**Професионална гимназия по електроника и енергетика, гр. Банско**

**България, Банско 2770, ул. "България**" **№23,**

**тел: 074988402, е-mail: pgeebansko@abv.bg**

|  |
| --- |
|  |

Дипломна работа

Тема:

Програма на Python за решаване а квадратно уравнение

## Съдържание

[Съдържание 2](#_Toc132185396)

[I. Увод 3](#_Toc132185397)

[II. Основна част 5](#_Toc132185398)

[Изкуственият интелект (ИИ) 5](#_Toc132185399)

[Как ще изглежда живота ни след появата на ИИ? 7](#_Toc132185400)

[Какви са предимствата и недостатъците на изкуствения интелект? 7](#_Toc132185401)

[Какви са приложенията на ИИ? 10](#_Toc132185402)

[Увеличен интелект. 11](#_Toc132185403)

[Невронни мрежи 12](#_Toc132185404)

[Как работят невронните мрежи? 13](#_Toc132185405)

[Видове невронни мрежи 15](#_Toc132185406)

[История на невронните мрежи 16](#_Toc132185407)

[Избор на библиотеки за разпознаване на реч в Python 16](#_Toc132185408)

[Гласови асистенти 21](#_Toc132185409)

[Гласови асистенти 22](#_Toc132185410)

[Защо избирам Python за разработка на гласовия асистент 24](#_Toc132185411)

[III. Разработка 27](#_Toc132185412)

[IV. Заключение 32](#_Toc132185413)

[V. Използвана литература 33](#_Toc132185414)

# Увод

Тъй като технологиите продължават да напредват, все повече хора се обръщат към гласови асистенти, за да им помогнат с ежедневните задачи. Гласовите асистенти използват обработка на естествен език, за да разберат и отговорят на изговорени команди. Със способността да изпълняват задачи като възпроизвеждане на музика, задаване на напомняния и предоставяне на информация при търсене, гласовите асистенти стават все по-популярен инструмент в домакинствата и на работните места по света. Създаването на гласов асистент не е лесна задача, но може да бъде възнаграждаващ и вълнуващ проект както за разработчиците, така и за компаниите. Процесът включва разработване на софтуер, който може да разпознава и отговаря на изговорени команди, както и изграждане на база данни с информация, която асистентът може да използва, за да предостави полезни отговори и препоръки. Крайният резултат е базиран на изкуствен интелект инструмент, който може да помогне на потребителите да вършат нещата по-ефективно, без да е необходимо физическо взаимодействие с техните устройства. В момента на пазара има редица популярни гласови асистенти, всеки със своите силни и слаби страни. Alexa например, е известна с широката си гама от съвместими интелигентни домашни устройства и възможността си да поръчва продукти през онлайн магазина на Amazon. Google Assistant, от друга страна, е известен със своите превъзходни възможности за обработка на естествен език, което улеснява потребителите да взаимодействат с него по по-разговорен начин. Siri на Apple също се използва широко, особено сред потребителите на устройства на Apple. Въпреки многото предимства на гласовите асистенти, има и някои потенциални недостатъци, които трябва да имаме предвид. От една страна, има опасения относно поверителността и сигурността на данните, тъй като гласовите асистенти трябва постоянно да слушат за изговорени команди, за да функционират. Това довежда до известен спор дали гласовите асистенти винаги записват и предават потребителски данни и как се използват тези данни. Освен това гласовите асистенти не са перфектни и понякога могат да тълкуват погрешно команди или да предоставят неточна информация. Въпреки тези опасения, гласовия асистент продължава да се разпространява и популяризира във всички области на техниката. Тъй като технологията продължава да се развива и подобрява, вероятно ще видим още по-усъвършенствани гласови асистенти през идните години, с още по-голям потенциал да ни помогнат да управляваме натоварения си живот. Този проект има за цел да предостави услугите на гласово управление. Приложението може да манипулира системата и да изпълнява основни задачи чрез разпознаване на реч. В днешния свят на нарастващо търсене на лични асистенти, този проект предлага решение, което опростява функционалността на съвременните асистенти. Стабилният растеж на тази технология в различни области доказва своята компетентност в сравнение с автоматизацията, базирана на текст и изображения. Този проект е самостоятелно приложение, което може да се използва от всеки, който не познава работата с гласовия асистент. Може да изпълнява задачи, като отваряне на приложения (като Word, Google и др.) отваряне на часовник или аларми. Тези операции могат да се извършват от произнасяне на командите на глас.

Може да анализира и синтезира реч и тук, поддържаният език ще бъде български. Гласовият асистент позволява на потребителите да използват приложението без използването на ръце, само с гласови команди, за да показва уебсайтове, отваряне и изпълняване на други приложения. Той ще разпознае изречените думи, ще ги преобразува в изречения и след това ще ги обработи в задача за изпълнение. Машинно изпълнимите команди ще взаимодействат с други приложения и ще връщат обратна връзка на потребителя, като резултат от търсенето. Този проект може да бъде разширен, за да бъде приложен върху различни приложения. Може да се използва на персонални компютри, работещи с всяка операционна система.

Дори човек, който не знае как да работи с гласовите асистенти, може лесно и ефективно да получи достъп с помощта на такова приложение. В този проект автоматизацията се извършва с помощта на Python. Чрез програмиране на Python ние също имаме достъп до сървъра на Google, запис, аудио от системния микрофон и др. Има два различни Python файлове, включени в този проект, един за аудио разпознаване и един за изпълняваща системи задачи. Файловете на Python се изпълняват като дъщерни процеси (процеси на рендеринг) заедно с основния електронен процес,

# Основна част

## Изкуственият интелект (ИИ)

Гласовите асистенти са вид технология за изкуствен интелект (ИИ), която използва обработка на естествен език и алгоритми за машинно обучение, за да разбере и отговори на гласови команди и въпроси от потребителите. Гласовите асистенти като Siri, Alexa, Google Assistant и Cortana използват ИИ, за да интерпретират речта, да разпознават шаблони в езика и да се учат от поведението и предпочитанията на потребителите с течение на времето. Те използват тези данни, за да подобрят своите отговори и да направят персонализирани препоръки към потребителите. ИИ играе решаваща роля в разработването на гласови асистенти, като им позволява непрекъснато да се учат и адаптират към променящите се потребителски нужди и предпочитания. С помощта на ИИ гласовите асистенти стават все по-напреднали и усъвършенствани, предлагайки на потребителите по-персонализирани и точни изживявания.

Изкуственият интелект (ИИ) е широка област на компютърните науки, която се фокусира върху създаването на роботи, които могат да изпълняват задачи, които обикновено изискват човешки интелект, като разпознаване на модели, разбиране на естествен език, вземане на решения и обучаване към желанията на потребителя. ИИ използва алгоритми, статистически модели и други техники, за да анализира данни и да прави прогнози или решения въз основа на тези данни. Той се използва в широк спектър от приложения, от гласови асистенти и самоуправляващи се автомобили до медицинска диагностика и финансов анализ. С напредването на технологиите се очаква изкуствения интелект да играе все по-важна роля в живота ни, трансформирайки начина, по който работим, комуникираме и взаимодействаме със света около нас.

Историята на изкуствения интелект (ИИ) датира от средата на 20-ти век, когато компютърните учени започнаха да изследват идеята за създаване на машини, които могат да изпълняват задачи, които обикновено изискват човешки интелект. През 50-те и 60-те години на миналия век изследователите разработиха програми, които можеха да играят игри, като шах и дама. Тези ранни програми разчитаха на прости системи, базирани на правила, и символно мислене, за да симулират интелигентно поведение. През 70-те години на миналия век изследванията на ИИ се насочиха към системи, базирани на знания и експертни системи. Тези системи използват правила и логически изводи за решаване на проблеми в специфични области, като медицинска диагноза или инженерен дизайн. През 80-те и 90-те години на миналия век изследванията на ИИ се разширяват, за да включат невронни мрежи, които са моделирани според структурата на човешкия мозък и могат да се учат от примери. Това доведе до значителен напредък в машинното обучение и компютърното зрение (термин, с който се означават група изследвания в областта на компютърните науки). През последните години развитието на техники за задълбочено обучение позволи на компютрите да обработват и анализират огромни количества данни, което доведе до пробиви в области, като разпознаване на реч, обработка на естествен език и разпознаване на изображения. Като цяло историята на ИИ е белязана от периоди на интензивни изследвания и разработки, както и периоди на застой и разочарование. Трябва ли ни да знаем какво е ИИ? На този въпрос ще отговоря просто: Да, ако искаме да го открием, то определено ще е по-лесно да го намерим, ако знаем какво търсим. В противен случай ще се окажем в положението на Алхимиците, които са търсили Философския камък, но почти не са имали представа какво е това. Най-известната дефиниция на ИИ е така наречения Трингов тест.

Тюринг е английски математик известен, както с Машините на Тюринг, така и с разбиването на немските кодове по време на Втората Световна Война. Тринговия тест е доста прост. Поставяме нещо зад една завеса и то разговаря с нас. Ако не можем да го различим от човек, то това е изкуствен интелект. Тази дефиниция е по-стара от петдесет години и затова ще се опитаме да направим нова, по-съвременна. Дефиницията на Тюринг предполага, че интелект е човек с натрупаните през годините знания. Интелект ли е бебето, което току що се е появило на бял свят? Нашият отговор ще е: Да. Тоест нашата дефиниция за Интелект ще е това, което не знае нищо, но може да се научи. Тук се различаваме от повечето хора, които като чуят интелект, си представят професор от университета. Изкуствената обща интелигентност се отнася до идеята за създаване на ИИ, която може да мисли като човек и да решава широк спектър от задачи, което може да бъде сравнимо с човешката интелигентност. Областите, които използват ИИ, са много разнообразни, включително машинно обучение, обработка на език, роботика и компютърно зрение. ИИ има приложения в различни области на живота, като медицината, автомобилната индустрия, търговията, финансите и други.



Защо е важен изкуственият интелект?

ИИ е важен, когато става въпрос за повтарящи се задачи, ориентирани към детайлите, като анализиране на голям брой правни документи, за да се гарантира, че съответните полета са попълнени правилно, инструментите с изкуствен интелект често изпълняват задачи бързо и с относително малко грешки. Преди настоящата вълна от ИИ би било трудно да си представим използването на компютърен софтуер за свързване водачите на таксита, но днес Uber се превърна в една от най-големите компании в света, правейки точно това. Той използва усъвършенствани алгоритми за машинно обучение, за да предвиди кога е вероятно хората да се нуждаят да достигнат до определени райони, което помага проактивно да накара шофьорите да тръгнат на пътя, преди да са необходими. Като друг пример, Google се превърна в един от най-големите играчи за набор от онлайн услуги, като използва машинно обучение, за да разбере как хората използват техните услуги и след това да ги подобри.

През 2017 г. главният изпълнителен директор на компанията, Сундар Пичай, обяви, че Google ще работи като компания, която е „Първо ИИ“. Днешните най-големи и успешни предприятия са използвали изкуствени интелекти, за да подобрят своите операции и да получат предимство пред своите конкуренти.

## Как ще изглежда живота ни след появата на ИИ?

Идеята на ИИ е, че човека създава робот, който е несравнимо по-умно от него самия. Макар, че това същество ще е добронамерено и ще ни служи вярно, за нас то може да е проблем, защото сега ние се гордеем с това, че сме най-умните и че сме по-умни от всички други животни и дори от машините. Нашия интелект е това, което ни дава самочувствието, че сме върха на еволюцията. Сега ние управляваме планетата Земя и ние решаваме кое животно и кое растение заслужава да живее и кое да се размножи и да заеме повече площ и кое да бъде ограничено само в резерватите.

Как ще изглежда нашият живот след появата на ИИ? Тогава живота ни ще бъде много лесен. Няма да има нужда да мислим за прехраната си, няма да ни се налага да работим, дори няма да е нужно да се забавляваме един друг, защото ИИ ще ни забавлява много по-добре, отколкото който и да е човек би ни забавлявал.

След появата на ИИ почти никой няма да умира. Човешкото тяло може да се ремонтира и да се клонира и да продължи да съществува практически вечно, но ние може да решим да поставим граница и да кажем, че никой няма да има право да живее повече от 120 години.

Колко хора ще оставим да живеят на Земята? Може да са 7 милиарда, може да ги увеличим до 70 или до 700, но може би е добре да се сложи някаква граница, защото ако сме прекалено много ще започнем да си пречим, а и няма да остане никакво място за другите.

Какво ще правим след появата на ИИ? След като няма да работим, единственото смислено нещо ще е да се отдадем на размножаване. То и сега размножаването е най-важното, но сега ние работим, за да се размножим. Когато работата не е важна, няма да са важни и парите, защото с пари измерваме труда на хората, тогава кой ще е новия критерии на естествения подбор? Сега критериите са: интелект, красота, здраве, образование, сила, смелост, бързина, честност, религия и мироглед. Силата и бързината са били много важни в миналото, но сега когато машините са много по-силни и по бързи от нас хората, силата и бързината не са най-важното. Когато машините станат по-умни от нас, тогава и интелекта няма да е най-важното. Смелостта е сложен критерии. От една страна печелят смелите, но от друга, най-смелите си чупят главата. Подобно е и положението с честността. Най-успешни са бизнесмените и политиците, които не блестят с особена честност, но най-нечестните влизат в затвора. Образованието е било еволюционно предимство в миналото, но днес то е по-скоро недостатък.

## Какви са предимствата и недостатъците на изкуствения интелект?

Изкуствените невронни мрежи и технологиите за изкуствен интелект за задълбочено обучение се развиват бързо, главно защото ИИ обработва големи количества данни много по-бързо и прави прогнози по-точни, отколкото е възможно на човека. Въпреки че огромният обем данни, създаван ежедневно, би погребал човешки изследовател, ИИ приложенията, които използват машинно обучение , могат да вземат тези данни и бързо да ги превърнат в полезна информация. Към момента на писане, основният недостатък на използването на ИИ е, че е скъпо да се обработват големите количества данни, които програмирането на ИИ изисква.

Предимства

* Добър в детайлно ориентирани работни места;
* Намалено време за тежки задачи с данни;
* Осигурява постоянни резултати и задвижваните от ИИ виртуални машини са винаги на разположение.
* Прост и лесен за работа с него.

Недостатъци

* Скъп.
* Изисква дълбока техническа експертиза;
* Ограничено предлагане на квалифицирани работници за изграждане на инструменти за изкуствен интелект;
* Знае само това, което му е показано;
* Липса на способност за обобщаване и преминаване от една задача към друга.

Има 4 типа изкуствен интелект според Аренд Хинце, асистент по интегративна биология и компютърни науки и инженерство в Мичиганския държавен университет, обясни в статия от 2016 г. , че ИИ може да бъде категоризиран в четири типа, като се започне с интелигентните системи, специфични за задачите, които се използват широко днес и се стигне до съзнателни системи , които все още не съществуват.

* Тип 1: Реактивни машини. Тези ИИ системи нямат памет и са специфични за задачите. Пример за това е Deep Blue, шахматната програма на IBM, която победи Гари Каспаров през 90-те години. Deep Blue може да идентифицира фигури на шахматната дъска и да прави прогнози, но тъй като няма памет, не може да използва минали преживявания, за да информира бъдещи такива.
* Тип 2: Ограничена памет. Тези ИИ системи имат памет, така че могат да използват минал опит, за да информират бъдещи решения. Някои от функциите за вземане на решения в самоуправляващите се автомобили са проектирани по този начин.
* Тип 3: Теорията на ума е термин от психологията. Когато се приложи към ИИ, това означава, че системата ще има социалната интелигентност, за да разбере емоциите. Този тип изкуствен интелект ще може да прави изводи за човешките намерения и да прогнозира поведението, което е необходимо умение за системите с изкуствен интелект, за да станат неразделни членове на човешките екипи.

* Тип 4: Самосъзнание. В тази категория ИИ системите имат самочувствие, което им дава съзнание. Машините със самосъзнание разбират собственото си текущо състояние. Този тип ИИ все още не съществува.



ИИ е включен в различни видове технологии. Сега ще разгледаме няколко примера:

* Автоматизация. Когато са сдвоени с ИИ технологии, инструментите за автоматизация могат да разширят обема и видовете изпълнявани задачи. Пример е роботизираната автоматизация на процеси RPA, вид софтуер, който автоматизира повтарящи се, базирани на правила задачи за обработка на данни, традиционно извършвани от хора.

Когато се комбинира с машинно обучение и нововъзникващи ИИ инструменти, RPA може да автоматизира по-големи части от корпоративни задачи, позволявайки на тактическите ботове на RPA да предават информация от ИИ и да реагират на промените в процесите.

* Машинно обучение. Това е науката да накараш компютъра да действа без програмиране. Дълбокото обучение е подмножество от машинно обучение, което с много прости думи може да се разглежда като автоматизация на предсказуемия анализ. Има три вида алгоритми за машинно обучение:

1. Учене под наблюдение . Наборите от данни са етикетирани, така че моделите да могат да бъдат открити и използвани за етикетиране на нови набори от данни.
2. Учене без надзор . Наборите от данни не са етикетирани и са сортирани според прилики или разлики.
3. Обучение с подсилване . Наборите от данни не са етикетирани, но след извършване на действие системата на ИИ получава обратна връзка.

* Машинно зрение. Тази технология дава на машината способността да вижда, улавя и анализира визуална информация с помощта на камера, аналогово-цифрово преобразуване и цифрова обработка на сигнала. Често се сравнява с човешкото зрение, но машинното зрение не е обвързано с биологията и може да бъде програмирано да вижда през стени например. Използва се в редица приложения от идентификация на подпис до анализ на медицински изображения. Компютърното зрение, което е фокусирано върху машинно-базирана обработка на изображения, често се смесва с машинно зрение.

Обработка на естествен език (NLP). Това е обработката на човешкия език от компютърна програма. Един от по-старите и най-известни примери за NLP е откриването на спам, което преглежда темата и текста на имейла и решава дали е боклук. Настоящите подходи към НЛП се основават на машинно обучение. Задачите на НЛП включват превод на текст, анализ на чувствата и разпознаване на реч. В роботиката се фокусира върху проектирането и производството на роботи. Роботите често се използват за изпълнение на задачи, които са трудни за изпълнение или изпълняват последователно от хората. Например, роботите се използват в поточните линии за производство на автомобили или от НАСА за преместване на големи обекти в космоса. Изследователите също използват машинно обучение, за да създават роботи, които могат да взаимодействат в социални условия. Самоуправляващи се автомобили. Автономните превозни средства използват комбинация от компютърно зрение, разпознаване на изображения и дълбоко обучение, за да изградят автоматизирани умения за пилотиране на превозно средство, докато остават в дадена лента и избягват неочаквани препятствия, като пешеходци.

## Какви са приложенията на ИИ?

Изкуственият интелект си проправи път в голямо разнообразие от пазари . Ето осем примера:

ИИ в здравеопазването. Най-големите залози са върху подобряването на резултатите за пациентите и намаляването на разходите. Компаниите прилагат машинно обучение, за да правят по-добри и по-бързи диагнози от хората. Една от най-известните здравни технологии е IBM Watson. То разбира естествения език и може да отговаря на зададени му въпроси. Системата извлича данни за пациенти и други налични източници на данни, за да формира хипотеза, която след това представя със схема за оценка на достоверността. Други ИИ приложения включват използване на онлайн виртуални здравни асистенти и чатботове , за да помогнат на пациентите и клиентите на здравни услуги да намерят медицинска информация, да планират срещи, да разберат процеса на фактуриране и да завършат други административни процеси. Множество ИИ технологии също се използват за прогнозиране, борба и разбиране на пандемии като COVID-19 .

ИИ в бизнеса. Алгоритмите за машинно обучение се интегрират в платформи за анализ и управление на взаимоотношенията с клиенти ( CRM ), за да разкрият информация как да обслужват по-добре клиентите. Чатботовете са включени в уебсайтове, за да осигурят незабавно обслужване на клиентите. Автоматизирането на работните позиции също се превърна в тема на разговор сред академици и ИТ анализатори.

ИИ в образованието**. Изкуственият интелект** може да автоматизира оценяването, като спестява на преподавателите повече време. Той може [да оценява учениците и да се адаптира към техните нужди](https://www.techtarget.com/searchcio/podcast/Cognitive-tutor-could-teach-employees-new-skills) , като им помага да работят със собствено темпо.

**ИИ в банкирането.** Банките успешно използват чатботове, за да информират клиентите си за услугите и да обработват транзакции, които не изискват човешка намеса. ИИ виртуалните асистенти се използват за подобряване и намаляване на разходите за спазване на банковите разпоредби. Банковите организации също използват ИИ, за да подобрят вземането на решения за заеми и да определят кредитни лимити, и да идентифицират инвестиционни възможности.

**ИИ в транспорта.** Допълнение към основната роля на ИИ при управлението на автономни превозни средства, ИИ технологиите се използват в транспорта за управление на трафика, прогнозиране на закъснения на полети и за по-безопасно и ефективно корабоплаване през океана.

Тези термини също така представляват наистина жизнеспособни технологии. Организациите използват машинно обучение в софтуера за управление на информация за сигурността и събития ( [SIEM](https://www.techtarget.com/searchsecurity/definition/security-information-and-event-management-SIEM) ) и свързаните с него области, за да откриват аномалии и идентифицират подозрителни дейности, които показват заплахи. Чрез анализиране на данни и използване на логика за идентифициране на прилики с познат злонамерен код, ИИ може да предостави сигнали за нови и нововъзникващи атаки много по-рано от служителите и предишните технологични итерации.

Узряващата технология играе голяма роля в подпомагането на организациите да се борят с кибератаките. Някои експерти от индустрията смятат, че терминът изкуствен интелект е твърде тясно свързан с популярната култура и това е накарало широката общественост да има невероятни очаквания за това как ИИ ще промени работното място и живота като цяло.

## **Увеличен интелект.**

Някои изследователи и търговци се надяват етикетът увеличен интелект , който има по-неутрална конотация, да помогне на хората да разберат, че повечето реализации на ИИ ще бъдат слаби и просто ще подобрят продуктите и услугите. Примерите включват автоматично извеждане на важна информация в доклади за бизнес разузнаване или подчертаване на важна информация в правни документи.

Истинският ИИ е тясно свързан с концепцията за [технологичната уникалност](https://www.techtarget.com/searchenterpriseai/definition/Singularity-the) - бъдеще, управлявано от изкуствен суперинтелект, който далеч надхвърля способността на човешкия мозък да го разбере или как той оформя нашата реалност. Това остава в сферата на научната фантастика, въпреки че някои разработчици работят по проблема. Мнозина вярват, че технологии като квантовите изчисления могат да играят важна роля за превръщането на AGI в реалност и че трябва да запазим използването на термина ИИ за този вид обща интелигентност.

Докато ИИ инструментите представят набор от нови функционалности за бизнеса, използването на изкуствен интелект повдига и етични въпроси, защото, за добро или лошо, ИИ системата ще подсили това, което вече е научила.

Това може да бъде проблематично, тъй като алгоритмите за машинно обучение, които са в основата на много от най-напредналите ИИ инструменти, са толкова интелигентни, колкото и данните, които им се дават в обучението. Тъй като човешкото същество избира какви данни да се използват за обучение на ИИ програма, потенциалът за [пристрастия при машинно обучение](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-bias-algorithm-bias-or-AI-bias) е присъщ и трябва да се наблюдава отблизо.

Всеки, който иска да използва машинното обучение като част от реални, производствени системи, трябва да вземе предвид етиката в своите процеси на обучение с ИИ и да се стреми да избягва пристрастия. Това е особено вярно при използване на алгоритми, които по своята същност са необясними в приложенията за [задълбочено обучение](https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network) и генеративни състезателни мрежи ( [GAN ).](https://www.techtarget.com/searchenterpriseai/definition/generative-adversarial-network-GAN)

Тези компоненти съставляват отговорно използване на ИИ.

Въпреки потенциалните рискове, понастоящем има малко регулации, регулиращи използването на ИИ инструменти, а там, където съществуват закони, те обикновено се отнасят косвено до ИИ. Например, разпоредбите на Съединените щати за справедливо кредитиране изискват финансовите институции да обясняват кредитните решения на потенциални клиенти.

 Това ограничава степента, до която заемодателите могат да използват алгоритми за задълбочено обучение, които по своето естество са непрозрачни и им липсва обяснение.

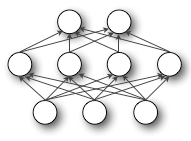
Общият регламент за защита на данните на Европейския съюз ( [GDPR](https://www.techtarget.com/whatis/definition/General-Data-Protection-Regulation-GDPR) ) поставя строги ограничения върху начина, по който предприятията могат да използват потребителски данни, което възпрепятства обучението и функционалността на много насочени към потребителите ИИ приложения.

През октомври 2016 г. Националният съвет за наука и технологии издаде доклад, разглеждащ потенциалната роля, която правителственото регулиране може да играе в развитието на ИИ, но не препоръча да се обмисли конкретно законодателство.

## Невронни мрежи

Невронната мрежа е вид алгоритъм за машинно обучение, който се моделира върху функционирането на мозъка. През 1943 г. двама невролози на име Уорън Маккълох и Уолтър Питс написаха статия, наречена "[Логическо смятане на идеите, предстоящи в нервната дейност](ttps://link.springer.com/article/10.1007/BF02478259)". В него те изразяват функционирането на невроните в мозъка математически, фундаментално достигайки до "принципа "всичко или нищо", който изразява невронната активност по много двоичен начин – ако съобщението от всички входни неврони е достатъчно, невронът се задейства. Ако е недостатъчно, не е така. Това двоично поведение на включване / изключване не отне много време, за да си проправи път до компютърните специалисти, които видяха очевидни връзки със същото поведение на "включване / изключване" на битовете в машината. Франк Розенблат работи с тази концепция за "математически мозък" и въвежда изкуствената невронна мрежа, "Perceptron" в [своята статия](https://pubmed.ncbi.nlm.nih.gov/13602029/) през 1958 година. Това беше много прост математически модел, адаптиран директно от концепциите "всичко или нищо", изложени от McCulloch и Pitts, където беше добавено теглото на група входни данни за модела. Ако достигне критичен праг, "невронът" се изстрелва и определена категория се връща. Ако не е достигнал този праг, не е било. Ще ви дадем пример за това как това може да се приложи към вашето обучение по-късно, но първо може да се изненадате да научите, че това не е червено горещо явление.

Невронните мрежи се опитват да подражават на човешкия мозък, комбинирайки компютърни науки и статистика за решаване на общи проблеми в областта на изкуственият интелект. Невронните мрежи, известни също като изкуствени невронни мрежи или симулирани невронни мрежи, са подгрупа на  машинно обучение  и са в основата на   алгоритмите [за дълбокото обучение](https://www-ibm-com.translate.goog/topics/deep-learning?_x_tr_sl=en&_x_tr_tl=bg&_x_tr_hl=bg&_x_tr_pto=wapp)Името и структурата им са вдъхновени от човешкия мозък, имитирайки начина, по който биологичните неврони сигнализират един на друг. Изкуствените невронни мрежи се състоят от възлови слоеве, съдържащи входен слой, един или повече скрити слоеве и изходен слой. Всеки възел или изкуствен неврон се свързва с друг и има свързано тегло и праг. Ако изходът на всеки отделен възел е над определената прагова стойност, този възел се активира, изпращайки данни към следващия слой на мрежата. В противен случай не се предават данни към следващия слой на мрежата.

Невронните мрежи разчитат на данни за обучение, за да научат и подобрят своята точност с течение на времето. Въпреки това, след като тези алгоритми за обучение са фино настроени за точност, те са мощни инструменти в компютърните науки и изкуствен интелект , позволявайки ни да класифицираме и групираме данни с висока скорост. Задачите за разпознаване на реч или разпознаване на изображения могат да отнемат минути срещу часове в сравнение с ръчната идентификация от човешки експерти. Една от най-известните невронни мрежи е алгоритъмът за търсене на Google.

## Как работят невронните мрежи?

Мислете за всеки отделен възел като за собствен [линеен регресионен](https://www-ibm-com.translate.goog/topics/linear-regression?_x_tr_sl=en&_x_tr_tl=bg&_x_tr_hl=bg&_x_tr_pto=wapp) модел, съставен от входни данни, тегла, отклонение (или праг) и изход. Формулата ще изглежда така:

∑wixi + отклонение = w1x1 + w2x2 + w3x3 + отклонение

изход = f(x) = 1, ако ∑w1x1 + b>= 0; 0, ако ∑w1x1 + b < 0

След като се определи входен слой, се присвояват тегла. Тези тегла помагат да се определи важността на дадена променлива, като по-големите допринасят по-значително за изхода в сравнение с други входове. След това всички входове се умножават по съответните им тегла и след това се сумират. След това изходът преминава през функция за активиране, която определя изхода. Ако този изход надвиши даден праг, той задейства възела, предавайки данни на следващия слой в мрежата. Това води до това, че изходът на един възел се превръща във вход на следващия възел. Този процес на предаване на данни от един слой към следващ слой определя тази невронна мрежа като мрежа за предаване напред. Нека разбием как може да изглежда един възел, използвайки двоични стойности. Можем да приложим тази концепция към по-осезаем пример, като например дали трябва да отидете на сърф (Да: 1, Не: 0). Решението да отидем или да не отидем е нашият прогнозиран резултат или y-hat. Да приемем, че има три фактора, които влияят върху вземането на решение:

* Добри ли са вълните? (Да: 1, Не: 0)
* Съставът празен ли е? (Да: 1, Не: 0)
* Имало ли е наскоро нападение от акула? (Да: 0, Не: 1)

Тогава нека приемем следното, което ни дава следните входове:

* X1 = 1, тъй като вълните се изпомпват
* X2 = 0, тъй като тълпите са навън
* X3 = 1, тъй като не е имало скорошна атака на акула

Сега трябва да присвоим някои тегла, за да определим важността. По-големите тегла означават, че определени променливи са от по-голямо значение за решението или резултата.

* W1 = 5, тъй като големи подувания не се появяват често
* W2 = 2, тъй като сте свикнали с тълпите
* W3 = 4, тъй като имате страх от акули

И накрая, ние също ще приемем прагова стойност от 3, което би означавало стойност на отклонение от –3. С всички различни входове можем да започнем да включваме стойности във формулата, за да получим желания резултат.

Y-шапка = (1\*5) + (0\*2) + (1\*4) – 3 = 6

Ако използваме функцията за активиране от началото на този раздел, можем да определим, че изходът на този възел ще бъде 1, тъй като 6 е по-голямо от 0. В този случай ще отидете да сърфирате; но ако коригираме теглата или прага, можем да постигнем различни резултати от модела. Когато наблюдаваме едно решение, както в горния пример, можем да видим как една невронна мрежа може да взема все по-сложни решения в зависимост от резултата от предишни решения или слоеве. В примера по-горе използвахме перцептрони, за да илюстрираме част от математиката, която се играе тук, но невронните мрежи използват сигмоидни неврони, които се отличават със стойности между 0 и 1. Тъй като невронните мрежи се държат подобно на дърветата на решенията, каскадните данни от един възел към друг, наличието на x стойности между 0 и 1 ще намали въздействието на всяка дадена промяна на една променлива върху изхода на всеки даден възел и впоследствие изхода на невронната мрежа. Когато започнем да мислим за по-практични случаи на използване на невронни мрежи, като разпознаване на изображения или класификация, ние ще използваме контролирано обучение или етикетирани набори от данни, за да обучим алгоритъма. Докато обучаваме модела, ще искаме да оценим точността му с помощта на функция за цена (или загуба). Това също обикновено се нарича средна квадратична грешка.

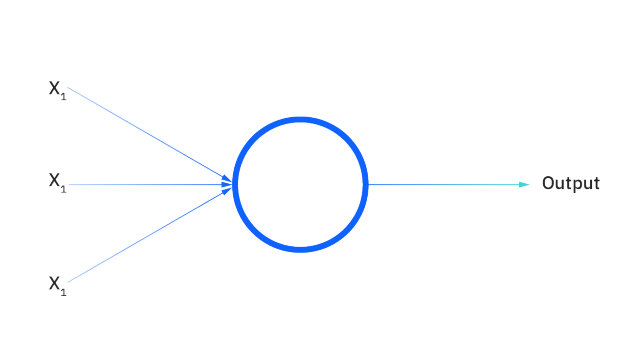
* i представлява индекса на извадката,
* y-hat е прогнозираният резултат,
* y е действителната стойност и
* m е броят на пробите.

𝐶𝑜𝑠𝑡 𝐹𝑢𝑛𝑐𝑡𝑖𝑜𝑛= 𝑀𝑆𝐸=1/2𝑚 ∑129\_(𝑖=1)^𝑚▒(𝑦 ̂^((𝑖) )−𝑦^((𝑖)2)

В крайна сметка, целта е да минимизираме нашата функция на разходите, за да гарантираме коректността на съответствие за всяко дадено наблюдение. Тъй като моделът коригира своите тегла и отклонения, той използва функцията на разходите и обучението за подсилване, за да достигне точката на конвергенция или локалния минимум. Процесът, при който алгоритъмът коригира своите тегла, е чрез градиентно спускане, което позволява на модела да определи посоката, която да поеме, за да намали грешките (или да минимизира функцията на разходите). С всеки пример за обучение параметрите на модела се настройват, за да се сближат постепенно до минимума.

## Видове невронни мрежи

Невронните мрежи могат да бъдат класифицирани в различни типове, които се използват за различни цели. Въпреки че това не е изчерпателен списък от типове, по-долу ще бъде представителен за най-често срещаните типове невронни мрежи, които ще срещнете за обичайните случаи на употреба: Перцептронът е най-старата невронна мрежа, създадена от Франк Розенблат през 1958 г.

Невронните мрежи с предварителна връзка или многослойните перцептрони (MLP). Те се състоят от входящ слой, скрит слой или слоеве и изходен слой. Въпреки че тези невронни мрежи също често се наричат ​​MLP, важно е да се отбележи, че те всъщност се състоят от сигмоидни неврони, а не от перцептрони, тъй като повечето проблеми в реалния свят са нелинейни. Данните обикновено се въвеждат в тези модели, за да ги обучават, и те са основата за компютърно зрение, [обработка на естествен език](https://www-ibm-com.translate.goog/topics/natural-language-processing?_x_tr_sl=en&_x_tr_tl=bg&_x_tr_hl=bg&_x_tr_pto=wapp) и други невронни мрежи. Конволюционните невронни мрежи (CNN) са подобни на мрежите за предаване напред, но обикновено се използват за разпознаване на изображения, разпознаване на образи и/или компютърно зрение. Тези мрежи използват принципи от линейната алгебра, особено умножение на матрици, за идентифициране на модели в изображение. [Повтарящите се невронни мрежи (RNN)](https://www-ibm-com.translate.goog/topics/recurrent-neural-networks?_x_tr_sl=en&_x_tr_tl=bg&_x_tr_hl=bg&_x_tr_pto=wapp) се идентифицират чрез техните вериги за обратна връзка. Тези алгоритми за обучение се използват предимно при използване на данни от времеви серии, за да се правят прогнози за бъдещи резултати, като например прогнози за фондовия пазар или прогнози за продажби.

## 

## История на невронните мрежи

Историята на невронните мрежи е по-дълга, отколкото повечето хора си мислят. Въпреки че идеята за „машина, която мисли“ може да бъде проследена до древните гърци, ние ще се съсредоточим върху ключовите събития, довели до еволюцията на мисленето около невронните мрежи, чиято популярност намалява и намалява през годините:

**1943:** Warren S. McCulloch и Walter Pitts публикуват „ [Логическо смятане на идеите, присъщи на нервната дейност](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://web.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf) (PDF, 1 MB) (връзката е извън ibm.com)“ Това изследване се стреми да разбере как човешкият мозък може да произвежда сложни модели чрез свързани мозъчни клетки или неврони. Една от основните идеи, произлезли от тази работа, беше сравнението на неврони с двоичен праг с булева логика (т.е. 0/1 или вярно/невярно твърдения).

**1958:** На Франк Розенблат се приписва разработката на перцептрона, документиран в неговото изследване „ [Персептронът: вероятностен модел за съхранение на информация и организация в мозъка](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://www.academia.edu/60542953/The_perceptron_a_probabilistic_model_for_information_storage_and_organization_in_the_brain) “ (връзката се намира извън ibm.com). Той издига работата на Маккълох и Пит една крачка напред, като въвежда тегла към уравнението. Използвайки IBM 704, Розенблат успя да накара компютър да се научи как да различава картите, маркирани отляво, от картите, маркирани отдясно.

**1974:** Въпреки че много изследователи допринесоха за идеята за обратно разпространение, Пол Вербос беше първият човек в САЩ, който отбеляза приложението му в невронни мрежи в своята [докторска дисертация](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://www.researchgate.net/publication/35657389_Beyond_regression_new_tools_for_prediction_and_analysis_in_the_behavioral_sciences) (PDF, 8,1 MB) (връзката се намира извън ibm.com).

**1989:** Yann LeCun публикува [документ](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf) (PDF, 5,7 MB) (връзката се намира извън ibm.com), илюстриращ как използването на ограничения в обратното разпространение и интегрирането му в архитектурата на невронната мрежа може да се използва за обучение на алгоритми. Това изследване успешно използва невронна мрежа за разпознаване на ръкописни цифри на пощенския код, предоставени от пощенската служба на САЩ.

## Избор на библиотеки за разпознаване на реч в Python

В PyPI съществуват няколко библиотеки за разпознаване на реч. Някои от тях включват:

* apiai
* assemblyai
* google-cloud-speech
* pocketsphinx
* SpeechRecognition
* watson-developer-cloud
* wit

Някои от тези библиотеки като wit и apiai – предлагат вградени функции, като обработка на естествен език за идентифициране на намерението на говорещия, които надхвърлят основното разпознаване на реч. Други, като google-cloud-speech, се фокусират единствено върху преобразуването на реч в текст. Има една библиотека, който се откроява по отношение на лекотата на използване която е SpeechRecognition. Разпознаването на реч изисква въвеждане на аудио, а SpeechRecognition прави извличането на този вход наистина лесно. Вместо да се налага да създавате скриптове за достъп до микрофони и обработка на аудио файлове от нулата, SpeechRecognition ще ви накара да стартирате само за няколко минути. Библиотеката SpeechRecognition действа като обвивка за няколко популярни API за реч и по този начин е изключително гъвкава. Един от тях – Google Web Speech API – поддържа API ключ по подразбиране, който е твърдо кодиран в библиотеката SpeechRecognition. Гъвкавостта и лекотата на използване на пакета SpeechRecognition го правят отличен избор за всеки проект на Python.

Въпреки това поддръжката за всяка функция на всеки API, който обвива. И така, сега, след като сте убедени, че трябва да изпробвате SpeechRecognition, следващата стъпка е да го инсталирате във вашата среда.

SpeechRecognition е съвместим с Python 2.6, 2.7 и 3.3+, но изисква някои допълнителни стъпки за инсталиране за Python 2. В нашият случай ще го инсталираме на Python 3.11.1

Можем да инсталираме SpeechRecognition от терминал с pip:

pip install SpeechRecognition

SpeechRecognition прави работата с аудио файлове лесна благодарение на своя удобен AudioFile клас. Този клас може да се инициализира с пътя към аудио файл и предоставя интерфейс на контекстния мениджър за четене и работа със съдържанието на файла.

В момента SpeechRecognition поддържа следните файлови формати:

* WAV: трябва да бъде във формат PCM/LPCM
* AIFF
* AIFF-C
* FLAC: трябва да е оригинален FLAC формат; OGG-FLAC не се поддържа

Библиотеката SpeechRecognition в Python е инструмент, който може да се използва за извършване на разпознаване на реч с помощта на различни техники и машини. Той предоставя удобен интерфейс за разработчиците за интегриране на възможности за разпознаване на реч в техните Python приложения. В днешната ера на бързо напредващи технологии, разпознаването на реч се откроява като нещо обичайно във всеки аспект от живота ни. Разпознаването на реч е лингвистика в компютрите, която позволява на системите да разпознават и превеждат реч в текст. Включва областите на електротехниката, компютърните науки и лингвистиката. Нивото на автоматизация и логика обаче, което виждаме в днешните електронни устройства не би било възможно без концепцията за изкуствен интелект. Използват се гласови асистенти техники за разпознаване на глас, синтез на реч и обработка на естествен език (NLP). Картографирайте гласовите входове към командните изходи. Тези техники са подполе на изкуствените интелект, който позволява на компютърните системи да имат интелект, близък до човешкия (в този случай, по отношение на разбирането на нашата реч). Разпознаването на реч включва няколко стъпки, които включват анализ на звука вълнова форма, разделяне на изказвания чрез мълчание, разпознаване на думите във всяко изказване и произвеждайки крайния резултат под формата на текст (оттук и терминът реч към текст (STT)). За всяко изказване характеристиките се извличат като вектор на характеристиките. Аудио моделът се нарича Hidden Моделът на Марков (HMM) описва всеки последователен процес като речта. В речта разпознаване, акустичният модел, езиковият модел и фонетичният речник са трите използвани неща. В днешния свят разпознаването на реч и гласовите асистенти са навсякъде от персонализирани гласови записи в кол център, които водят обаждащите се през опциите до бизнеса използвайки имейли и документи, генерирани от говор, тази концепция нарасна много в миналото десетилетие. Гласовите асистенти могат да предоставят голямо разнообразие от услуги и по-специално тези от Amazon Alexa и Google Assistant растат с всеки изминал ден.

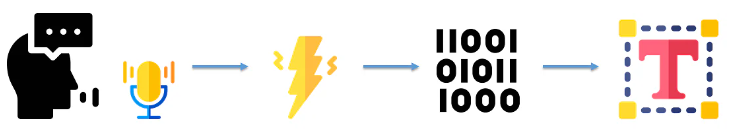
Те включват предоставяне на информация извлечени от интернет, възпроизвеждане на музика или видеоклипове и заместване на хора в областта на обслужване на клиенти и разговорна търговия. Те също са внедрени в интелигентен гласов асистент в автомобилният сектор за управление на функции и настройки от вътрешността на автомобила чрез гласови команди. Технологичните компании работят за създаването на все по-сложни технология, която ще автоматизира повечето процеси, които извършваме за един ден. За повечето от нас върховният лукс би бил асистент, който винаги ни изслушва вашето обаждане, предвижда всяка ваша нужда и предприема действия, когато е необходимо. Този лукс е сега достъпно благодарение на асистентите с изкуствен интелект, известни още като гласови асистенти. Те влизат донякъде малки пакети и може да извършва различни действия, след като чуе дума за събуждане или команда. Те могат да включват светлини, да отговарят на въпроси, да пускат музика, да правят онлайн поръчки, и т.н. Те също така откриха нарастваща съвместимост с IoT (Интернет на нещата), свързан устройства. Гласовите асистенти не трябва да се бъркат с виртуалните асистенти, които са хора които работят дистанционно и следователно могат да се справят с всякакви задачи. По-скоро гласови асистенти са базирани на технологии. Тъй като гласовите асистенти стават по-здрави, тяхната полезност както в личните и бизнес области също ще растат. Проектът, озаглавен „гласов асистент“, е самостоятелно приложение, което предоставя уникален потребителски интерфейс за изпълнение на ежедневни задачи в системата. Този софтуер може да помогне на потребителите да постигнат задачи, които обикновено биха направили сами. Причината за това е, че изговарянето на командата ще бъде по-лесно и по-бързо от въвеждане или щракване. То предлага потребителски интерфейс на естествен език за клиентите. Приложението може да приеме въвеждане в Интелигентен гласов асистент. Това може да се вземе от микрофона, който повечето устройства вече са вградени в тях. Получава се съответната команда и изпълнен. След като командата приключи изпълнението, се осигурява подходяща обратна връзка продължаване на процеса на разпознаване. Приложението е комбинация от Python и Електронни среди.



Как работи разпознаването на реч

Разпознаването на реч в Python работи с алгоритми , които извършват лингвистично и акустично моделиране. Акустичното моделиране се използва за разпознаване на фенони/фонетика в нашата реч, за да се получи по-значимата част от речта, като думи и изречения.

Фиг. Работа на разпознаването на реч



Говорене ел. енергия цифрови данни текст

Разпознаването на реч започва с вземане на звуковата енергия, произведена от човека, който говори, и преобразуването й в електрическа енергия с помощта на микрофон. След това преобразува тази електрическа енергия от аналогова в цифрова и накрая в текст. Той разделя аудио данните на звуци и анализира звуците с помощта на алгоритми, за да намери най-вероятната дума, която отговаря на това аудио. Всичко това се прави с помощта на обработка на естествен език и невронни мрежи. Налични са много приложения и устройства за разпознаване на реч, но по-модерните решения използват изкуствен интелект и машинно обучение. Те интегрират граматика, синтаксис, структура и състав на аудио и гласови сигнали, за да разберат и обработят човешката реч. В идеалния случай те се учат, докато вървят - развиващи се реакции с всяко взаимодействие. Най-добрият вид системи също така позволяват на организациите да персонализират и адаптират технологията към техните специфични изисквания - всичко от езика и нюансите на речта до разпознаването на марката. Например езиково претегляне, подобрете прецизността чрез претегляне на конкретни думи, които се изговарят често (като имена на продукти или индустриален жаргон), извън термините, които вече са в основния речник. Етикетиране на говорител: Изведете транскрипция, която цитира или маркира приноса на всеки говорител към разговор с множество участници. Обучение по акустика обърнете внимание на акустичната страна на бизнеса. Обучете системата да се адаптира към акустична среда (като околния шум в център за обаждания) и стилове на високоговорителите (като височина на гласа, сила и темпо). Филтриране на ругатни използвайте филтри, за да идентифицирате определени думи или фрази. Междувременно разпознаването на реч продължава да напредва. Компании, като IBM, навлизат в няколко области, за да подобрят взаимодействието между хората и машините.

Капризите на човешката реч направиха развитието предизвикателство. Смята се за една от най-сложните области на компютърните науки – включваща лингвистика и статистика. Устройствата за разпознаване на реч се състоят от няколко компонента, като въвеждане на реч, извличане на функции, вектори на характеристики, декодер и изход на дума. Декодерът използва акустични модели, речник на произношението и езикови модели, за да определи подходящия изход. Технологията за разпознаване на реч се оценява по нейния процент на точност, т.е. честота на грешка в думата и скорост. Редица фактори могат да повлияят на процента грешки в думите, като произношение, акцент, височина, сила на звука и фонов шум. Постигането на човешки паритет – което означава процент грешки, равен на този при говорене на двама души – отдавна е цел на системите за разпознаване на реч.

Използват се различни алгоритми и изчислителни техники за разпознаване на реч в текст и подобряване на точността на транскрипцията. По-долу са дадени кратки обяснения на някои от най-често използваните методи Обработка на естествен език (NLP) Въпреки че NLP не е непременно конкретен алгоритъм, използван за разпознаване на реч, това е областта на изкуствения интелект, която се фокусира върху взаимодействието между хора и машини чрез език чрез реч и текст. Много мобилни устройства включват разпознаване на реч в своите системи, за да извършват гласово търсене – например Siri – или да предоставят по-голяма достъпност при изпращане на текстови съобщения. Скрити модели на Марков (HMM)

Скритите модели на Марков се основават на верижния модел, който предвижда, че вероятността за дадено състояние зависи от текущото състояние, а не от предишните му състояния. Докато верижният модел на Марков е полезен за наблюдавани събития, като въвеждане на текст, скритите модели на Марков ни позволяват да включим скрити събития, като етикети за част от речта, във вероятностен модел. Те се използват, като модели на последователност в рамките на разпознаването на реч, присвоявайки етикети на всяка единица - т.е. думи, срички, изречения и т.н. - в последователността. Тези етикети създават картографиране с предоставения вход, което му позволява да определи най-подходящата последователност от етикети. N-грами-Това е най-простият тип езиков модел (LM), който присвоява вероятности на изречения или фрази. N-грама е последователност от N-думи. Например „поръчайте пицата“ 3 грама, а „моля, поръчайте пицата“ е 4 грама. Граматиката и вероятността от определени последователности от думи се използват за подобряване на разпознаването и точността. Невронни мрежи-Основно използвани за алгоритми за дълбоко обучение , невронните мрежи обработват данни за обучение, като имитират взаимосвързаността на човешкия мозък чрез слоеве от възли. Всеки възел се състои от входове, тегла, отклонение (или праг) и изход. Ако тази изходна стойност надвишава даден праг, тя „задейства“ или активира възела, предавайки данни на следващия слой в мрежата. Невронните мрежи научават тази функция за картографиране чрез контролирано обучение, коригирайки въз основа на функцията на загуба чрез процеса на градиентно спускане. Въпреки че невронните мрежи са склонни да бъдат по-точни и могат да приемат повече данни, това идва на цена за ефективност на производителността, тъй като те обикновено се обучават по-бавно в сравнение с традиционните езикови модели. Диаризация на говорещия (SD) Алгоритмите за диаризация на говорещия идентифицират и сегментират речта по идентичност на говорещия. Това помага на програмите да разграничават по-добре хората в разговор и често се прилага в центровете за обаждания, разграничавайки клиентите и търговските агенти.

Голям брой индустрии използват различни приложения на говорната технология днес, помагайки на бизнеса и потребителите да спестят време и дори животи. Някои примери включват:

Автомобили: Разпознавателите на реч подобряват безопасността на водача, като активират гласово активирани навигационни системи и възможности за търсене в автомобилните радиостанции.

Технология: Виртуалните агенти все повече се интегрират в нашето ежедневие, особено в нашите мобилни устройства. Използваме гласови команди за достъп до тях чрез нашите смартфони, като например чрез Google Assistant или Siri на Apple, за задачи, като например гласово търсене, или през нашите високоговорители, чрез Alexa на Amazon или Cortana на Microsoft, за възпроизвеждане на музика. Те само ще продължат да се интегрират в ежедневните продукти, които използваме, подхранвайки движението „Интернет на нещата“.

Продажби: Технологията за разпознаване на реч има няколко приложения в продажбите. Може да помогне на кол център да транскрибира хиляди телефонни обаждания между клиенти и агенти, за да идентифицира често срещани модели на обаждания и проблеми. Чат ботове с изкуствен интелект също могат да говорят с хора чрез уеб страница, като отговарят на често срещани запитвания и решават основни заявки, без да е необходимо да чакат служител от контактния център да бъде на разположение. И в двата случая системите за разпознаване на реч помагат да се намали времето за разрешаване на потребителски проблеми.

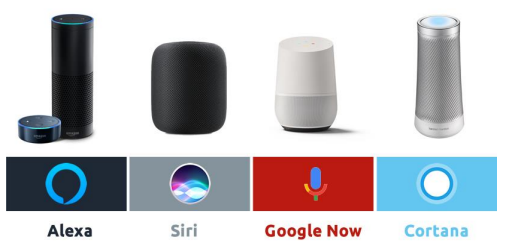
## Гласови асистенти

Разпознаването на реч е способността на машината да разпознава думи, изречени в някой език и превежда думите в машинно четим формат. Асистентът е софтуер, който изпълнява задачи на потребителя. Тези асистенти могат да работят с текст, реч или изображения. Гласовите асистенти извършват операции на устройства, базирани на гласова команда. Най-широко използваните гласови асистенти са Siri на Apple, GoogleAssistant, Alexa на Amazon и Cortana на Microsoft. До голяма степен инсталирани са в смартфони, настолни компютри и самостоятелни устройства, тези софтуерни агенти обикновено се интегрират с операционната система и предоставят средство за подобрена достъпност при изпълнение на всяка задача .В началото на 60-те години на миналия век Уилям К. Дерш от IBM създава „Кутия за обувки“, първата в историята реч машина за разпознаване, която можеше да разбере общо шестнадесет думи и да изпълнява математически изчисления, базирани на изречената командна фраза. Предшественикът на днешните системи с гласово управление, имаше хардуер, който се състоеше от микрофон, в който потребителят ще произнесе командата. Машината не беше като последните иновации и изискваше всяка дума да бъде ясно, ясно и бавно, с паузи между тях. Въпреки това, следващите 57 години донесе много подобрения в технологиите с навлизането на интернет и облачните изчисления, което доведе до разрастването на виртуалните асистенти, каквито ги познаваме днес. IBMS hoebox започна революция със своята технология.

## Гласови асистенти

С навлизането на интернет и изкуствения интелект, гласовите асистенти са едно цяло от най-често използваните софтуери днес. Иновации като Siri, Alexa и Cortana направят нашия личен и професионален живот по-лесен и по-ефективен. Над деветдесет милиони души само в САЩ използват гласови асистенти поне веднъж месечно. Според изследвания, около 20% от търсенията в Google се правят чрез гласова среда, като най предпочитан начин за комуникация между хора и устройства. Те са дигитализирали джаджи до такава степен, че да могат да намират упътвания, провеждат обаждания, резервират срещи и дори поръчват храна чрез проста гласова команда. Тези асистенти имат достъп до голяма колекция от онлайн и сървърни данни, до които се осъществява достъп бързо да интерпретира и изпълнява желаните задачи, според избора на потребителя. Проучване установи, че около 600 милиона души по света използват гласови приложения поне веднъж седмично под формата на Google Assistant, Amazon Alexa, Apple Siri или друг софтуер с гласово управление. С появата на интелигентните високоговорители и домакински уреди, гласовите асистенти се използват все повече. В началото търсачките се нуждаеха от изключително точни и кратки данни, за да извършване на основни търсения. С възхода на IoT и smart приложения, такива асистенти могат да взаимодействат не само с едно устройство, но и да автоматизират много уреди в къщата.

Фиг. 2 популярни гласови асистенти



Някои от функциите, които гласовият асистент може да изпълнява, включват отключване на устройства, отваряне на приложения, провеждане на повиквания, изпращане на съобщения, извличане на последните новини, заснемане изображения, възпроизвеждане на музика, задаване на напомняния или аларми, изпращане на имейл до колеги, изпълнение онлайн търсения, резервиране на билети и предлагане на препоръки за храна, развлечения и още. Основната причина за преминаването от традиционни системи към гласови потребителски интерфейси е поради променящите се потребителски изисквания. Постоянното подобряване и оптимизиране на бързината, точността, ефективността и удобството доведоха до необходимостта от гласово управление на системи. Друг основен фактор е растежът на изкуствения интелект във всяка фаза от живота ни с нарастващия брой IoT устройства като интелигентни хладилници, термостати, телевизори, високоговорители и микровълнови печки, животът на потребителите става все повече свързани и гласови асистенти помагат да се създаде тази връзка между всички уреди. В банковия сектор гласовата технология позволи на клиентите да проверяват своите балансират и плащат сметките си с помощта на гласовия асистент.

В рекламата, човешките ресурси и маркетинга, гласовите асистенти ще формират интерфейса между клиенти и компании, позволявайки по-бързо, по-ефективен анализ на данни и грижа за повтарящите се и транзакционни фази на индустрията. Дори в търговията на дребно гласовите асистенти дадоха на клиентите възможността да поръчат всеки продукт онлайн само чрез няколко прости команди. Това се отнася и за обществения транспортен сектор, където потребителите могат да запазят такси или да направят запитване за разписанията на полетите с помощта на гласовите асистенти. Всички индустрии вече са наситени с множество продукти които са се опитали да интегрират гласова технология, включително развлекателни сектори. Има много технически предизвикателства, пред които са изправени разработчиците, когато опит за създаване на гласов асистент. Първо, има отделни процеси за всяка стъпка от входа до изхода, вместо рационализиран, интегриран подход, който позволява на машините действително да разбират командите. Това означава, че въвеждането, разпознаването, обучението и декодирането фазите са отделни единици. По този начин машината всъщност не разбира команди, но просто преминава от една фаза към следващата, нанасяйки всеки вход на изход. Поради това интелигентността на машината е ограничена до определено ниво и не се изравнява с човешкото поведение. На второ място, има липса на съобразени с контекста отговори от машината. Машината не взема под внимание гласа на потребителите, интонациите, възприеманите емоции, ситуациите на околната среда или други контекстуални фактори това понякога може да доведе до отговори, които не отговарят на очакванията на потребителя. И накрая, има трудности при постигането на 100% точност поради безкрайните вариации в реч. Има безброй акценти и вариации в самият език, с речников запас достигайки до милиони. Тъй като клиентите очакват гласовите асистенти да разбират добре няколко езика, става трудно да се постигне 100% точност, поради сумата от данни и безкрайните възможности по отношение на вариации или произношения. Има няколко ключови фактора, които влизат в действие при обмислянето на създаването на интелигентен личен асистент. Те включват индивидуални и обширни разработки които обхващат следните полета: реч към текст, текст към реч, шум управление, компресиране на реч, гласова биометрия и гласов потребителски интерфейс. Всеки от тези областите представляват важна част от развитието на гласовия асистент и трябва да бъдат подготвени ефективно, като се отдава значение на тяхната точност и бързина. В бъдеще гласовите асистенти ще стават все по-сложни. Някои от подобренията, които очакваме, ще бъдат персонализирани и съобразени с контекста. Тъй като пазарът за това технологията непрекъснато се разраства, това затруднява разработчиците да създават продукти, които се открояват от съществуващите системи. Има няколко платформи за разработчици с отворен код които са пуснати от много компании, които позволяват на програмистите да създават и персонализират своя собствен виртуален асистент. Тези независими услуги съдържат модулите за разпознаване на реч, библиотеките за гласови проби, алгоритмите за машина обучение и средствата за преобразуване и картографиране на текст в изпълними команди.

## Защо избирам Python за разработка на гласовия асистент

Какво е Python? Python е език за програмиране, използван за писане на програми за всичко - от анализ на данни до вграждане на [отчети с бели етикети](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://whatagraph.com/white-label) на вашия уебсайт. Това е език с общо предназначение, което означава, че може да се използва в различни програми и за решаване на много проблеми.

Ето само няколко от възможните приложения на Python:

* Анализ на данни
* Машинно обучение
* Уеб разработка
* Автоматизация
* Тестване на софтуер

Python е невероятно популярен език за програмиране, който се използва все повече през последните години, изпреварвайки Java като най-използвания език за програмиране. Това се дължи главно на неговата гъвкавост и колко лесен е за използване както от програмисти, така и от хора, които не кодират. Има много причини, поради които Python се е превърнал в най-широко използвания език за програмиране и защо е страхотен за изграждане гласов асистент.

Лесен за научаване Python има прост синтаксис, подобен на английски, който имитира естествения език, което го прави лесен за четене, разбиране и писане. Той изисква по-малко редове код за изпълнение на същата задача в сравнение с други езици за програмиране като Java. Това е изключително важно, тъй като начинът, по който работим онлайн, се променя. Тъй като практики като [изчислителни облаци стават все по-широко използвани](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://www.g2.com/articles/cloud-computing-trends-2022) , от решаващо значение е бързо да се адаптирате към променящата се работна среда, за да избегнете изоставане от вашите конкуренти.

Python има широк набор от библиотеки и модули, до които имаме достъп. Това са пакети от код, разработен от потребители на трети страни за различни задачи. Каквато и функция да ви е необходима, за да изпълните задачата си, има голяма вероятност тя да е разработена и готова и чака да я използвате. Много езици за програмиране изискват да промените кода си, за да стартирате програма на различни платформи. Python обаче не го прави. Трябва да напишете кода само веднъж и след това той може да се изпълнява практически навсякъде, стига да не сте включили никакви зависещи от системата функции. Това е особено полезно, ако наемате голям брой [служители, които работят дистанционно](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://blog.airdroid.com/post/5-management-tips-that-make-or-break-remote-it-support-teams/) и може да използват различни устройства и операционни системи. Python е невероятно гъвкав. Изпълнява много различни задачи. Това означава, че трябва да овладеете само един език за програмиране, за да автоматизирате всички задачи, които искате, вместо да изучавате конкретни езици за различни задачи.

Тъй като е одобрен от OSI лиценз за отворен код, Python е свободен за използване и разпространение. Можете да изтеглите изходния код, да го промените и след това да разпространите вашата версия, ако желаете. Това позволява на вашата организация лесно да разработва програмите, от които се нуждаете, без да се притеснявате да плащате за скъпи лицензи.

Python е широко използван, което означава, че около него има голяма и активна общност. Това означава, че ако срещнете проблеми, когато се опитвате да го използвате, намирането на решение е относително лесно, защото някой друг в общността вероятно е имал същия проблем и може да ви помогне да го преодолеете. Точно както [високоплатените партньорски програми](https://translate.google.com/website?sl=en&tl=bg&hl=bg&client=webapp&u=https://affise.com/blog/best-affiliate-programs/) могат да осигурят достъп до вашия продукт на по-широк кръг потребители, общността на Python може да осигури достъп до по-широк набор от изходни кодове на трети страни и канали за поддръжка.

Тъй като е динамично въведен, няма декларации, което го прави компактно и сбито. Най-важната част от Python е отстъпът, тъй като той определя поток от изявления. Python също така съдържа изкуствен интелект и обработка на естествен език библиотеки, което го прави полезен в тези области. Използва се и в информационната сигурност и като основен език за Raspberry Pi. Въпреки това, в сравнение с C/C++, Python е малко по-бавен и не поддържа браузъри и мобилни устройства. Python се инсталира в много компютри с Linux и Windows. Ако вашата система няма Python, инсталационната програма може да бъде изтеглена от страницата за изтегляне на уебсайт python.org. Стартиране на инсталатора и избиране на опцията за добавяне към променлива PATH ще настрои Python файловете. Има две версии на Python: 2.x и 3.x. Малко са незначителни разлики между тях като функция за печат, обработка на грешки и оператор за разделяне. Има също много начини за изпълнение на код в Python. Един от начините е да напишете „python“ командния ред под инсталираната папка, която ще отвори обвивката на Python. поредица от твърдения. Друг начин е да използвате IDE (интегриран софтуер развитие) като PyCharms. В Python има няколко важни библиотеки като:

• Пакетът “Numpy” за обработка на масиви за научни изчисления на многомерни масиви

• Библиотеката “keras”, която е пакет за невронни мрежи с отворен код

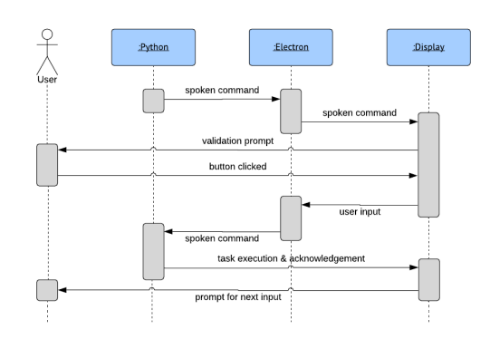
• “Pandas DataFrame”, която е двуизмерна таблична структура от данни с две оси. Освен това има множество методи за внедряване на GUI (графичен потребителски интерфейс). от които „tkinter“ е един от най-популярните. Също така има и модули за математически функции, календар, време, автоматизация и обработка на URL адреси. Следват някои от предимствата на Python.

Анализ на изискванията

Изискванията, които определят какви услуги една система може да предостави на крайния потребител се наричат функционални изисквания. Функционалните изисквания са тясно свързани с изискванията на потребителските спецификации. Това може да включва изчисления, обработка на данни, технически операции и друга такава функционалност, която има за цел да изпълни целите на приложението. Всички операции по проследяване, законови изисквания, подробности за интерфейса, нива на оторизация, актуализации на транзакции и административни функции отговарят на функционални изисквания. Техническата архитектура на системата е определени от тези изисквания.

Ефективността се оценява чрез следните спецификации:

• Време за реакция: Това е времето, необходимо на системата да приеме въвеждане от потребителя и да отговори към него чрез показване на някакъв резултат. Обикновено съобщенията за обратна връзка се показват в рамките интелигентен гласов асистент. Максимум 10 секунди на диалог прозорец гарантира, че потребителят няма да загуби интерес или ход на мисълта. Отговорът времето също трябва да е последователно и да не варира в зависимост от броя на едновременните сесии.

• Работно натоварване: Това е количеството стрес или работа, което системата може да издържи наведнъж. Това може да бъде по отношение на паралелни сесии, брой активни потребители или брой транзакции с бази данни. Работното натоварване обикновено се описва като сценарии, които потребителите най-вероятно ще срещнат. Специални случаи като сценарии за грешка, архивиране и исканията за управление трябва да се вземат предвид при уточняването на натовареност.

Процесът на проектиране на системната архитектура се фокусира върху разбиването на система в различни компоненти и техните взаимодействия, които удовлетворяват функционални и нефункционални изисквания. Входовете за проектиране на софтуерна архитектура са изискванията и хардуерната архитектура. Този проект няма периферия хардуерни устройства, така че софтуерните компоненти взаимодействат само със системата микрофон и дисплей.

Въвеждане

Слушане на изречени команди

микрофон

Разпознаване на език и превръщане в текст

Валидиране на потребителя

монитор

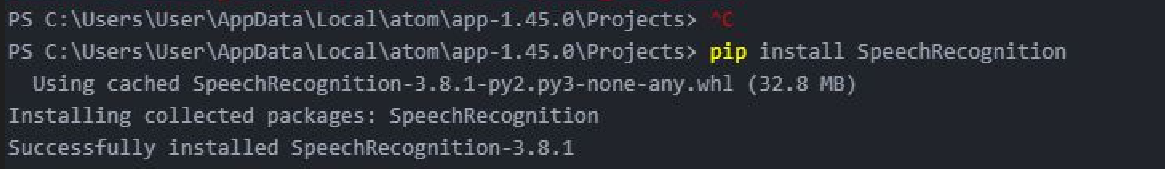
Потребител

Изпълняване на задачата

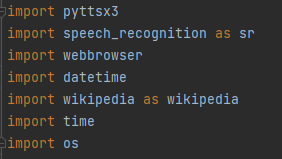
Диаграмата на системната за последователност показва събитията, които са генерирани от външни потребители в последователен ред, за конкретен сценарий на случай на употреба. Те са начин за визуално обобщаване на това, което се случва в един единствен случай на употреба.

# Разработка

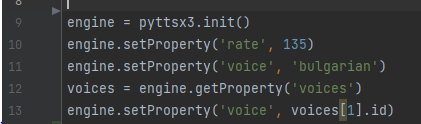
Първи стъпки от нашата практическа разработка е инсталиране на среда за програмиране, аз съм се спрял на PyCharm. Отварям средата за програмиране създавам нов проект. Инсталираме необходимите библиотеки с помощта на pip в терминала пишем pip install SpeechRecognition , pyttsx3,webbrowser ,datetime, Wikipedia, time и os, а ако имаме проблем с някоя от библиотеките си отваряме pypi.org и си четем как точно се инсталира, защото може да има някой особености.



След, като съм приключил с инсталирането от терминала създавам нов Python файл и в него импортирам всички необходими библиотеки

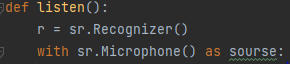


Сега започвам с библиотеката pyttsx3, като задам скорост на възпроизвеждане и мога да си избираме между мъжки и женски глас, като 1 е за женски, а 0 за мъжки. В бъдеще може да обучим собствения глас на базата на вашия собствен глас или на някой друг и след това да го използвам.



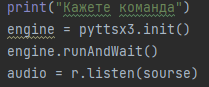
Python предоставя няколко библиотеки за разпознаване на реч аз съм се спрял на SpeechRecognition-тази библиотека предоставя прост интерфейс за извършване на разпознаване на реч с помощта на различни API, като например Google Speech Recognition, Microsoft Bing Voice Recognition и CMU Sphinx.

Дефинираме функция с име Listen, която ще ни взима аудиото от микрофона

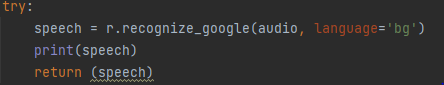


В кода създаваме разпознаващ обект r. След това използваме оператора with sr.Microphone() as source:, за да зададем микрофона като аудио източник.

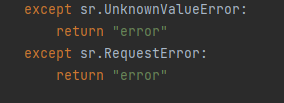
След това подканваме потребителя да каже нещо и използваме метода r.listen(source), за да запишем аудио данните от микрофона. Този метод блокира, докато не открие края на речта на потребителя.



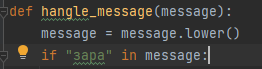
Накрая извикваме метода r.recognize\_google(audio\_data), за да транскрибираме аудио данните и да ги конвертираме в текст. След това текстът се отпечатва на конзолата.



Функцията се опитва да транскрибира речта с помощта на метода accept\_google() и връща транскрибирания текст като низ. Въпреки това, ако речта не може да бъде разпозната поради грешка с неизвестна стойност или грешка при заявка, функцията връща низа „грешка“. Важно е да се отбележи, че без пълния контекст и изпълнение на функцията е трудно да се осигури пълно разбиране на нейната функционалност или потенциални проблеми, които могат да възникнат.

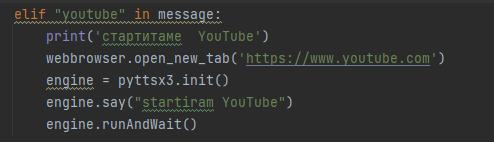


Създаваме нова функция наречена „handle\_message“ функцията приема параметър, наречен "message". След това функцията проверява дали низът „Зара“ присъства в изречените думи от потребителя, ако присъства ключовата дума “Зара“ продължава да претърсва надолу написания код, а ако не чуе ключовата дума излиза от функцията.



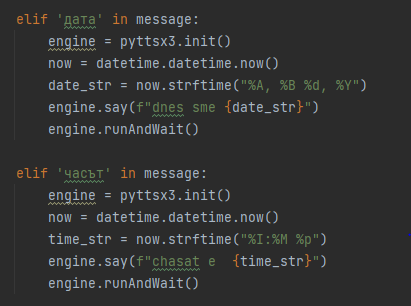
Ще направим и едно условие, ако условието „чао“ е вярно излиза от програмата с помощта на exit()

Библиотеката на уеб браузъра в Python предоставя прост интерфейс за отваряне на уеб страници в уеб браузъра по подразбиране на системата на потребителя. Той поддържа отваряне на уеб страници в нов раздел, нов прозорец или в същия прозорец, в зависимост от настройките на браузъра по подразбиране на потребителя. За да използвате библиотеката на уеб браузъра, първо трябва да я импортирате във вашата програма на Python. След като библиотеката бъде импортирана, можете да използвате функцията webbrowser.open(), за да отворите URL адрес в уеб браузъра по подразбиране на потребителя. Например следният код ще търси ключова дума YouTube, ако намери думата, която сме задали ще отвори в уеб браузъра по подразбиране на потребителя YouTube.



Функцията webbrowser.open() ще отвори URL адреса в нов раздел. Ако искате да отворите URL адреса в нов прозорец, можете вместо това да използвате функцията webbrowser.open\_new(). За да отворите URL адреса в същия прозорец, можете да използвате функцията webbrowser.open\_new\_tab().

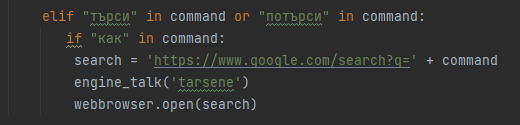
Няма как да правим гласов асистент без да може да ни казва дата и час. Ще използваме библиотеката pyttsx3 за получава текущата дата и час с помощта на функцията datetime.datetime.now(). След това датата се форматира като низ с помощта на метода strftime() със специфичен форматиращ низ "%A, %B %d, %Y", който указва деня от седмицата, името на месеца, деня от месеца, и годината. След това кодът използва f-низове, за да конструира гласов отговор при часа е същото само че заместваме един ред с този код: ‘’ time\_str = now.strftime("%I:%M %p")’’, a накрая методът runAndWait() се извиква, за да възпроизведе аудио.



Ще направя и още една интересна функция към моят гласов асистент, а именно да може да се изключва когато му кажеш за целта ще използваме библиотеката time.

Задавам променлива time\_left на 5 и след това влиза в цикъл while, който работи, докато time\_left е по-голямо от 0. Вътре в цикъла кодът отпечатва съобщение, което показва колко секунди остават до изключване, намалява time\_left с 1 и след това изчаква 1 секунда с помощта на функцията time.sleep().След като цикълът приключи, кодът използва функцията os.system(), за да изпълни системна команда, която изключва компютъра. По-конкретно, командата shutdown /s /t 1 казва на операционната система да инициира незабавно изключване (/t 1) и да извърши пълно изключване на системата.

Накрая проверяваме дали кодът се изпълнява if \_\_name\_\_ == '\_\_main\_\_' Кодовият блок под този оператор ще се изпълни само ако скриптът се изпълнява като основна програма. След това кодът отпечатва низа "test" на конзолата. След това има цикъл while True, което означава, че цикълът ще работи за неопределено време, докато програмата бъде прекъсната или спряна. В рамките на този цикъл се извиква функцията listen(), за да извлече команда, а след това се извиква функцията handle\_message(command), за да обработи командата. 

Крайният резултат от този проект е гласов асистент, както е описано в заглавието. Той съчетава техники за обработка на естествен, за да представи ефективен личен асистент. Може да изпълнява ежедневни задачи в системата въз основа на устната команда на потребителя. Може да разпознава думите, да картографира речта в текст и да реши каква задача да изпълни съответно. Приложението може да изпълнява операции на устройството като отворени приложения като настройки, калкулатор, Microsoft Word, мултимедийни плейъри и т.н. 

И да може да направим зара още по умна ще напишем код, който да търси изречените от нас команди в Google. Този код проверява дали низът "как"присъства в променливата "команда". Ако е така, кодът създава URL адрес за търсене с Google с командата като заявка за търсене и го отваря в уеб браузър с помощта на библиотеката за уеб браузър. Функцията engine\_talk('tarsene') ни изписва в конзолата, че функцията се изпълнява и търси.

# Заключение

Тъй като разходите за хардуер, софтуер и персонал за ИИ могат да бъдат скъпи, много доставчици включват ИИ компоненти в своите стандартни предложения или предоставят достъп до платформи за изкуствен интелект като услуга. AIaaS позволява на физически лица и компании да експериментират с ИИ за различни бизнес цели и да пробват множество платформи, преди да поемат ангажимент.

Невронни мрежи Или като дете, те се раждат без да знаят много и чрез излагане на житейски опит, те бавно се научават да решават проблеми в света. За невронните мрежи данните са единственото преживяване.)

Бъдещо подобрение

В бъдещите версии на този проект някои атрибути, които могат да бъдат добавени, са:

* + Възможност за разширен разговор (като чатбот)
  + Функции за постоянно съхранение на данни
  + Персонализиране за различни потребители
  + Гласова биометрия и сигурност
  + Лесен и удобен интерфейс
* Персонализиране: Позволете на потребителите да персонализират своето изживяване с гласов асистент, като зададат предпочитания от тях музикален жанр, източници на новини или събития в календара.
* Интегриране с интелигентни домашни устройства: Позволете на гласовия асистент да управлява интелигентни домашни устройства като осветление, термостат и системи за сигурност.
* Многоезична поддръжка: Позволете на потребителите да комуникират с гласовия асистент на множество езици.
* Непрекъснато обучение: Позволете на гласовия асистент да се учи от предишни взаимодействия и да подобрява своите отговори с течение на времето. Разпознаване на контекста: Добавете разпознаване на контекста към гласовия асистент, така че да може да отговаря на запитвания въз основа на местоположението на потребителя, времето от деня и друга подходяща информация.
* Добавяне повече функционалност към гласовия асистент, като например възможност да резервирате полети, да правите резервации или да поръчвате храна.

# Използвана литература

Невронни мрежи - <https://wiki.pathmind.com/neural-network>

<https://studypython.site/?gclid=Cj0KCQjwla-hBhD7ARIsAM9tQKtcIeQAkUKiDch71lR-dXRiQsoSBaaGizuC0UY-x4fdc0MfxjhjKYQaAtC3EALw_wcB>

<https://pyttsx3.readthedocs.io/en/latest/engine.html>

<https://pypi.org>

<https://pypi.org/project/SpeechRecognition/>

https://www-ibm-com.translate.goog/topics/neural-networks?\_x\_tr\_sl=en&\_x\_tr\_tl=bg&\_x\_tr\_hl=bg&\_x\_tr\_pto=wapp