**УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ**

**ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ**

**Никола Карпић**

**Развој рјешења за предвиђање броја особа у просторији**

**дипломски рад**

**Бања Лука, новембар 2023.**

**Тема: РАЗВОЈ РЈЕШЕЊА ЗА ПРЕДВИЂАЊЕ БРОЈА ОСОБА У ПРОСТОРИЈИ**

**Кључне ријечи:**

**Машинско учење**

**Регресија**

**Класификација**

**LightGBM**

**Комисија:**

**проф. др Милош Љубојевић, предсједник**

**проф. др Зоран Ђурић, ментор**

**Александар Келеч, ма, члан**

**Уз рад је приложен CD.**

**Кандидат:**

**Никола Карпић**

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ

ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

КАТЕДРА ЗА РАЧУНАРСТВО И ИНФОРМАТИКУ

Тема: РАЗВОЈ РЈЕШЕЊА ЗА ПРЕДВИЂАЊЕ БРОЈА ОСОБА У ПРОСТОРИЈИ

Задатак: Машинско учење. Описати класификационе и регресионе алгоритме машинског учења и њихове типичне представнике. Припрема скупова података за тренирање, валидацију и тестирање. У практичном дијелу рада анализирати моделе за предвиђања броја особа у просторији креиране кориштењем неколико алгоритама (класификационих и регресионих). Моделе је потребно тренирати подацима који садрже температуру, влажност, ниво угљен-диоксида и др. За реализацију користити *Jupyter Notebook* алат. Извршити компаративну анализу перформаси добијених модела.

Ментор: проф. др Зоран Ђурић

Кандидат: Никола Карпић (1144/14)

Бања Лука, новембар 2023.

Садржај

[1. Увод 1](#_Toc148897954)

[2. Машинско учење 4](#_Toc148897955)

[2.1. Надгледано учење 7](#_Toc148897956)

[2.1.1. Класификација 9](#_Toc148897957)

[2.1.2. Регресија 10](#_Toc148897958)

[2.2. Ненадгледано учење 11](#_Toc148897959)

[2.2.1. Груписање 13](#_Toc148897960)

[2.2.2. Смањење димензионалности 14](#_Toc148897961)

[2.2.3. Учење уз подстицај 14](#_Toc148897962)

[3. Процес машинског учења 17](#_Toc148897963)

[3.1. Дефинисање проблема 17](#_Toc148897964)

[3.2. Прикупљање података 18](#_Toc148897965)

[3.3. Припрема података 18](#_Toc148897966)

[3.4. Избор модела 19](#_Toc148897967)

[3.5. Тренирање модела 19](#_Toc148897968)

[3.6. Оцјењивање модела 20](#_Toc148897969)

[3.7. Прилагођавање модела 20](#_Toc148897970)

[3.8. Тестирање и примјена модела 21](#_Toc148897971)

[4. Избор алгоритма 22](#_Toc148897972)

[4.1. Логистичка регресија 22](#_Toc148897973)

[4.2. Gaussian Naive Byes Classifier 23](#_Toc148897974)

[4.3. K Nearest Neighbors Classifier 23](#_Toc148897975)

[4.4. Decision Tree Classifier 23](#_Toc148897976)

[4.5. Random Forest Classifier 24](#_Toc148897977)

[4.6. Gradient Boosting Classifier 24](#_Toc148897978)

[4.7. Support Vector Machine Classifier 25](#_Toc148897979)

[4.8. LightGBM Classifier 25](#_Toc148897980)

[4.9. K Nearest Neighbors Regressor 26](#_Toc148897981)

[4.10. LightGBM Regressor 26](#_Toc148897982)

[5. Практични рад 27](#_Toc148897983)

[5.1. Помоћне функције 27](#_Toc148897984)

[5.2. Прикупљање података 29](#_Toc148897985)

[5.3. Припрема података 31](#_Toc148897986)

[5.4. Избор и тренирање модела 35](#_Toc148897987)

[5.5. Оцјењивање модела 36](#_Toc148897988)

[6. Резутати 38](#_Toc148897989)

[7. Закључак 39](#_Toc148897990)

[8. Литература 40](#_Toc148897991)

# Увод

У данашње вријеме се суочавамо са феноменом све веће производње дигиталних података у разним форматима. Количина података се експоненцијално повећава и ти подаци су огроман ресурс који остаје неискориштен. Због наглог повећања количине података, појавили су се могућност и потреба за обрадом тих података. Како је велики проценат тих података неструктурисан и некласификован, појавила се потреба и за креирањем ефикасних алгоритама и процеса за разврставање, именовање и анализу тих података. Једна од најзаступљенијих области вјештачке интелигенције (*Artificial Intelligence*) која се бави разврставањем, именовањем и анализом података је машинско учење (*Machine learning*).

Велике компаније као што су Google[[1]](#footnote-1), Microsoft[[2]](#footnote-2), Facebook[[3]](#footnote-3) и Amazon[[4]](#footnote-4) су почеле да улажу огромне напоре и средства у своје могућности чувања, обраде и класификације великих количина несређених података како би остале релевантне и оствариле тржишну предност у данашњем брзорастућем и брзомијењајућем дигиталном пространству. Велика количина средстава која се улажу у ову област је довела до експлозије научних радова на тему машинског учења у посљедњих 15 година (слика 1.1).

Владе, војске, Министарства унутрашњих послова и остале институције разних држава улажу у ову област, такође, било због могућности да боље и ефикасније прате своје грађане и брже проналазе могуће терористе и преступнике, било због постизања предности над другим државама у војном или обавјештајном сектору.

Машинско учење је облик вјештачке интелигенције који омогућава систему да учи из података, а не путем експлицитног програмирања. Машинско учење користи низ алгоритама који итеративно уче из података да би побољшали, описали податке и предвидјели исходе. Како алгоритми уносе више података за учење, тако је могуће произвести све прецизније моделе засноване на тим подацима. [1]



*Слика 1.1. Број научних радова на тему машинског учења у свијету од 2000. године[[5]](#footnote-5)*

Алгоритми машинског учења се убрзано увлаче у све сфере живота данашњих људи. Све је чешће да породице посједују дигиталног асистента који се контролише звуком или помоћу паметног телефона. Све су чешћи паметни кућански уређаји и имплементације концепата као што су паметне куће са разним сензорима (камере, микрофони, детектори пожара, нивоа влаге, угљен-диоксида, освијетљености, итд...) у свим просторијама.

Компаније често користе податке о броју особа у просторији, како би направиле уштеде на трошковима гријања и како би смањиле свој угљенични отисак. Смањењем угљеничног отиска компаније повећавају своју прихватљивост у очима све више еколошки освијештених потенцијалних нових клијената и генеришу милијарде долара годишње продаје. [2] Ти сензори могу да представљају проблем за осјећај приватности у дому или на радном мјесту.



*Слика 1.2. Паметни усисивач LUCY који посједује огроман број сензора (чак и камеру)*

Из наведених разлога се тежи да се што више корисних података добија из сензора који не нарушавају приватност директно као што то раде микрофони и камере, па се прибјегава техникама које индиректно процјењују тражене величине из података које нам дају сензори који нису толико инвазивни у погледу приватности. Још једна брига је и очување пословних тајни у компанијама, пошто се помоћу микрофона и камера, које се могу користити за процјену броја особа у просторијама, релативно једноставно може доћи до пословних тајни изговорених на затвореним састанцима, ако постоје и најмањи сигурносни пропусти у сигурносном систему компаније.



*Слика 1.3. Хакер долази до пословних тајни без уласка у просторије компаније-жртве*

У овом раду ће се обрађивати примјењивање одређеног броја класификационих и регресионих машинских алгоритама за процјену броја особа у просторији на основу датума, времена и података прикупљених сензорима за ниво угљен-диоксида у просторији, освијетљеност просторије, количину влаге у просторији и сензорима за количину кретања у просторији. Алгоритми машинског учења које ћемо користити су: логистичка регресија, *Gaussian Naive Byes, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine, LightGBM*.

У другој глави се даје објашњење вјештачке интелигенције и уопштено објашњење метода машинског учења и процеса примјене алгоритама машинског учења.

У трећој глави су детаљно објашњени кораци креирања модела машинског учења помоћу алгоритама машинског учења.

У четвртој глави су детаљније описани алгоритми који су кориштени у овом раду.

У петој глави се налазе појединости које се односе на практични дио. У њој су детаљно описани подаци и примјена сваког алгоритма, те упоредна анализе резултата. Сав код је написан у програмском језику Python.

У седмој глави се налази опис упоредне анализе резултата и коначан одабир најбољег алгоритма.

У шестој глави се налази закључак који је добијен у овом раду.

# Машинско учење

Вјештачка интелигенција је једно од најновијих поља у науци и инжењерству. Ова област је почела озбиљно да се развија убрзо након Другог свјетског рата, а само име је настало 1956. Заједно са молекуларном биологијом, вјештачка интелигенција се редовно наводи као „поље у којем бих највише волио да будем“ од научника из других дисциплина.

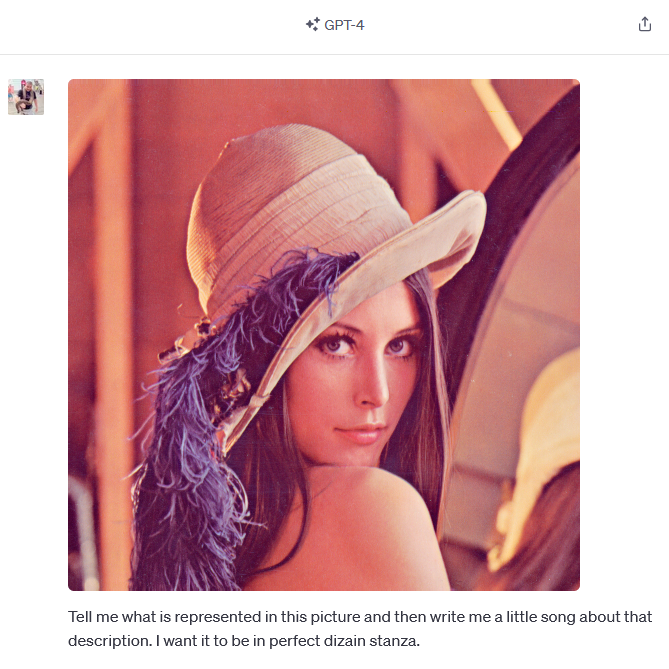
Студент физике може оправдано да сматра да су све добре идеје већ преузели Галилео, Њутн, Ајнштајн и остали. Вјештачка интелигенција, с друге стране, још увек има мјеста за неколико нових Ајнштајна и Едисона. Вјештачка интелигенција тренутно обухвата огроман број области, у распону од општег (учење и перцепција) до специфичних, као што су играње шаха, доказивање математичких теорема, писање поезије, вожња аутомобила у препуној улици и дијагностиковање болести. Вјештачка интелигенција је релевантна за било који интелектуални задатак; то је заиста универзално поље. [3]

До нагле популаризације вјештачке интелигенције и појачаног интересовања за њене могуће примјене у физичком свијету, довели су пројекти човјеколиког робота (Atlas) и роботског пса (Spot) америчке компаније *Boston Dynamic*.

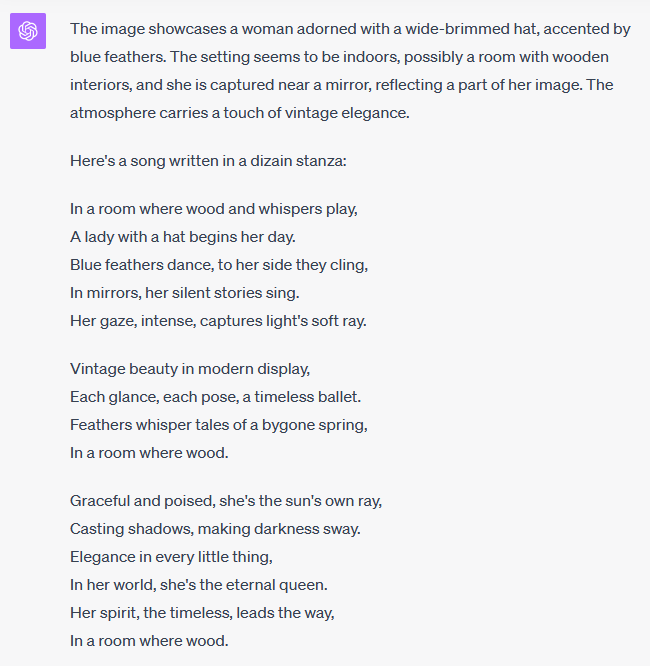


*Слика 2.1. Роботски пас и човјеколики робот компаније Boston Dynamics[[6]](#footnote-6)*

У последње време, *ChatGPT* компаније *Open AI* је довео до наглог повећања интересовања шире јавности за вјештачку интелигенцију. Међутим, ово повећање интересовања истовремено отвара и питање ограничавања развоја вјештачке интелигенције. Људи постају свјесни важности етичких, правних и друштвених аспеката ове технологије и постаје све битније потражити одговоре на питања о томе како осигурати да вјештачка интелигенција буде развијена на начин који користи цјелокупном човечанству.



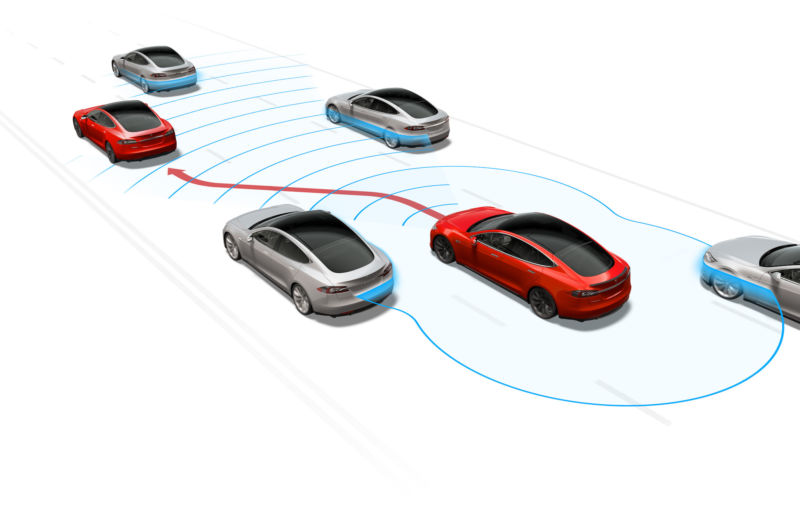
*Слика 2.2. Захтјев упућен систему ChatGPT компаније Open AI*



*Слика 2.3. ChatGPT препознаје фотографију, описује је и пише пјесму о њој[[7]](#footnote-7) [[8]](#footnote-8)*

Као што је већ речено у уводу, машинско учење је облик вјештачке интелигенције који омогућава систему да учи из података, а не путем експлицитног програмирања. Међутим, машинско учење није једноставан процес. Машинско учење користи низ алгоритама који итеративно уче из података да би побољшали, описали податке и предвидјели исходе. Како алгоритми уносе више података за учење, тако је могуће произвести све прецизније моделе засноване на тим подацима. [1] Алгоритми машинског учења као улаз примају огромне скупове података који описују одређене појаве и обрађују их и у њима проналазе правилности, а као излаз стварају моделе који помоћу пронађених правилности могу да предвиђају резултате тих појава над невиђеним улазним подацима.

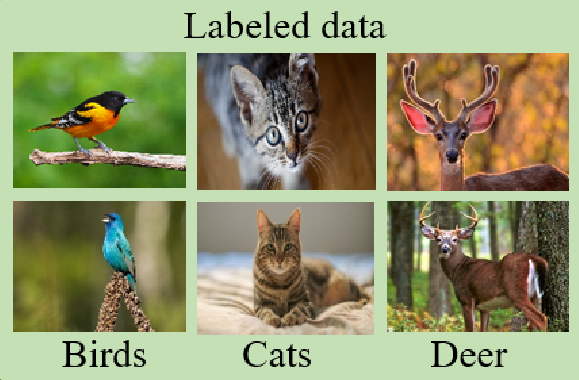
Неки од типичних примјера у којима се машинско учење показало као одговарајући алат су аутономна возила, препоручена претрага код интернетских претраживача, добијање повратние информације о томе шта купци мисле о нама, препознавање нежељених порука, откривање превара. Технике машинског учења се обично дијеле на три области у зависности од врсте повратне информације доступне систему учења, а то су: надгледано учење, ненадгледано учење и учење уз подстицај.



*Слика 2.4. Приказ аутономне промјене возне траке возила компаније Tesla Motors [[9]](#footnote-9)*

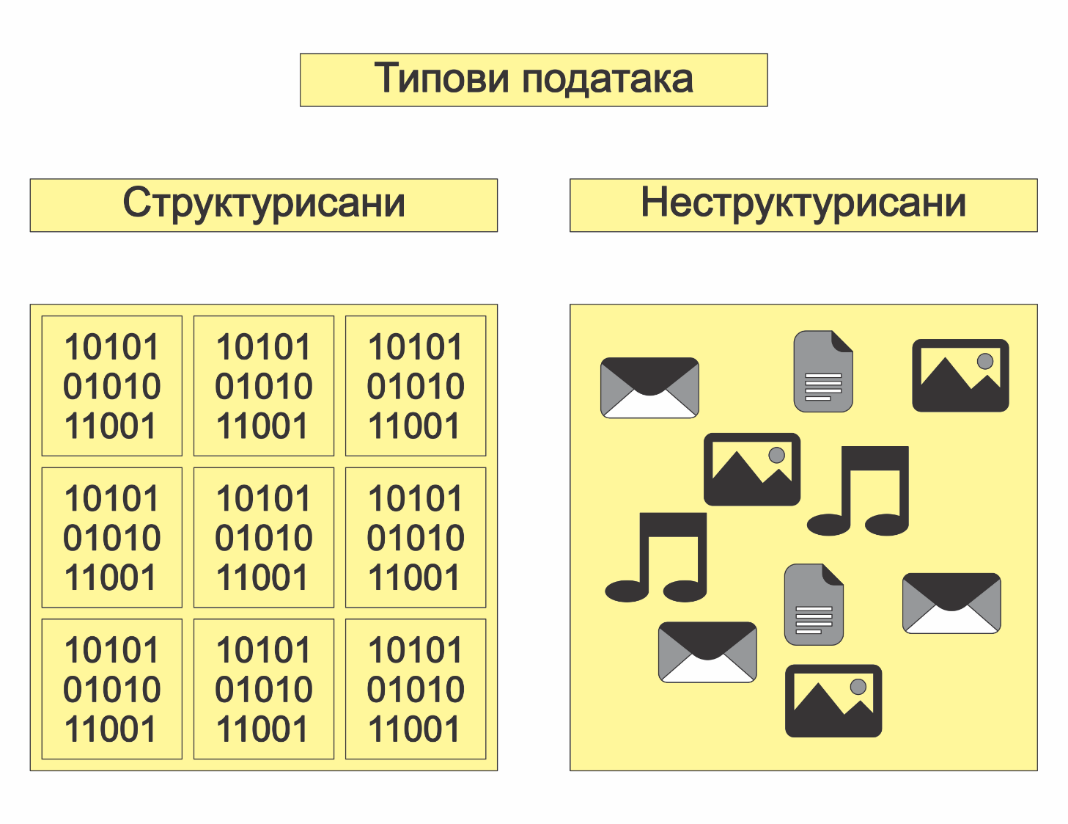
## Надгледано учење

Надгледано учење *(енг. Supervised Learning)* је врста машинског учења у коме алгоритам проучава функцију која пресликава улазне податке на излазне како би научио функцију којом може да предвиди излазе за нове, непознате улазне податке. [3] Та функција се назива моделом. Алгоритам учења добија одређен скуп парова улаз-излаз као скуп података за учење и даје функцију предвиђања за све могуће улазе. Укупан број ставки и природа улазних података може утицати на одабир најбољег алгоритма за специфичан проблем.



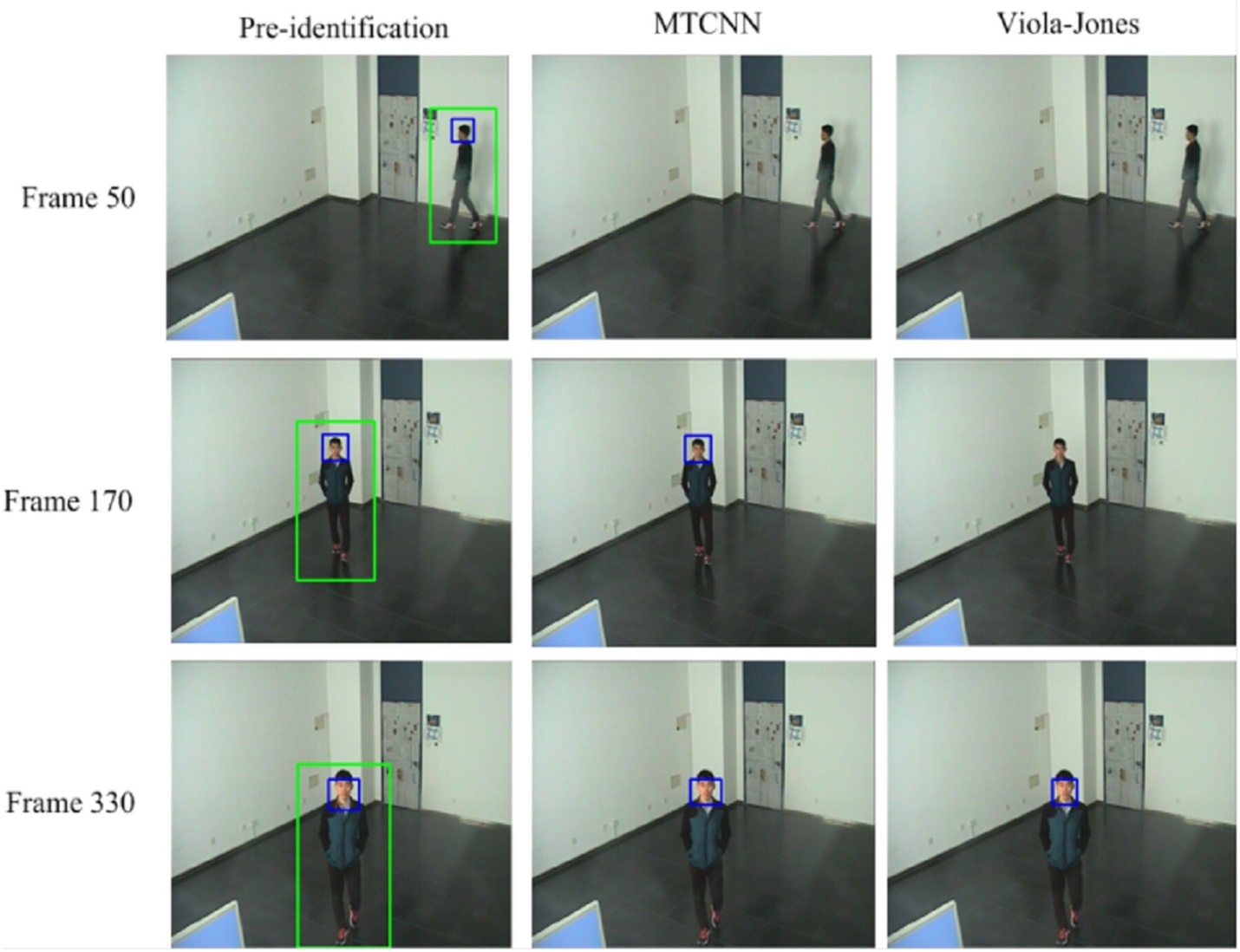
*Слика 2.5. Примјер означеног скупа података*

Пошто улаз модела чине подаци, они се могу јавити у структурисаном, неструктурисаном и полуструктурисаном облику. Структурисани подаци су сви подаци који се могу интерпретирати у неком фиксном формату (физичка мјерења, вријеме, бројеви телефона, број особа, спол), док су неструктурисани подаци они подаци за које не постоји фиксна структура (разне текстуалне датотеке, фотографије, видео записи, звучни записи, веб странице, резултати претраживања на вебу).



*Слика 2.6. Примјер означеног скупа података*

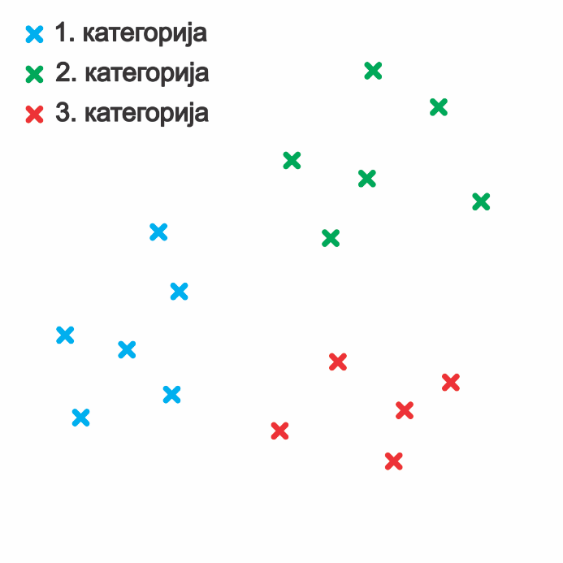
Надгледано учење је најчешћа врста машинског учења која се користи код проблема класификације, регресије и рангирања. Проблем откривања нежељене поште је један од најпознатијих примјера код којих се користи надгледано учење. [4] Надгледано машинско учење се често користи и у апликацијама попут кредитне процјене, медицинске дијагностике, временске прогнозе и персонализације реклама. [5]



*Слика 2.7. Препознавање лица у реалном времену* [6]

### Класификација

Класификација је једна од најчешћих примјена машинског учења и представља технику додељивања категорије свакој ставци из скупа података. На пример, класификација докумената састоји се од додељивања категорија као што су политика, посао, спорт или вријеме сваком документу, док се класификација слика састоји од додељивања категорије свакој слици као што су аутомобил, воз или авион. Број категорија у таквим задацима често је мањи од неколико стотина, али може бити много већи у неким тешким задацима и чак неограничен као у класификацији текста или препознавању говора. [4]



*Слика 2.8. Примјер класификације у три класе*

Постоје разне технике класификације, укључујући логистичку регресију, к-нн, стабла одлучивања, и наивни бајес. Ове технике су различите по начину на који се користе атрибути мјерења и њихови односи са познатим категоријама. Технике се разликују и по начину додјеле нових објекта у категорије.

Класификација може бити надгледана или ненадгледана. У надгледаној класификацији, подаци садрже позитивне примјере за сваку категорију, док се у ненадгледаној класификацији тражи да алгоритам сам пронађе категорије, а можемо му и рећи колико категорија заправо има у улазном скупу података. Да би класификација била квалитетно одрађена подребно је да се припреме подаци за класификацију, укључујући процес нормализације и екстракције ставки.

Процјена и оцјењивање квалитета класификације се ради помоћу метрика као што су тачности, прецизности и ф-мјере. Овај процес помаже у одређивању колико добро алгоритам функционише у стварним примјенама и омогућује избор најбољег модела за одређени проблем.

Класификација у машинском учењу користи се за аутоматско разврставање објеката и препознавање значаја података, што помаже у доношењу одлука и побољшању пословања у многим индустријама, као што су здравство, финансије, и маркетинг. [5]

### Регресија

Регресија у машинском учењу представља технику коришћену за прогнозирање нумеричке вриједности на основу постојећих података. Регресијско учење користи функцију која повезује улазне вриједности са циљном вриједношћу како би се добиле прогнозе за нове примјере. Регресија се користи за рјешавање проблема предвиђања нумеричке вриједности на основу неколико улазних промјењивих.

У регресији, „казна“ за нетачно предвиђање зависи од величине разлике између истинске и предвиђене вриједности, за разлику од проблема класификације, гдје обично не постоји појам блискости између различитих категорија. [4]

Коришћењем регресије, модел може да стекне увид у релације између улазних вриједности и циљне вриједности, као и да научи да процијени циљну вриједност на основу нових улазних вриједности.

Постоји више врста регресионих модела, укључујући линеарну регресију, полиномијалну регресију и регресију са сигмоидном функцијом. Коришћењем различитих модела, може се доћи до различитих резултата, што захтијева одабир модела који најбоље одговара задатом проблему.

Регресија се може користити за предвиђање различитих вриједности, укључујући количину новца коју ће неко потрошити на куповину производа, цијену некретнине, предвиђање вриједности залиха, варијацију економских промјењивих.

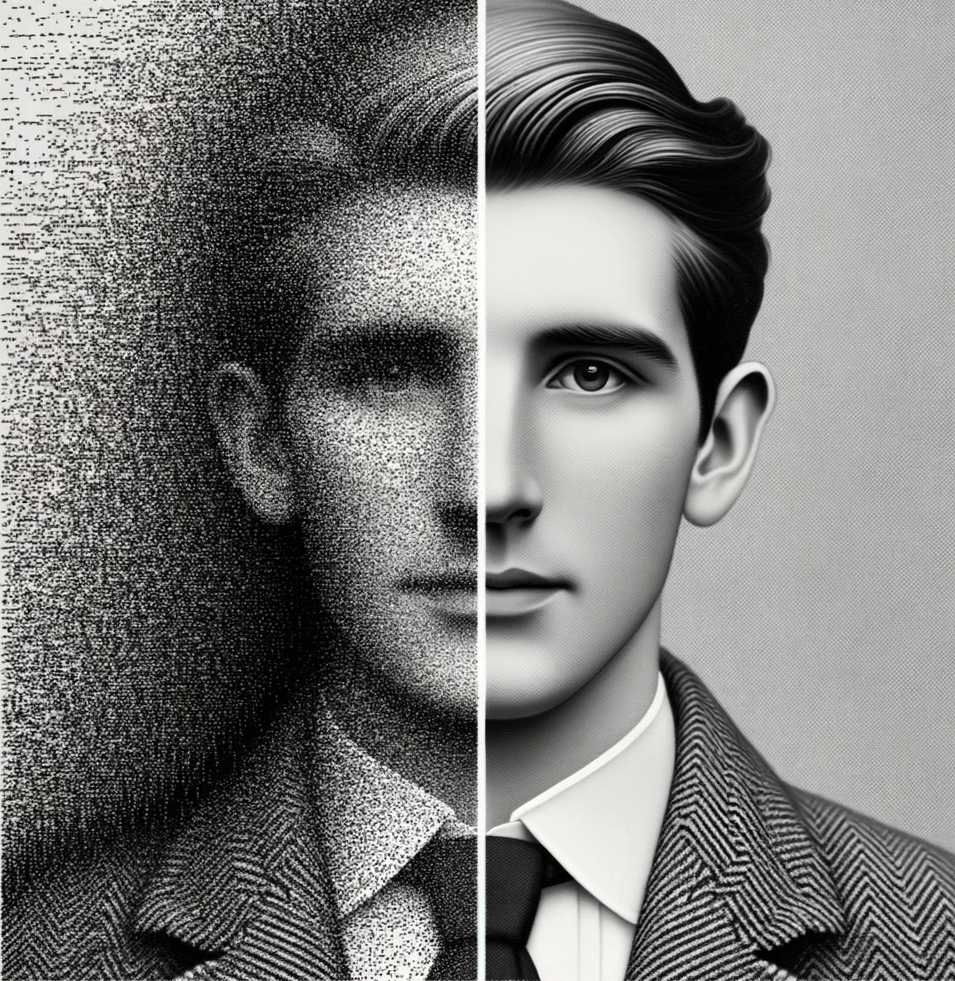


*Слика 2.9. Примјер линеарне регресије*

## Ненадгледано учење

Ненадгледано учење је врста машинског учења код кога алгоритам учења искључиво прима необиљежене податке за учење и даје предвиђања за све невиђене тачке. Алгоритам не прима унапријед дефинисане ознаке или категорије података, већ умјесто тога, покушава сам пронаћи структуру у подацима.

Будући да уопште нема доступних означених примјера, може бити тешко квантитативно процијенити перформансе алгоритма. Груписање и смањење димензионалности су примјери проблема код којих се ненагдледано учење показало као најбоље. [4]



*Слика 2.10. Примјер уклањања шума са фотографије помоћу аутоенкодерске неуронске мреже генерисан помоћу DALE-E алата компаније OpenAI[[10]](#footnote-10)*

У ненадгледаном учењу, систем користи алгоритме попут кластерисања како би груписао сличне податке и створио категорије. Овакав приступ је користан када имамо велики број података са сложеном структуром, а не знамо које су категорије важне за дати проблем.

Постоје два главна подтипа ненадгледаног учења: груписање и смањење димензионалности. Груписање покушава груписати сличне податке у исте кластере како би се створиле категорије. Смањење димензионалности покушава смањити број димензија података како би се добила слика о главним вриједностима и структури података.

Ненадгледано учење често се користи у апликацијама попут сегментације тржишта, анализе купаца и анализе података. Међутим, његове резултате треба пажљиво интерпретирати и процијенити њихову тачност јер недостају унапријед дефинисане ознаке за провјеру резултата.

### Груписање

Груписање или кластеринг (*енг. Clustering)* у машинском учењу је техника сврставања сличних објеката у исте групе, тзв. кластере. Овај приступ се користи за организовање великих количина необрађених података у корисне групе које имају сличне карактеристике.

Груписање се описује као једна од метода за апстраховање информација из података без да се користе подаци о ознакама објеката. Груписање се може провести кориштењем различитих алгоритама, као што су хијерархијско груписање *(енг. Hierarchical clustering)*, груписање методом к-средњих вриједности *(енг. K-means clustering)*, итд.

Груписање се описује као проблем ненадгледаног учења, што значи да алгоритам ради без предефинисаног означавања података. Умјесто тога, алгоритам тражи структуру у подацима и сам саставља групе. може се мјерити различитим метрикама, попут еуклидске удаљености или других.

У сваком случају, груписање је важна техника која се често користи у обради података прије даљег машинског учења, помажући у смањењу димензије података и учинковитијем моделирању. [5]

 *Слика 2.11. Примјер груписања домаћинстав према удаљености од електричног разводника*

### Смањење димензионалности

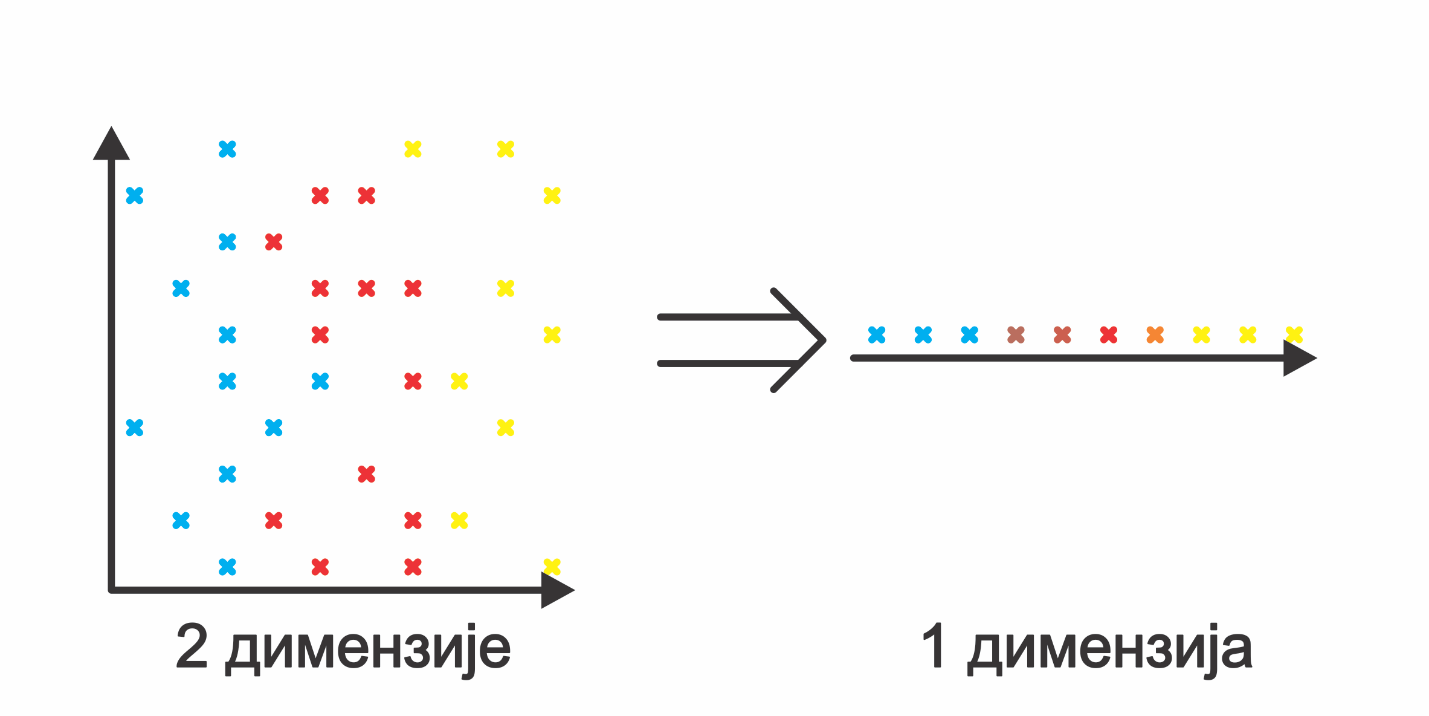
Смањење димензионалности је техника у машинском учењу која се користи за смањивање броја атрибута у подацима. Ова техника се користи како би се смањила сложеност података и убрзао процес машинског учења. Смањење димензионалности се често користи у комбинацији са другим техникама машинског учења, како би се створили бољи модели. Смањење димензионалности може да буде од помоћи и за избјегавање преприлагођавања.

Различите технике смањења димензионалности укључују анализу главних компоненти *(енг. Principal component analysis - PCA*), линеарна дискриминантна анализа *(енг. Linear Discriminant Analisys – LDA)* и мултидимензионално скалирање *(енг. Multidimensional Scaling - MDS)*. Свака од ових техника се користи за различите ситуације и има своје предности и недостатке.

Редукција димензионалности није рјешење за све проблеме у машинском учењу и треба се користити са опрезом како се не би изгубиле важне информације. [5]

Алгоритми који спадају у ову врсту учења су :

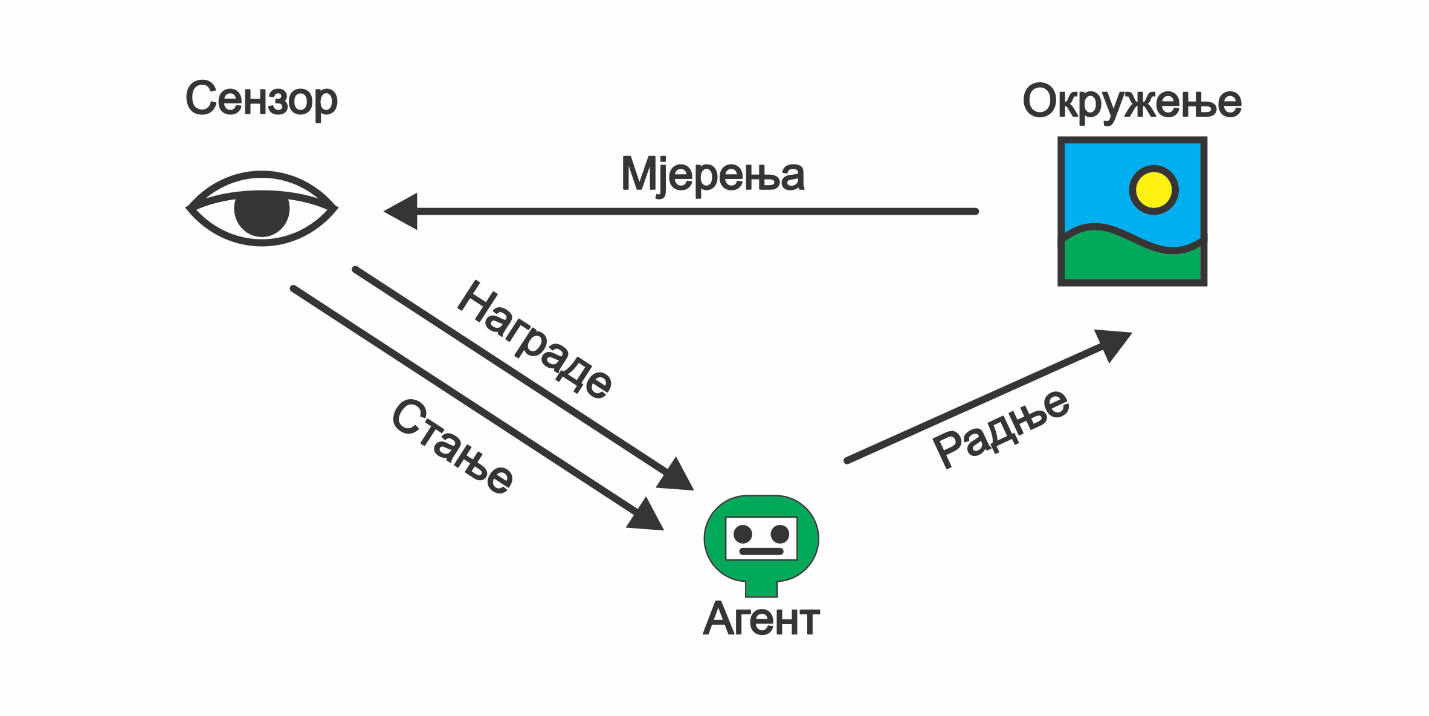
1. K Means
2. PCA (енг. Principal Component Analysis)
3. t-SNE (енг. t-Distributed Stochastic Neighbor Embedding)
4. Правило удруживања (енг. Association rule)



*Слика 2.12. Примјер смањења димензионалности ненадгледаним учењем*

### Учење уз подстицај

Учење уз подстицај *(енг. Reinforcement learning)* је врста машинског учења код кога су фазе обуке и тестирања измијешане у процесу подстицања. Да би прикупио информације, алгоритам учења активно комуницира са окружењем и, у неким случајевима, утиче на окружење и за сваку радњу прима тренутну „награду“ у облику неке нумеричке вриједности, што можемо видјети на *слици 2.4.*

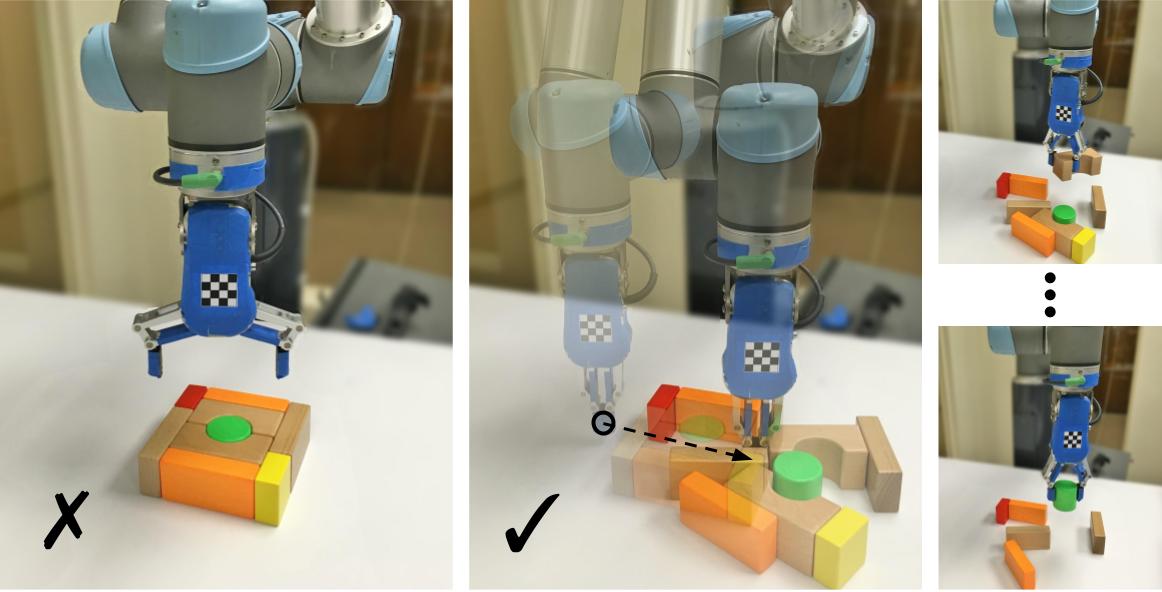


*Слика 2.13. Илустрација агента који учи уз подстицаје*

Циљ алгоритма учења је максимизирање његове награде током процеса подстицања, међутим, окружење не пружа повратне информације о дугорочним наградама, а алгоритам учења се суочава са дилемом истраживање или експлоатација, јер мора да бира између истраживања непознатих радњи (како би стекао више информација) и искориштавања већ прикупљених информација. [4]

Другим ријечима, учење уз подстицај је врста учења у којој модел учи кроз искуства. Он понавља неке активности у окружењу и на основу својих поступака добија награду (подстицај) или казну. Циљ овог учења је пронаћи оптималне поступке којима се постиже највећа награда.

Учење уз подстицај се заснива на идеји да модел представља неког агента у окружењу које садржи неке ресурсе и препреке. Агент се креће кроз окружење и на основу свог понашања добија одређени подстицај или казну. Задатак агента је да пронађе оптималан низ корака који му доноси највећи подстицај.



*Слика 2.14. Примјер роботске руке која учи уз подстицаје да подигне зелени предмет[[11]](#footnote-11)*

Ова врста учења се разликује од других врста машинског учења јер не захтијева да се унапријед дефинишу правила за рјешавање проблема. Умјесто тога, модел учи кроз интеракцију са својим окружењем и на основу својих искустава. Ова врста учења се често користи у проблемима гдје није унапријед познато како се неки проблем може формално дефинисати, као што је случај са играма, управљањем енергијом и управљањем саобраћајем. Учење уз подстицај се имплементира кориштењем алгоритама Q-учења у којима се користе функције награде и Q-вриједности за дефинисање понашања агента у окружењу. [5]

# Процес машинског учења

Процес машинског учења почиње дефинисањем проблема. Послије тога, настављамо са прикупљањем података, који су најчешће велики број мјерења неке појаве (са великим бројем мјерених вриједности). Затим, припремамо податке тако што их чистимо, нормализујемо и претварамо у облик који се може користити у учењу. Наредни корак је одабир модела, што се може урадити уз помоћ литературе, искуства и испробавања различитих модела. Модел се затим тренира на припремљеним подацима кориштењем одређене оптимизационе функције. Након што је модел трениран, важно је да се оцијени квалитет модела, кориштењем метрика попут прецизности и грешке учења. Ако је потребно, може се урадити и подешавање хиперпараметара модела. Када се модел доведе до жељених перформанси, он се користи у стварном свијету како би се предвидјели нови случајеви на којима модел није трениран. [6] На *слици 2.5.* се налазе фазе процеса машинског учења.



*Слика 3.1. Фазе процеса машинског учења*

## Дефинисање проблема

Фаза дефинисања проблема у машинском учењу подразумијева јасно одређивање проблема који се жели ријешити и кориштење алата и техника машинског учења за рјешавање тог проблема. Овај процес укључује утврђивање вриједности које се жели предвидјети, као и одређивање да ли се ради о проблему класификације или регресије. У проблемима класификације, излазна вриједност може бити ознака категорије, док је у проблемима регресије најчешће нека бројна вриједност.

Без јасне дефиниције проблема, модел се може тренирати на погрешан начин, што може довести до недостатка жељених резултата. Стога је кључно прецизно одредити проблем и одговарајући приступ у рјешавању проблема, како би се постигао жељени исход. [7]

## Прикупљање података

Фаза прикупљања података у машинском учењу представља процес узимања података из различитих извора и њихову идентификацију и сакупљање у јединствену колекцију података који ће се користити за тренирање и тестирање машинских модела. Подаци морају бити релевантни за проблем који се покушава ријешити и требали би бити квалитетни и служити као представник скупа података из стварног свијета. Важна пажња се посвећује прикупљању квалитетних података, јер лоши подаци доводе до лоших модела и неадекватних резултата.

Подаци могу бити прикупљени (преузимањем, екстракцијом из база података) из једног извора података или из више извора (базе података, интернет странице). На интернету постоји велики број бесплатних и јавно доступних колекција података. Те колекције података су најчешће унапријед означене и припремљене што олакшава ову фазу машинског учења.

Приликом прикупљања података, потребно је узети у обзир и репрезентативност података и величину података. Потребно је идентификовати и обрадити изворе података, укључујући базе података, текстуалне документе и друге неформализоване изворе. Важно је такође размотрити проблеме као што су неконзистентност и нерепрезентативност података и њихов утицај на перформансе модела.

Потребно је пажљиво одабрати и прикупити податке који ће бити репрезентативни за проблем који се покушава ријешити како би се створио поуздан модел са високим перформансама.[8].

## Припрема података

Фаза припреме података се често описује као кључна у процесу машинског учења. [7] Ова фаза укључује чишћење, трансформацију и припрему података за обраду и тренирање модела.

Једна од најважнијих активности у припреми података је уклањање недостатака и неконзистентности у подацима. То може укључивати исправљање грешака у уносу података, уклањање дуплих записа и неважећих вриједности.

Трансформација података укључује обраду података тако да се припреме за тренирање модела. То може укључивати нормализацију података, кодирање категоризацијских промјенљивих и издвајање нових атрибута из постојећих података.

Затим, подаци се морају припремити за тренирање модела. То укључује раздвајање података у скупове за тренирање и тестирање, како би се модел тренирао на једном скупу, а потом проверио његову перформансу на другом скупу.

Без добро припремљених података, модел може бити неуспјешан у њиховој предикцији и генерализацији на нове податке.

## Избор модела

Избор модела у машинском учењу подразумијева одабир најприкладнијег алгоритма и имплементацију тог алгоритма у виду модела. Одабир модела се обично темељи на неколико фактора, укључујући тип проблема који се жели ријешити, квалитет и количину прикупљених података, и ресурсе које имате на располагању.

Избор модела се објашњава кроз неколико фаза. Прва ствар коју треба учинити је дефинисати проблем који се жели ријешити. [7] Ова дефиниција омогућава одабир адекватног типа модела. Након што се дефинише проблем, следећи корак је преглед релевантних модела и препорука од стране стручњака у пољу.

Након што се дефинише проблем и препоручи неколико модела, следећи корак је евалуација ових модела у складу са дефинисаним критеријумима, као што су прецизност, брзина тренирања и примјене, стабилност, робустност, интерпретабилност и сл. Након што се дефинишу критеријуми, следећи корак је стварни избор модела уз помоћ приступа који укључује, примјерице, одабир најбољег модела уз помоћ одабира на основу испитивања, одабир модела на основу стручног знања или мјешавине тих приступа.

Погрешан избор модела може резултирати неадекватним перформансама. Због тога је важно да се врши евалуација различитих модела како би се одабрао најбољи, кориштењем метода попут крос-валидације и евалуације перформанси на тестном скупу података.

Неки од фактора које треба узети у обзир при одабиру модела су врста проблема, квалитет података и ресурси на располагању.

## Тренирање модела

Тренирање модела у машинском учењу представља процес у коме се моделу дају подаци како би се научио како да класификује или регресира нове примјере. Тренинг модела се може описати као процес оптимизације функције губитка (loss function) или функције циља (objective function). [7] Функција губитка одређује колико је добар модел у предвиђању, а функција циља се користи за оптимизацију модела у смислу да се максимизије та функција.

Процес тренинга се обично спроводи коришћењем алгоритама за оптимизацију као што су градијентно спуштање (*енг. gradient descent*), стохастичко градијентно спуштање и други. Модел се тренира користећи тренинг податке, а затим се оцјењује користећи валидационе податке.

Након што је модел трениран, вриједност функције губитка се смањује, што значи да модел постаје бољи у предвиђању. Укупан број епоха које се тренира модел може варирати, а одлука о томе када прекинути тренинг зависи од многих фактора, као што су сложеност модела, величина скупа података, брзина оптимизације итд.

Важно је напоменути да процес тренирања модела није увијек једноставан и може захтјевати велике ресурсе, посебно ако се користе комплексни модели или велики скупови података. Стога је важно добро разумјети овај процес како би се постигла адекватна предикција и генерализација модела на нове податке.

## Оцјењивање модела

Оцјена перформанси модела се користи како би се процијенила његова способност предвиђања, што омогућава праћење и тестирање квалитета модела.

Постоје различите методе за оцјењивање модела, укључујући унакрсну провјеру (cross-validation), провјеру са непознатим скупом података (holdout validation) и процјену грешке (error estimation). Ове процедуре оцјењивања користе се за одређивање прецизности модела, капацитета генерализације и сличних мјера квалитете.

У књизи се такођер описују различити критерији за оцјењивање модела, укључујући прецизност (accuracy), прецизност учења (learning accuracy), прецизност генерализације (generalization accuracy), F1 скор и ROC криву. Ови критеријуми помажу у одређивању на које метрике се потребно фокусирати када се процјењују модели.

Важно је напоменути да коришћење ових критеријума зависи од конкретне ситуације у којој се модел примјењује и циљевима које се желе постићи, стога је важно разумјети њихове предности и недостатке при одабиру метрика за оцјењивање модела. Потребно је прилагодити одабир критерија за оцјењивање модела специфичним потребама поједине примјене, те процјенити предности и мане сваког критерија у односу на циљеве и очекиване перформансе модела.

## Прилагођавање модела

Прилагођавање (адаптација) модела у машинском учењу представља процес подешавања модела који се користи за објашњавање или предвиђање података. Циљ прилагођавања је побољшање перформанси модела тако што ће се оптимизовати његови параметри.

Нпр. ако се модел обучава на скупу података са сликама мачака и паса, а потом се примењује на скупу података са сликама дивљих животиња, модел ће се вјероватно лоше понашати јер се структура и дистрибуција података разликују. У том случају, прилагођавање модела би се могло провести додавањем нових примјера из новог скупа података у оригинални скуп података за обуку модела или подешавањем постојећих параметара модела како би се побољшала његова способност за препознавање нових примјера.

Постоје два основна начина за прилагођавање модела: надгледано и ненадгледано учење. Надгледано учење користи означене податке с жељеним излазима и користи их за изградњу модела који се потом оптимизира користећи алгоритме који подешавају параметре модела на најбољи начин. Ненадгледано учење користи податке који нису означени с жељеним излазима и користи их за идентифицирање структура и значајки у подацима. Ова метода користи алгоритме који групишу податке и оптимизирају параметре модела како би се добила што боља груписања.

Постоје разне методе оптимизације које се користе за прилагођавање модела, укључујући градијентни спуст, квази-Њутн методе и Марков-Ланов процес. Пприлагођавање модела може бити комплицковано и захтјевно, па често захтијева интеракцију стручњака за машинско учење с експертима у конкретном домену како би се постигли најбољи резултати.

## Тестирање и примјена модела

Процес тестирања и примјене модела у машинском учењу представља последњи корак у изради модела. Циљ је провјерити колико добро модел ради на независном скупу података који није кориштен за обуку. Ако се модел показао прецизним на обучавајућим подацима, то не значи да ће бити прецизан и на новим, независним подацима.

Након што се модел тестира и процијени, потребно је провјерити колико је он генерализирајући и способан радити на цијелом скупу података. Ако се модел покажe неадекватним, потребно га је прилагодити и поново тестирати док се не постигне жељени ниво тачности.

Након што се постигне жељени ниво тачности, модел се може примијенити на стварним проблемима и користити за доношење одлука или предвиђање резултата. Међутим, ако се модел користи у стварној примјени, неопходно је редовно пратити његову прецизност и прилагођавати га како би се осигурало његово постојано постизање прецизности.

# Избор алгоритма

Избор алгоритма у машинском учењу је један од најваљнијих корака у процесу креирања модела, јер одређује квалитет модела и његову ефективност. Различити фактори играју клучну улогу у избору алгоритма:

* Природа података: карактеристике података, као што су величина и врста, могу утицати на избор алгоритма. На пример, ако су подаци дискретни, неки алгоритми као што су дрво одлучивања могу бити погоднији од алгоритама заснованих на регресији.
* Циљ проблема: циљ проблема, као што је класификација, регресија или кластеровање, такође може утицати на избор алгоритма.
* Квалитет података: квалитет података, укључујући присуство непотпуних и неодређених података, може утицати на избор алгоритма.
* Перформансе: треба узети у обзир перформансе алгоритма у погледу брзине и прецизности.
* Комплексност модела: такође је важно размотрити комплексност модела и способност тумачења модела.

Избор алгоритма захтијева балансирање између перформанси, комплексности и интерпретабилности, у зависности од специфичности проблема и природе података. Коришћењем тих фактора, може се изабрати најбољи алгоритам за дати проблем.

## Логистичка регресија

Логистичка регресија (енг. Логистиц Регрессион) је класична статистичка метода која се користи за моделирање односа између једне или више значајки (предицтора) и бинарне категоријске зависне варијабле. У машинском учењу, овај алгоритам често се користи за класификацију проблема, у којима је циљ да се предвиди припадност објеката некој од двије класе. Основа алгоритма је линеарна регресија, али умјесто да се користи континуална зависна варијабла, логистичка регресија користи логистичку функцију (енг. логистиц фунцтион) како би се моделирао однос између значајки и бинарне зависне варијабле. Логистичка функција враћа излаз између 0 и 1 који се може интерпретирати као вероватноћу припадности објекта једној од двије класе.

У процесу тренирања, алгоритам користи градијентни спуст (енг. градиент десцент) или неку другу оптимизацијску технику како би се пронашли оптимални коефицијенти у логистичкој функцији. Ова оптимизација се врши кроз итеративне кораке у којима се рачунају градијенти функције губитка (енг. лосс фунцтион) и користећи их се ажурирају коефицијенти у логистичкој функцији.

Након што се алгоритам тренира, може се користити за предвиђање припадности нових објеката једној од двије класе на основу вриједности њихових значајки. У овом случају, логистичка функција се користи за израчунавање вероватноће припадности објекта некој класи, а коначна класификација се врши тако што се вероватноћа пореди са неким прагом (енг. тхресхолд), обично 0.5.

Укупно, логистичка регресија се сматра једноставним и ефикасним алгоритмом у многим класификацијским проблемима. Међутим, његова ефикасност зависи од многих фактора, укључујући квалитет значајки, величину скупа података и врсту проблем

## Gaussian Naive Byes Classifier

Гауссиан Наиве Баyес Цлассифиер је алгоритам класификације базиран на Баyесовој теорему и претпоставци да свака карактеристика у подацима има нормалну расподелу. Алгоритам се користи за класификацију у којима имамо н-категорија и н-карактеристике за сваку инстанцу у подацима.

Процес функционисања Гауссиан Наиве Баyес Цлассифиер-а почиње са рачунањем Гауссиан Пробабилитy Денситy Фунцтион (ПДФ) за сваку карактеристику у свакој категорији. Овај ПДФ описује како се карактеристике расподељују у датој категорији. Након тога, Баyес теорема се користи за рачунање вјеројатности за сваку категорију за дату инстанцу података. Вјеројатности се комбинирају за све карактеристике у датој инстанци и на крају се израчунава коначна вјеројатност за сваку категорију. Инстанца се класификује у категорију са највећом вјеројатношћу.

Међутим, претпоставка да свака карактеристика има нормалну расподелу често може бити погрешна у стварним подацима, што може довести до лоших резултата класификације. Стога је важно провјеравати и примјењивати адекватне претпоставке о расподели када се користи овај алгоритам.

Укупно, Гауссиан Наиве Баyес Цлассифиер је једноставан и ефикасан алгоритам за класификацију, а посебно се добро показује у случајевима са великим бројем категорија и малим бројем карактеристика.

## K Nearest Neighbors Classifier

К Неарест Неигхборс Цлассифиер (КНН) алгоритам је један од најједноставнијих и најчешће кориштених алгоритама у машинском учењу, коришћен за класификацију. Овај алгоритам користи концепт к-најближих суседа за класификацију нових објеката.

Како ради овај алгоритам? Алгоритам КНН представља једноставну верзију алгоритма класификације, где се узима у обзир к-најближих суседа објекта који се покушава класификовати. Ова к-вредност се најчешће бира преко кросс-валидације.

Одређивање к-најближих суседа се врши на основу израчунате удаљености између новог објекта и свих објеката из тренинг скупа. Након тога, објекти са најмањом удаљеношћу су дефинисани као к-најближи суседи.

На крају, класификација новог објекта се врши преко гласања к-најближих суседа, тако што се узима у обзир најчешћа класа међу њима.

У поређењу са другим алгоритмима, КНН је једноставан за примену, а истовремено ефикасан у класификацији различитих типова података. Међутим, један од недостатака овог алгоритма је што се рачунаћују удаљености између свих објеката у тренинг скупу и новог објекта, што може бити захтевно за израчунавање за велике податке.

## Decision Tree Classifier

Децисион трее Цлассифиер алгоритам је један од најчешћих алгоритама у машинском учењу, који се користи за разврставање података. Овај алгоритам користи стабло одлучивања као своју главну структуру, где се разматрају различите атрибуте унутар података и доносе се одлуке о томе како да се подаци разврстају.

Стабло одлучивања користи две врсте чворова: чвор са члановима и лист. Чвор са члановима разматра неки атрибут и одлучује да ли подаци треба да иду у један од својих подчворова, који се затим поново тумаче. Ова процес тумачења се понавља све док не дође до листа, који означава коначно разврставање података.

Децисион трее Цлассифиер алгоритам се може користити за разврставање података у више класа, као и за класификацију података. Алгоритам функционише тако што се користи прошли подаци како би се створило стабло одлучивања, које се онда користи да се разврстају нови подаци.

Овај алгоритам је једноставан за разумевање и имплементирање, што га чини идеалним за многе апликације. Међутим, постоје неке недостатке као што су склони пренаучењу и потреба за одређивањем праве границе када се разматрају атрибути.

Укупно, децисион трее Цлассифиер алгоритам је корисчан и ефикасан алат за разврставање података у машинском учењу, који се користи у широком спектру апликација, укључујући финансије, трговину, медицинске и друге индустрије.

## Random Forest Classifier

Рандом Форест Цлассифиер је енсембле машинског учења алгоритам који се користи за класификацију и регресију. Овај алгоритам користи мноштво одвојених стабала одлучивања (тј. Децисион Треес) за добијање закључка. На почетку, користи се метода случајног узорковања да се формира више одвојених стабала одлучивања. Свако од њих даје свој закључак на основу своје процене, а затим се користи гласање да се добије финална процена.

У односу на класично стабло одлучивања, рандом форест користи више стабала одлучивања које су створене користећи случајни избор атрибута и података. То доводи до више робустности и мање вероватноће оверфиттинг-а. Такође, овај алгоритам има бољу перформансу у случајевима када има много атрибута, што често може довести до проблема у класичном стаблу одлучивања.

Коришћење рандом форест алгоритма може се реализовати за различите типове проблема, укључујући класификацију, регресију и идентификацију најважнијих атрибута. Укупно, рандом форест је један од најчешће коришћених алгоритама у машинском учењу због своје робустности и широке примене у различитим областима.

## Gradient Boosting Classifier

Градиент Боостинг Цлассифиер је алгоритам за машинско учење који се користи за класификацију. Овај алгоритам је комбинација неколико класификатора, а сваки од њих је научен да поправља грешке претходних класификатора.

Градиент Боостинг Цлассифиер користи технику "боостинг" за побољшање перформанси класификације. Ова техника укључује додавање нових класификатора док се не постигне жељени ниво тачности. Нови класификатори се додају на бази претходно направљених грешака у нади да ће се оне исправити.

Градиент Боостинг Цлассифиер такође користи технику "градиент десцент" за одређивање правца у ком ће се класификатори научавати. Градиент десцент користи функцију губитка (лосс фунцтион) да би се утврдила удаљеност између тренутног стања и жељеног стања. Ова функција губитка се користи да би се пронашло оптимално решење, које ће смањити укупне грешке.

Као резултат, Градиент Боостинг Цлассифиер обично даје боље резултате од других једноставнијих класификатора, попут логистичке регресије или Наиве Баyес класификатора. Међутим, Градиент Боостинг Цлассифиер је такође склон оверфиттинг-у, стога је важно пронаћи оптималну комбинацију броја класификатора и хиперпараметара.

## Support Vector Machine Classifier

Суппорт Вецтор Мацхине (СВМ) Цлассифиер је алгоритам машинског учења који користи математичке моделе како би предвидио категорију новог примера. СВМ користи теорију оптимизације да би формирао један или више хиперсуперфиција (тј. линија или хиперраван) која најбоље раздваја примере у скупу података у одговарајуће категорије.

Приликом формирања хиперсуперфиција, СВМ циља на то да пронађе хиперсуперфицију која је најудаљенија од најближих примера из сваке категорије, што се назива маргином. Овај приступ омогућава СВМ-у да створи робустније и генерализоване моделе, што смањује ризик од преувеличавања (тј. претренираности) на скупу података за тренирање.

СВМ такође има и два главна мода рада: Линеарни СВМ који користи линеарне хиперсуперфиције за раздвајање примера у категорије, и нелинеарни СВМ који користи нелинеарне хиперсуперфиције. Нелинеарни СВМ се може користити у ситуацијама када линеарни СВМ не може довољно добро да раздвоји примере у категорије.

Поред тога, СВМ такође подржава и технике регуларизације, што га чини погодним за рад са великим скуповима података и тешким проблемима класификације. Укупно, СВМ Цлассифиер је један од водећих алгоритама у области машинског учења и често се користи у различитим применама, укључујући анализу текста, слика и геномске податке.

## LightGBM Classifier

Лигхт ГБМ је алгоритам за машинско учење који се користи за класификацију и регресију. Овај алгоритам је развијен као оптимизована верзија Градиент Боостинг Мацхине (ГБМ), користећи технике које убрзавају процес тренинга. Лигхт ГБМ користи технологију усмереног успоравања учења, што значи да се фокусира на тешке примјере у сваком кораку учења, како би се што брже извршио процес.

Лигхт ГБМ се разликује од стандардних ГБМ алгоритама по томе што користи хистограм за распођелу особина у сваком чвору. Овај приступ значајно смањује потребан број итерација у процесу учења, а такође олакшава и бржи приступ подацима. Лигхт ГБМ такође користи бинарну тражњу, што даље смањује вријеме потребно за извршење процеса.

Лигхт ГБМ такође има способност учења на различитим типовима података, укључујући и велике величине података и спојене особине. Овај алгоритам такође има добру перформансу на проблемима са небалансираним подацима, када има више примјера једне класе него друге.

Укупно, Лигхт ГБМ је ефикасан и брз алгоритам за машинско учење, посебно за проблем класификације и регресије, а његова употреба се све више шири у индустрији и истраживачким заједницама.

## K Nearest Neighbors Regressor

К Неарест Неигхборс Регрессор алгоритам у машинском учењу је тип регресијског модела који се заснива на принципу к најближих сусједа. Овај алгоритам се често користи за предвиђање континуалних вриједности, попут цијене некретнине или просјечне мјесечне температуре. Алгоритам функционира тако што се за сваку инстанцу у скупу података за учење проучавају н најближих сусједа и израчунава се просјек вриједности варијабле циља за те сусједе. Коначно, добивени просјек се користи као предвиђена вриједност за дату инстанцу.

Кључни параметар овог алгоритма је број најближих сусједа, што се може одабрати помоћу валидацијске методе, као што је цросс-валидатион. Такођер, укључивање тежина за сусједе такођер може утјецати на квалитету предикција, гђе су сусједи који су ближе инстанци за предикцију тежи.

Међутим, овај алгоритам има и неколико недостатака, укључујући слабу перформансу у случајевима гђе постоји много димензија и висок ступањ интеракције међу варијаблама, што може резултирати проблемима с пренаучености.

Укратко, К Неарест Неигхборс Регрессор алгоритам је једноставан и брз за тренирање те често даје добре резултате у случајевима гђе су подаци хомогени и дистрибуција варијабле циља није превише сложена. Међутим, потребно је пажљиво одабрати параметре и провјерити перформансу модела на скупу података за тестирање како би се избјегли проблеми са пренаучењем.

## LightGBM Regressor

ЛигхтГБМ је врста регресијског алгоритма у машинском учењу који користи технику усмјерену према градијентном боостингу. Главна разлика између ЛигхтГБМ и других алгоритама градијентног боостинга је у томе што користи двоструко бинарно пођелу за брзо рјешавање проблема скалирања.

У раду ЛигхтГБМ-а, стабла одлучивања се користе као главни модели за регресију. Они се настављају једно за другим у фазама, у свакој фази користећи информације из претходне фазе за побољшање тачности.

Код ЛигхтГБМ-а, постоје два главна параметра за оптимизацију: број стабала одлучивања и максимална дубина сваког стабла. Број стабала одлучивања одређује колико ће модела бити створено, док максимална дубина сваког стабла одређује колико ће комплексан бити сваки модел.

ЛигхтГБМ такођер користи технику смањења димензионалности како би се избјегло превише сложености и оверфиттинг-а. То се постиже одабиром најзначајнијих значајки које ће се користити за сваки модел.

ЛигхтГБМ је популаран због своје брзине и ефикасности, посебно када се ради са великим скуповима података. Он је такођер способан радити с различитим врстама података, укључујући и податке с различитим скалама и категоричке податке. Укупно, ЛигхтГБМ је један од најбољих алгоритама регресије у машинском учењу, и користи се у многим реал-њорлд примјенама, укључујући предвиђање цијена, кредитне процјене и персонализацију препорука.

# Практични рад

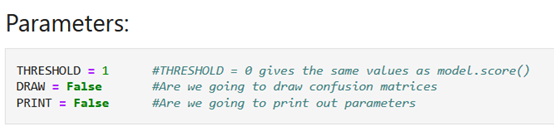
Задатак: Анализирати моделе за предвиђање броја особа у просторији креиране кориштењем неколико алгоритама (класификационих и регресионих). Моделе је потребно тренирати подацима који садрже температуру, влажност, ниво угљен-диоксида и др. За реализацију користити Jupyter Notebook алат. Извршити компаративну анализу перформанси добијених модела.

Jupyter Notebook је интерактивно окружење за писање и извршавање кода које је првобитно развијено за подршку језицима као што су Julia, Python и R, али сада подржава многе друге језике. Карактерише га способност комбиновања кода, текста, графике и формула у једном документу.

Документ се састоји од ћелија које могу садржавати различите типове садржаја и најчешће се инсталира кроз дистрибуцију Anaconda, која садрже многе научне библиотеке за Python. Поред класичног Notebook-а, постоји и JupyterLab, који је сљедећа генерација Jupyter интерфејса и нуди проширене могућности за рад са више докумената и бољу интеграцију с другим алатима.

Због своје интерактивности и флексибилности, Jupyter Notebook је постао стандардни алат у областима анализе података и машинског учења. Постоји и могућност извоза Notebook докумената у разне формате, што олакшава дијељење и презентацију резултата.

У току развоја рјешења смо користили сљедеће параметре да означимо да ли желимо да се исцртавају или исписују одређени излази. Такође, имамо и праг за прорачун тачности модела.



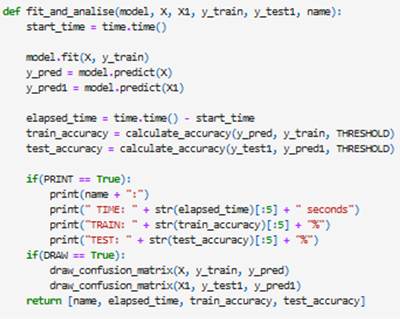
*Слика 5.1. Дефинисање параметара*

## Помоћне функције

Како бисмо што читкије ријешили проблем, дефинисали смо три помоћне функције. Прва фукција је кориштена за рачунање тачности модела. Друга функција је била за исцртавање матрице конфузије. Посљедња фунција моји смо морали да креирамо је била функција за тренирање и анализу модела.



*Слика 5.2. Помоћне функције за рачунање прецизности и исцртавање матрице*



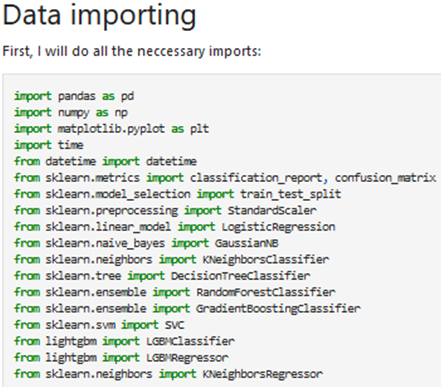
*Слика 5.3. Помоћна функција за прилагођавање и анализирање модела*

## Прикупљање података

Подаци су преузети са отвореног BitLab репозиторијума који је креиран за мастер рад "CO2 based room occupancy detection : an IoT and machine learning application" Bockstael, Nicolas ; Jadin, Alexandre.

У првој цјелини увозимо све библиотеке које ће бити кориштене у коду:

* **pandas** за манипулацију и анализу података
* **numpy** за математичке операције над подацима
* **matplotlib.pyplot** за визуализацију података
* **time** за мјерење времена извршавања тренирања модела
* **datetime** за рад са датумима и временима
* **classification\_report** и **confusion\_matrix** за процјену перформанси модела
* **train\_test\_split** за подјелу скупа података на скупове за тренирање и тестирање
* **LogisticRegression, GaussianNB, KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, SVC, LGBMClassifier, LGBMRegressor, KNeighborsRegressor** за израду модела за класификацију и регресију на основу одговарајућих алгоритама и техника машинског учења

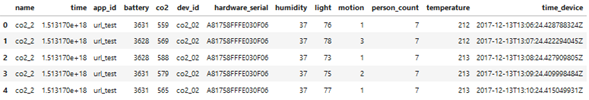
****

*Слика 5.4. Увоз потребних спољашњих библиотека*

Даље, учитавамо csv датотеку која садржи скуп података за тренирање и исписуемо првих пет редова скупа података.



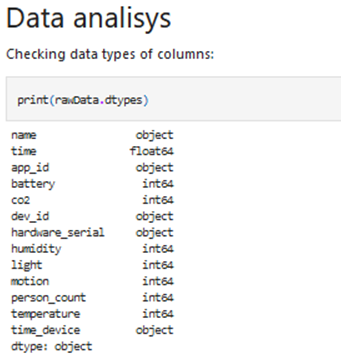
*Слика 5.6. Учитавање датотеке са подацима*



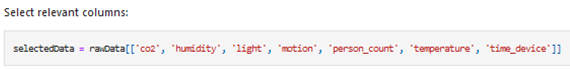
*Слика 5.7. Приказ првих пет редова табеле са подацима*

## Припрема података

У овој цјелини прво исписујемо типове података за сваку појединачну колону и бирамо који ће подскуп података бити кориштен за генерисање модела. У овом случају бирамо седам атрибуте: co2, humidity, light, motion, person\_count, temperature, и time\_device.

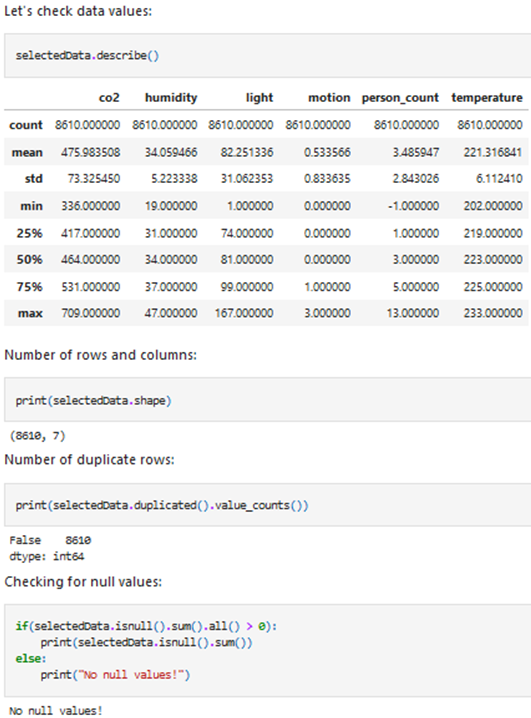


*Слика 5.8. Исписивање типова података за колоне скупа података*



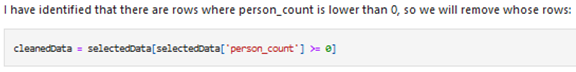
*Слика 5.9. Избор одговарајућих колона у скупу података*

Након тога, рачунамо основне статистичке мјере (број, средњез вриједност, стандарднз девијација, минимум, максимум, квартили) за сваку колону у скупу изабраних података, исписујемо облик (димензије) изабраног скупа података, број дупликата у њему и провјеравамо да ли у скупу података имамо null вриједности и исписујемо њихов број.



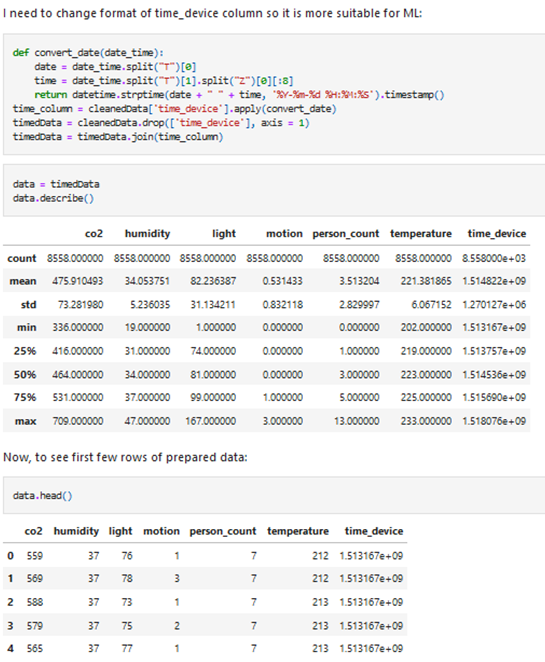
*Слика 5.10. Исписивање основних информација о скупу података*

Даље уклањамо све редове у којима је просторија била празна.



*Слика 5.11. Уклањање редова за случајеве када је просторија била празна*

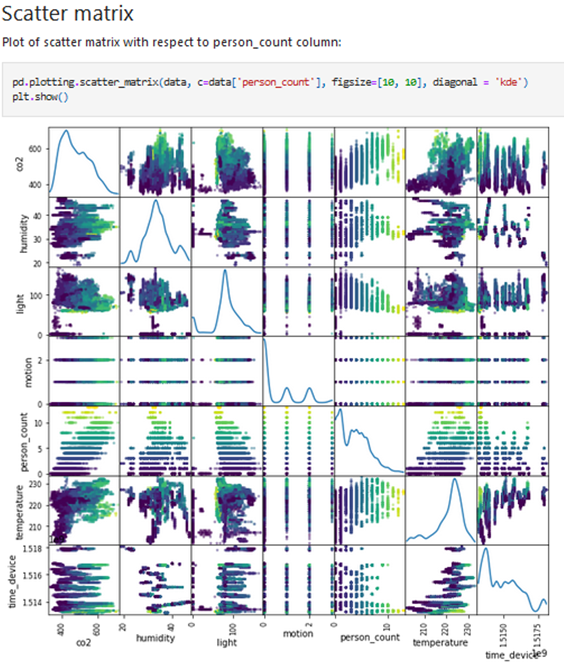
Пошто смо примијетили да датум и вријеме нису у одговарајућем формату, дефинишемо convert\_date функцију како би се претворила колона time\_device из низа знакова у формат датума и времена, па извршавамо функцију над скупом података и спремамо нове вриједнсти датума и времена у нову колону time\_column, а затим поново приказујемо основне статистичке мјере и првих 5 редова скупа података.



*Слика 5.12. Дефинисање функције за конверзију датума у одговарајући облик*

Након овога, можемо да израчунамо и прикажемо и матрицу распршености нашег скупа података. Овај графикон омогућава визуално поређење дистрибуција између атрибута, као и идентификацију евентуалних зависности или образаца међу атрибутима у скупу података.

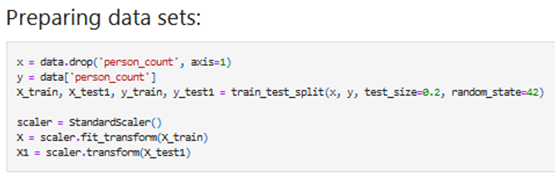
Пошто је нема смисла да рачунамо корелацију колоне према самој себи, на главној дијагонали можемо да видимо глатку криувуљу дистрибуције података за поједине колоне, чија висина и облик показују густоћу расподјеле података, што може бити корисно за разумијевање карактеристика података.



*Слика 5.13. Приказ матрице распршености*

Скуп података смо подијелили на скупове тако што смо прво подијелили колону за број особа из скупа података, а онда смо креирали сљедеће скупове:

* X – скуп улазних података за тренирање модела
* X1 – скуп улазних података за тестирање и оцјењивање модела
* y\_train – скуп излазних података за тренирање модела
* y\_test.- скуп излазних података за тестирање и оцјењивање модела

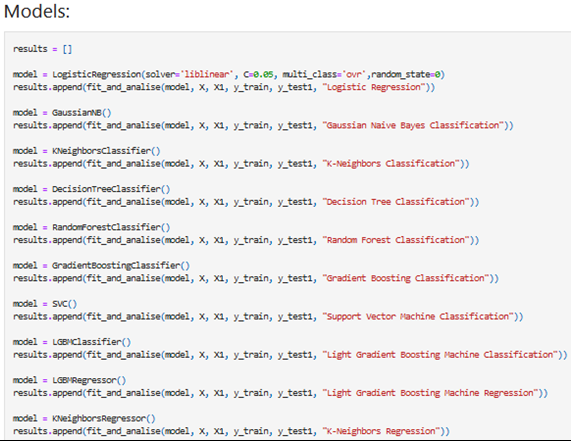


*Слика 5.14. Подјела скупа података на одговарајуће подскупове*

## Избор и тренирање модела

Наизмјенично смо бирали, креирали и записивали резултате за сљедеће алгоритме:

* LogisticRegression
* GaussianNB
* KNeighborsClassifier
* DecisionTreeClassifier
* RandomForestClassifier
* GradientBoostingClassifier
* SVC
* LGBMClassifier
* LGBMRegressor
* KNeighborsRegressor

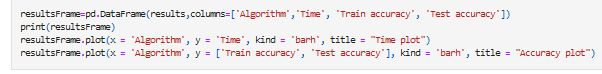


*Слика 5.15. Тренирање модела*

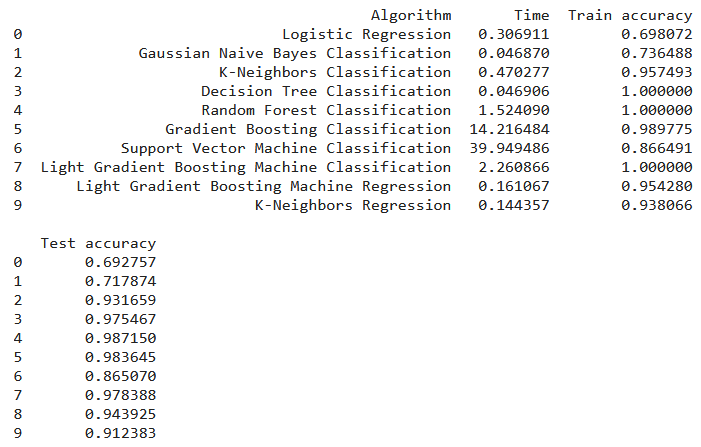
## Оцјењивање модела

На крају, потребо је да прикажемо резултате и перформансе свих модела кориштених за тренирање и тестирање.

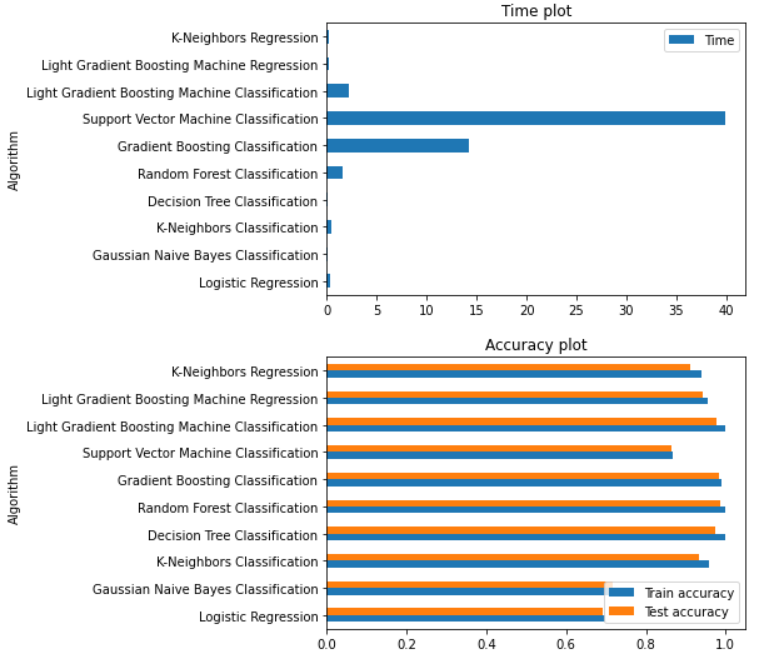
Приказујемо табеларно помоћу Time plot дијаграма све моделе који су кориштени заједно са временом извршавања и тачношћу алгоритма. Овај дијаграм нам помаже да визуално упоредимо брзину извршавања сваког модела и идентификујемо моделе који су релативно спори у односу на остале.

Accuracy plot дијаграм приказује тачност сваког модела на скуповима за тренирање и тестирање. Ова линија кода пружа визуелни преглед тачности различитих модела и помаже у идентификовању модела који су најпрецизнији.

*Слика 5.16. Приказивање резултата тренирања*



*Слика 5.17. Приказ резултата тренирања*



*Слика 5.8. Графички приказ резултата тренирања*

# Резутати

У анализи је испитано десет модела за процјену особа у просторији. Сваки модел пружа јединствен скуп предности и недостатака, а њихова ефикасност може се разликовати у зацисности од специфичног контекста употребе.

Logistic Regression је брз модел с врећемон извршавања од само 0.307 секунди, али са тачношћу тестирања од 69.3%, не пружа довољну прецизност за неке захтјевније примјене. Слично томе, Gaussian Naive Bayes Classification је још бржи, али и даље пружа тачност испод 75%.

Gradient Boosting Classification, Support Vector Machine Classification и Support Vector Machine Classification су спорији модели али пружају врло добре резултате тачности тестирања.

Модел попут K-Neighbors Classification, Decision Tree Classification i Light Gradient Boosting Machine Classification комбинују брзину с високом тачношћу, прелазећи 93% на тестирању. Међутим, савршена тачност тренирања код Decision Tree и Light Gradient Boosting алгоритама може указивати на потенцијално преприлагођавање, што значи да модел може неадекватно генерализовати на невиђеним подацима.

Иако и други алгоритми пружају снажне перформансе, Random Forest Classification се истиче као најпрецизнији са изузетном тачношћу тестирања од 98.7%. Иако није алгоритам који се извршава најбрже, вријеме извршавања од око 1.5 секунди је прихватљиво за потребе мјерења броја особа у просторији, пошто ће се модел само једном тренирати. Његова савршена тачност тренирања такође указује на могућност преприлагођавања.

# Закључак

Машинско учење ствара нове могућности у свијету технологије, омогућавајући рачунарима да „уче“ из података и доносе одлуке које превазилазе класично програмирање. Ове способности постају кључне у ситуацијама гдје су проблеми комплексни и динамични.

У посљедњих неколико година, машинско учење и вјештачка интелигенција доживјели су праву ренесансу захваљујући брзим технолошким иновацијама. Савремене примјене машинског учења нису само унаприједиле традиционалне области попут препознавања слика или предвиђање текста, већ су и отвориле врата за револуционарне могућности у интеракцији са рачунарима и обради природног језика (OpenAI ChatGPT) и (Google BARD).

У оквиру овог дипломског рада, фокус је на примјени машинског учења у контексту предвиђања броја особа у просторији. Таква предвиђања могу имати широку примјену, од оптимизације клима уређаја до повећања безбједности простора.

Током израде овог рада, нагласак је стављен на цјелокупан процес развоја рјешења. Почетак је обиљежило прикупљање релевантних података (проналазак одговарајућег скупа података), који су основ сваког модела машинског учења. Након тога, подаци су обрађени и анализирани како ви се идентификовали кључни обрасци и информације. Овај корак је од суштинског значаја, јер квалитет улазних података директно утиче на тачност и ефикасност коначног модела.

Правилна процјена модела је неопходна како би се осигурало да модел не само да ради добро на познатим подацима, већ и да може генерализовати своје знање на нове, невиђене ситуације. То се постиже пажљивом подјелом података на тренинг, валидациони и тест скуп.

Без обзира на све техничке аспекте и изазове, циљ овог рада је био развити рјешење које може да се користи у стварном свијету, пружајући тачна и брза предвиђања броја особа у просторији. И док сваки модел има своје предности и слабости, права вриједност долази из његове способности да пружи практичне резултате у стварном окружењу.

# Литература

|  |  |
| --- | --- |
| [1] | D. Kirsch и J. Hurwitz, Machine Learning For Dummies, New Jersey: John Wiley & Sons, Inc., 2018. |
| [2] | F. Williams, Meet the nine billion-dollar companies turning a profit from sustainability, The Guardian, 2016. |
| [3] | S. Russell и P. Norvig, Artificial IntelligenceA Modern Approach, New Jersey: Pearson Education, Inc., 2010. |
| [4] | M. Mohri, A. Rostamizadeh и A. Talwalkar, Foundations of machine learning, Cambridge, Massachusetts: The MIT Press, 2018. |
| [5] | T. Mitchell, Machine Learning, New York: McGraw-Hill, 1997. |
| [6] | E. Alpaydin, Introduction to Machine Learning, Cambridge, Massachusetts: The MIT Press, 2010. |
| [7] | G. James, D. Witten, T. Hastie и R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, New York City, USA: Springer, 2017. |
| [8] | I. H. Witten, E. Frank и M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Burlington, USA: Morgan Kaufmann, 2011. |

1. <https://www.google.com/> [↑](#footnote-ref-1)
2. <https://www.microsoft.com/> [↑](#footnote-ref-2)
3. <https://www.facebook.com/> [↑](#footnote-ref-3)
4. <https://www.amazon.com/> [↑](#footnote-ref-4)
5. <https://www.forbes.com/sites/louiscolumbus/2018/01/12/10-charts-that-will-change-your-perspective-on-artificial-intelligences-growth> [↑](#footnote-ref-5)
6. <https://bostondynamics.com/> [↑](#footnote-ref-6)
7. <https://chat.openai.com/> [↑](#footnote-ref-7)
8. <https://openai.com/> [↑](#footnote-ref-8)
9. <https://www.tesla.com/> [↑](#footnote-ref-9)
10. <https://labs.openai.com/> [↑](#footnote-ref-10)
11. <https://vpg.cs.princeton.edu/> [↑](#footnote-ref-11)