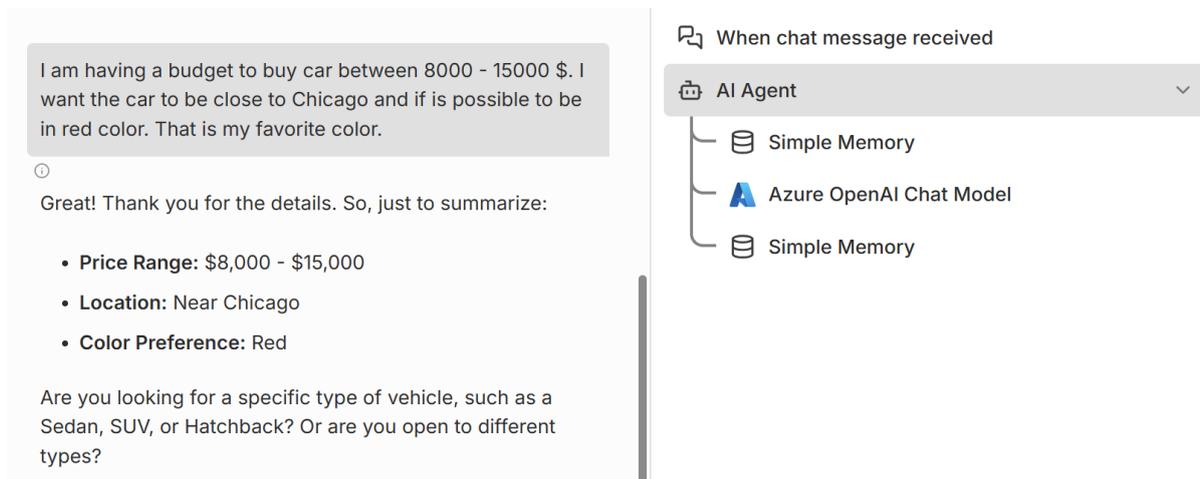
## 5.5. Фаза 5: Персонализирана интеракција со интелигентниот агент

Корисникот ја започнува интеракцијата со цел да пронајде посакуван автомобил без да прави пребарувања на интернет, односно има определен буџет и желба за да го пронајде вистинскиот автомобил за него. Агентот пак поставува одредени прашања за да може да му помогне на корисникот за побрзо да го пронајди посакуваниот автомобил, слика 14.



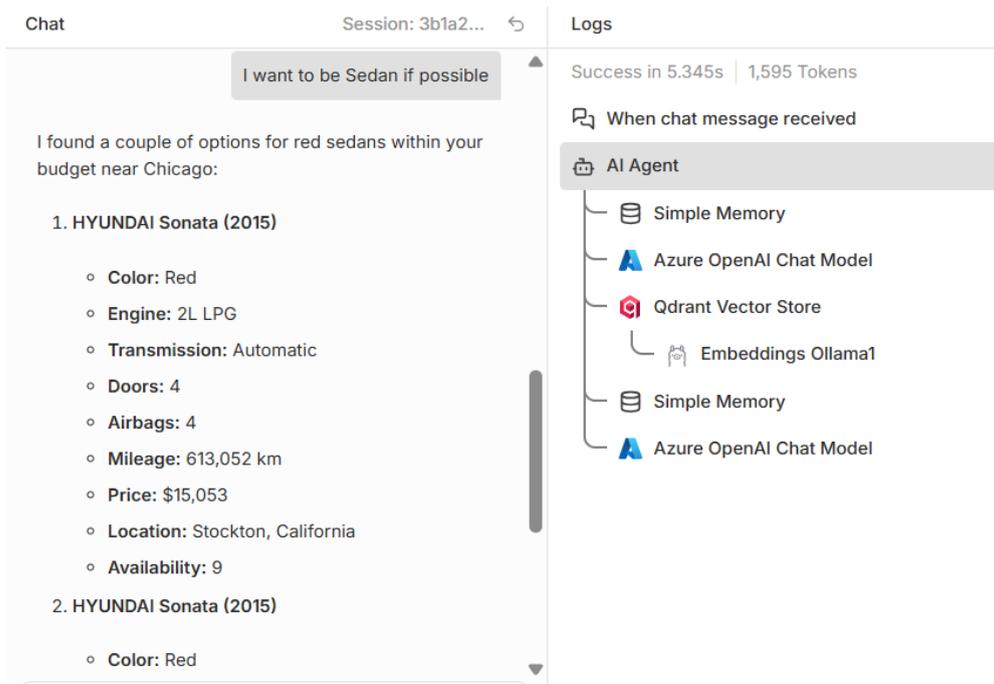
Слика 14: Почетна комуникација помеѓу корисникот со интелигентниот агент

Корисникот споделува со агентот кои се неговите потреби и дополнително освен информации за очекуван буџет и локација, на пребарувањето додава и преференца за боја, како што може да видиме на слика 15.



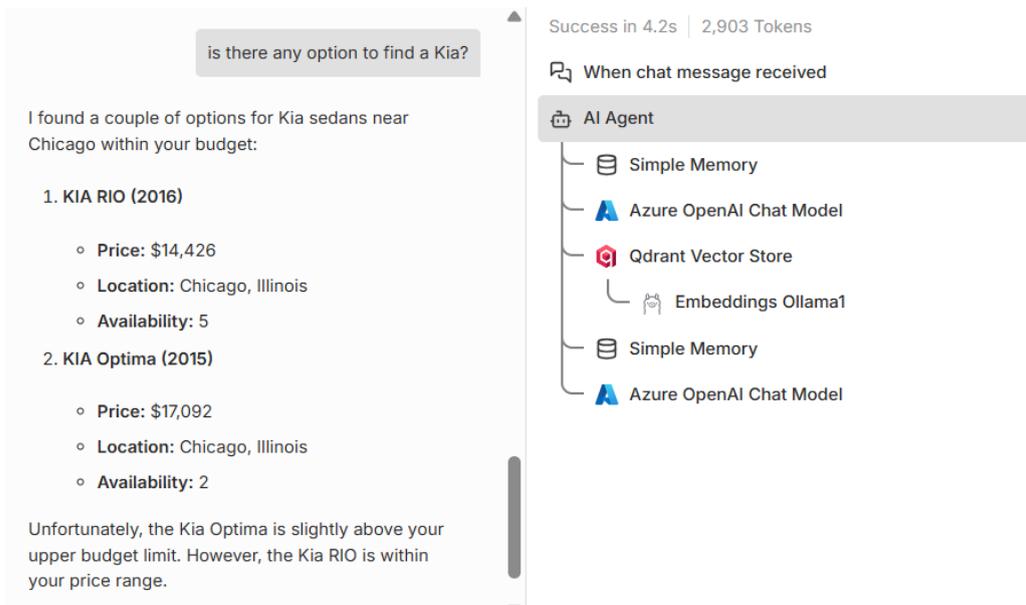
Слика 15: Комуникација со агентот за преференции на автомобил според локација, боја и буџет на корисникот

Пред да направи интелигентниот агент пребарување, прави сумирање на барањата од корисникот и дополнително прашува за тип на модел со цел да направи филтрирање и да овозможи корисникот полесно да го пронајде вистинскиот автомобил за него, слика 16.



Слика 16: Пребарување во вектор базата според избор на модел на автомобилот

Агентот ги доставува селектираните автомобили до корисникот според зададените параметри. Доколку корисникот има дополнителни прашања или пак сака да добие нов предлог според дополнително барање, може да го направи тоа. Како во наредниот пример, слика 17, корисникот прави дополнително пребарување според бренд на автомобил. Агентот прави ново пребарување до векторската база и генерира нов предлог за корисникот каде што наоѓа две опции кои можеби би му одговарале на корисникот.



Слика 17: Корисникот врши пребарување преку конкретизирање на тип на бренд за автомобилот

## **6. Валидација и верификација**

Вгнездената примена на верификација и валидација овозможува намалување на ризикот од критични грешки, оптимизација на процесите и обезбедување на висок квалитет на системите.

Тестирањето на функционалноста на системот ќе се спроведе преку комбинација на мануелно и автоматско тестирање, при што автоматското тестирање ќе се реализира со користење на тест-скрипти. Дополнително, ќе биде извршена паралелна верификација преку имплементација на друг модел, со цел проценка на релевантноста и конзистентноста на излезните податоци.

Понатаму, по целосната имплементација на решението, ќе се спроведе корисничка евалуација преку анкетен прашалник, кој ќе опфати корисници од различни возрастни групи, со цел добивање квантитативни и квалитативни показатели за употребливоста, ефикасноста и целокупното корисничко искуство.

## **7. Мониторинг и анализа на резултатите**

Јасното претставување и интеграција на податоците во персонализирани контролни прегледи ќе доведе до донесување на клучни одлуки. Имплементацијата на мониторинг е од суштинско значење бидејќи овозможува континуирано следење, анализа и оценување на работата на интелегентниот агент и на системите во кои тие се интегрирани. Преку мониторинг се обезбедува навремено откривање на неправилности, грешки или отстапувања во однесувањето на моделите, што директно придонесува за зголемена сигурност, стабилност и доверливост на системот. Дополнително, мониторингот овозможува подобрување на перформансите преку собирање податоци за реалната употреба, што е клучно за понатамошна оптимизација, прилагодување и одржлив развој на ВИ решенијата [21] .

## **8. Евалуација на резултатите и предмет на дискусија**

Истражувањето е во фаза на дизајн и развој. Врз основа на поставените цели и методолошки фази, се предвидува детална анализа врз база на податоци од експериментална или симулациска евалуација. Перформансите на интелегентниот агент ќе се анализираат по неколку критични димензии како технички, интеракциски и персонализирани за различни корисници на системот. Целта на оваа анализа е да се процени ефикасноста и применливоста на агентот.

Спроведено е иницијално корисничко тестирање со цел евалуација на функционалноста и квалитетот на интеракцијата на агентот. Тестирањето беше насочено кон неколку клучни аспекти: разбирање на контекстот, нивото на персонализација на одговорите, брзината на одговор, како и целокупното задоволство од интеракцијата.

Во тестирањето учествуваа помала група од неколку корисници од различни возрастни групи (од 18 до 57 години), што овозможи поширока перспектива. Добиените резултати укажуваат на високо ниво на успешност на агентот, при што повеќето оценки се движат меѓу 4 и 5 на сите евалуирани критериуми. Особено високи оценки се забележани кај разбирањето на контекстот и персонализацијата на одговорите, што укажува на ефективна адаптација на агентот кон индивидуалните кориснички потреби.

Овие иницијални резултати потврдуваат дека интелегентниот агент обезбедува конзистентна, брза и кориснички ориентирана интеракција, и претставуваат солидна основа за понатамошна евалуација и проширување на тестирањето со поголем број корисници.

## 9. Заклучок

Истражувањето за примена на адаптирани јазични модели и векторски бази во развојот на интелегентните агенти со персонализирана интеракција претставува значаен придонес во насока на создавање поинтелигентни, флексибилни системи [36],[37]. Ваквите агенти овозможуваат подлабоко разбирање на контекстот и потребите на корисниците, што резултира со интеракција која е природна, интуитивна и адаптирана на индивидуалниот профил.

Комбинирањето на адаптирани јазични модели со векторски бази претставува клучна технологија која го подобрува начинот на обработка, поврзување и извлекување на информации, овозможувајќи им на агентите да учат од искуство и континуирано да ја подобруваат својата комуникација, [39]. Овој пристап отвора можности за примена во широк спектар на области, каде што персонализираната интеракција станува фактор на ефикасност, продуктивност и конкурентност.

Покрај технолошките придобивки, од особена важност е обезбедувањето на сигурност, транспарентност и етичко функционирање на ваквите системи. Мониторингот, заштитата на податоците и прилагодливоста на моделите кон различни контексти се неопходни предуслови за нивна успешна имплементација.

Во целина, оваа област претставува динамичен и перспективен правец на развој во рамките на современата ВИ, кој овозможува создавање на агенти што не само што комуницираат, туку и разбираат, создавајќи основа за нова ера на интелегентна и персонализирана интеракција помеѓу човекот и машината.

Во идниот период ќе работиме во однос на проширување на истражувачкиот домен и развојот на предвидените чекори за архитектурата на предложеното решение. Како дел од понатамошниот работен тек ќе се фокусираме на анализи од различна природа во однос на интелегентниот агент како и севкупната функционалност на системот.

## 10. Референци

- [1] Insa-Cabrera, J., Dowe, D.L., España-Cubillo, S., Hernández-Lloreda, M.V., Hernández-Orallo, J. (2011). Comparing Humans and AI Agents. In: Schmidhuber, J., Thórisson, K.R., Looks, M. (eds) Artificial General Intelligence. AGI 2011. Lecture Notes in Computer Science(), vol 6830. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-22887-2\\_13](https://doi.org/10.1007/978-3-642-22887-2_13)
- [2] Salunke M, Shah T, Bhokre V, Sreemathy R, (2025), RAG-Based Chatbots for Document Query-A Comprehensive Review Smart Trends in Computing and Communications [https://doi.org/10.1007/978-981-96-7499-2\\_21\(245-256\)](https://doi.org/10.1007/978-981-96-7499-2_21(245-256))
- [3] van Wynsberghe, A. Sustainable AI: AI for sustainability and the sustainability of AI. *AI Ethics* 1, 213–218 (2021). <https://doi.org/10.1007/s43681-021-00043-6>
- [4] Shankar S, Chopra B, Hasan M, Lee S, Hartmann B, Hellerstein J, Parameswaran A, Wu E, (2025), Steering Semantic Data Processing With DocWrangler Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology [https://doi.org/10.1145/3746059.3747625\(1-18\)](https://doi.org/10.1145/3746059.3747625(1-18))
- [5] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, Yue Zhang.(2024) , A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly, High-Confidence Computing, Volume 4, Issue 2,100211, ISSN 2667-2952, <https://doi.org/10.1016/j.hcc.2024.100211>
- [6] Z. Yao, Z. Tang, J. Lou, P. Shen and W. Jia, "VELO: A Vector Database-Assisted Cloud-Edge Collaborative LLM QoS Optimization Framework," 2024 *IEEE International*

- Conference on Web Services (ICWS)*, Shenzhen, China, 2024, pp. 865-876, doi: 10.1109/ICWS62655.2024.00105.
- [7] Pawlik, Lukasz. 2025. "LLM Selection and Vector Database Tuning: A Methodology for Enhancing RAG Systems" *Applied Sciences* 15, no. 20: 10886. <https://doi.org/10.3390/app152010886>
- [8] J. Johnson, M. Douze and H. Jégou, "Billion-Scale Similarity Search with GPUs," in *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535-547, 1 July 2021, doi: <https://doi.org/10.1109/TBDATA.2019.2921572>
- [9] (2006). Foundations of Metric Space Searching. In: Similarity Search The Metric Space Approach. *Advances in Database Systems*, vol 32. Springer, Boston, MA. [https://doi.org/10.1007/0-387-29151-2\\_1](https://doi.org/10.1007/0-387-29151-2_1)
- [10] Borghoff, U. M., Bottoni, P., and Pareschi, R. (2025). Human-artificial interaction in the age of agentic AI: a system-theoretical approach. *arXiv Preprint, arXiv:2502.14000*. doi: 10.48550/arXiv.2502.14000
- [11] Bradshaw, J. M., Feltovich, P. J., and Johnson, M. (2011). "Human-agent interaction," in *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*, eds. G. A. Boy (CRC Press), 283–300. doi: 10.1201/9781315557380-14
- [12] Capitanelli, A., and Mastrogiovanni, F. (2024). A framework for neurosymbolic robot action planning using large language models. *Front. Neurobotics*. 18:1342786. doi: 10.3389/fnbot.2024.1342786
- [13] Schwartz Sivan, Yaeli Avi, Shlomov Segev, (2023), Enhancing Trust in LLM-Based AI Automation Agents: New Considerations and Future Challenges. In: *Computer Science at Cornell University*, <https://doi.org/10.48550/arXiv.2308.05391>
- [14] Choudhuri, R., Liu, D., Steinmacher, I., Gerosa, M., and Sarma, A. (2024). "How far are we? the triumphs and trials of generative AI in learning software engineering," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 1–13. doi: 10.1145/3597503.3639201
- [15] Karlan, B. (2023). Human achievement and artificial intelligence. *Ethics Inf. Technol.* 25:40. doi: 10.1007/s10676-023-09713-x
- [16] A. Trajkovska, B. Ristevski, T. Trajkov, N. Rendevski and K. Veljanovska, "From Data to Decisions: Real-Time Analytics and ML with Kafka and Databricks," 2025 60th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Ohrid, North Macedonia, 2025, pp. 1-4, doi: 10.1109/ICEST66328.2025.11098429
- [17] Trajkovska A, Jolevski I, (2024). Comparative Sentiment Analysis and Semantic Meaning in Text using sentiment models from Hugging Face and Power Automate. XVI International Conference ETAI 2024, Struga, North Macedonia, ISSN 2545-4990, Vol.3, Issue 1, pp.182-187, <https://doi.org/10.5281/zenodo.14054137>
- [18] Trajkovska A, Markoski A, (2024) AI at the edge: trends and innovations in tiny machine learning models for iot and embedded systems in synergy with neutron. AI. In: *Proceedings of the 14th International Conference on Applied Internet and Information Technologies AIIT* (pp. 304–311). University of Novi Sad, Technical faculty "Mihajlo Pupin", Zrenjanin, Republic of Serbia
- [19] Trajkovska A, Veljanovska K, (2024) Robotic Process Automation Implementation for Streamlining Repetitive Administrative Tasks in Synergy with Artificial Intelligence. In: 16th International conference ICT Innovations, [https://eprints.uklo.edu.mk/id/eprint/10785/1/webproc\\_final\\_1.pdf](https://eprints.uklo.edu.mk/id/eprint/10785/1/webproc_final_1.pdf)
- [20] Trajkovska A, Dimovski T, Markoska R, and Kotevski Z. (2023) "Automation and Monitoring on Integration ETL Processes while Distributing Data." pp.: 212-219. In: *Proceedings of the 13th International Conference on Applied Internet and Information*

Technologies AIT, Faculty of Information and Communication Technologies-Bitola, Republic of North Macedonia.

- [21] A. Trajkovska, A. Bocevska and B. Ristevski, "3D Scientific Visualization Using Mayavi," 2024 59th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Sozopol, Bulgaria, 2024, pp. 1-4, doi: 10.1109/ICEST62335.2024.10639745.
- [22] Veljanovska, Kostandina and Trajkovska, Aneta and Veljanovski, Nikolce (2024) *KNN Algorithm Implementation in Real-World Problem of Water Quality Classification*. In: 14th International conference on Applied Internet and Information Technologies (AIIT2024), November 8th 2024, Zrenjanin, Serbia.
- [23] Trajkovska, Aneta, Blagoj Ristevski, Kostandina Veljanovska, Trajche Trajkov, and Nikola Rendevski. "Optimizing Real-Time Data Processing with Kafka and Databricks Integration for Scalable Machine Learning Solutions." (2025): 187-194.
- [24] Trajkovska, Aneta, Violeta Manevska, Kostandina Veljanovska. "Scalable ETL Processes with Change Data Capture (CDC) and Monitoring Using Apache Superset." (2025): 118-125.
- [25] Liu, Xiao, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. "GPT understands, too." *AI Open* 5 (2024): 208-215.
- [26] Jackson, Freeman. "A Unified Governance Architecture for Agentic AI: Integrating Microsoft Azure AI Foundry, Cloud-Native Guardrails, Frontier AI Safety Research, Zero-Trust Principles, and the 4th. GRC Governance Engine." *GRC Governance Engine* (November 20, 2025) (2025).
- [27] Li, Xiu, Aron Henriksson, Martin Duneld, Jalal Nouri, and Yongchao Wu. "Evaluating embeddings from pre-trained language models and knowledge graphs for educational content recommendation." *Future Internet* 16, no. 1 (2023): 12.
- [28] Chandrasekaran, Dhivya, and Vijay Mago. "Evolution of semantic similarity—a survey." *Acm Computing Surveys (Csur)* 54, no. 2 (2021): 1-37.
- [29] Zheng, Hongling, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, Yonggang Wen, and Dacheng Tao. "Learning from models beyond fine-tuning." *Nature Machine Intelligence* 7, no. 1 (2025): 6-17.
- [30] Church, Kenneth Ward, Zeyu Chen, and Yanjun Ma. "Emerging trends: A gentle introduction to fine-tuning." *Natural Language Engineering* 27, no. 6 (2021): 763-778.
- [31] Arslan, Muhammad, Hussam Ghanem, Saba Munawar, and Christophe Cruz. "A Survey on RAG with LLMs." *Procedia computer science* 246 (2024): 3781-3790.
- [32] Balaguer, Angels, Vinamra Benara, Renato Luiz de Freitas Cunha, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg et al. "RAG vs fine-tuning: pipelines, tradeoffs, and a case study on agriculture." *arXiv preprint arXiv:2401.08406* (2024).
- [33] Vieira, Marco Antônio Athayde de Aguiar. "Improving approximate nearest neighbor search in HNSW graphs with data clustering." (2025).
- [34] Xiao, Wentao, Yueyang Zhan, Rui Xi, Mengshu Hou, and Jianming Liao. "Enhancing HNSW Index for Real-Time Updates: Addressing Unreachable Points and Performance Degradation." *arXiv preprint arXiv:2407.07871* (2024).
- [35] Foster, Cole, and Benjamin Kimia. "Computational enhancements of HNSW targeted to very large datasets." In *International Conference on Similarity Search and Applications*, pp. 291-299. Cham: Springer Nature Switzerland, 2023.
- [36] Rizinski, Maryan, and Dimitar Trajanov. "AI Agents in Finance and Fintech: A Scientific Review of Agent-Based Systems, Applications, and Future Horizons." *Computers, Materials and Continua* 86, no. 1 (2025): 1-34.

- [37] Kumar, Sanjay Nakharu Prasad. "Multi-agent AI systems in finance: Models, applications, and challenges." *International Journal of Advanced Research in Computer Science & Technology (IJARCST)* 8, no. 1 (2025): 11555-11573.
- [38] Jeong, Cheonsu. "A study on the implementation method of an agent-based advanced rag system using graph." *arXiv preprint arXiv:2407.19994* (2024).
- [39] Zhao, Xiuyuan, Tiejiang Sun, Shaochen Ren, Jingyun Yang, and Yang Liu. "RAG-Based AI Agents for Enterprise Software Development: Implementation Patterns and Production Deployment." *Frontiers in Artificial Intelligence Research* 2, no. 3 (2025): 501-520.