

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

Никола Карпић

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

дипломски рад

Бања Лука, април 2024.

**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 перформаци добијених модела.

Ментор: проф. др Зоран Ђурић

Кандидат: Никола Карпић (1144/14)

Бања Лука, април 2024.

Садржај

1. Увод	1
2. Машинско учење	4
2.1. Надгледано учење	6
2.1.1. Класификација	7
2.1.2. Регресија	8
2.2. Ненадгледано учење	9
2.2.1. Груписање	10
2.2.2. Смањење димензионалности.....	11
2.2.3. Учење уз подстицај	12
3. Процес машинског учења	14
3.1. Дефинисање проблема	14
3.2. Прикупљање података	15
3.3. Припрема података	15
3.4. Избор модела.....	16
3.5. Тренирање модела	16
3.6. Оцјењивање модела.....	17
3.7. Прилагођавање модела	17
3.8. Тестирање и примјена модела.....	18
4. Избор алгоритма	19
4.1. Логистичка регресија	19
4.2. Gaussian Naive Byes Classifier	20
4.3. K Nearest Neighbors Classifier	20
4.4. Decision Tree Classifier	20
4.5. Random Forest Classifier	21
4.6. Gradient Boosting Classifier	21
4.7. Support Vector Machine Classifier	22
4.8. LightGBM Classifier	22
4.9. K Nearest Neighbors Regressor	22
4.10. LightGBM Regressor	23
5. Практични рад	24
5.1. Прикупљање и припрема података	25
5.2. Избор и тренирање модела	28

5.3. Оцјењивање модела и резултати	28
6. Закључак	32
Литература	33

1. 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 често 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 неклассификован, појавила се потреба и за креирањем ефикасних алгоритама и процеса за разврставање, именовање и анализу тих података. Једна од најзаступљенијих области вјештачке интелигенције (*енг. Artificial Intelligence*) која се бави разврставањем, именовањем и анализом података је машинско учење (*енг. Machine Learning*).

Велике компаније као што су Google¹, Microsoft², Facebook³, Amazon⁴ и OpenAI⁵ су почеле да улажу огромне напоре и средства у своје могућности чувања и обраде 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).



Слика 1.1. Број научних радова на тему машинског учења у свијету од 2000. године⁶

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 област вјештачке интелигенције која 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

¹ <https://www.google.com/>

² <https://www.microsoft.com/>

³ <https://www.facebook.com/>

⁴ <https://www.amazon.com/>

⁵ <https://openai.com/>

⁶ <https://www.forbes.com/sites/louiscolombus/2018/01/12/10-charts-that-will-change-your-perspective-on-artificial-intelligences-growth>

предвидјели исходе. Како алгоритми уносе више података за учење, тако је могуће произвести све прецизније моделе засноване на тим подацима. [1]

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 (слика 1.2).



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

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.

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 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.

У овом раду се обрађује примјена 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 користимо су: логистичка регресија, *Gaussian Naive Bayes*, *K-Nearest Neighbors*, *Decision Tree*, *Random Forest*, *Gradient Boosting*, *Support Vector Machine*, *LightGBM*.

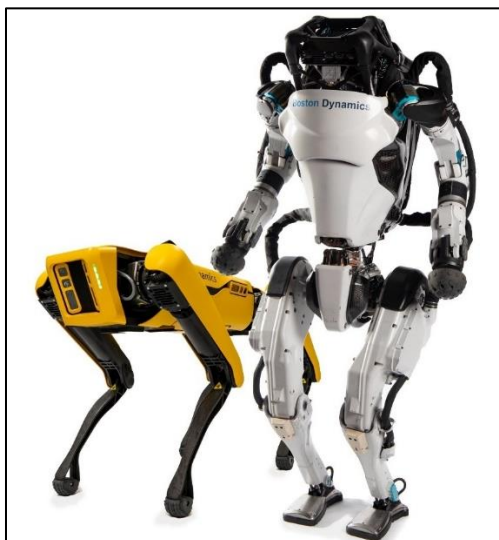
У другој глави се даје објашњење вјештачке интелигенције и уопштено објашњење метода машинског учења и процеса примјене алгоритама машинског учења. У трећој глави су детаљно објашњени кораци креирања модела машинског учења помоћу алгоритама машинског учења. У четвртој глави су детаљније описани алгоритми 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.

2. 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. године. Заједно са молекуларном биологијом, вјештачка интелигенција се редовно наводи као „поље у којем бих највише волио да будем“ од научника из других дисциплина. [3]

Вјештачка интелигенција тренутно обухвата огроман број области, као што су играње шаха, доказивање математичких теорема, писање поезије, возња аутомобила у препуној улици и дијагностиковање болести. Она може значајно унаприједити многе задатке за које је до сада сматрано да је потребна интелигенција. [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 Dynamics*⁷ (слика 2.1).



Слика 2.1. Роботски пас и човјеколики робот компаније *Boston Dynamics*

У последње вријеме, *ChatGPT*⁸ компаније *Open AI*⁹ и *Gemini*¹⁰ компаније *Google* су довели 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 човјечанству.

ChatGPT је изазвао велики број расправа о будућности и регулацији вјештачке интелигенције. Ово се догодило јер је широј јавности први пут дочаран прави потенцијал

⁷ <https://bostondynamics.com/>

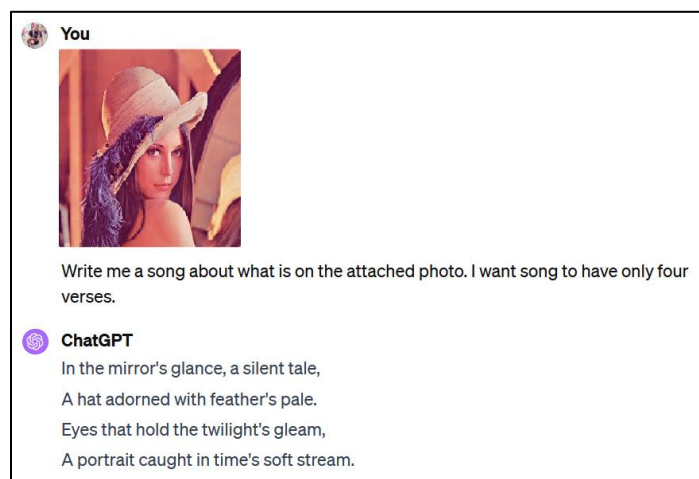
⁸ <https://chat.openai.com/>

⁹ <https://openai.com/>

¹⁰ <https://gemini.google.com/>

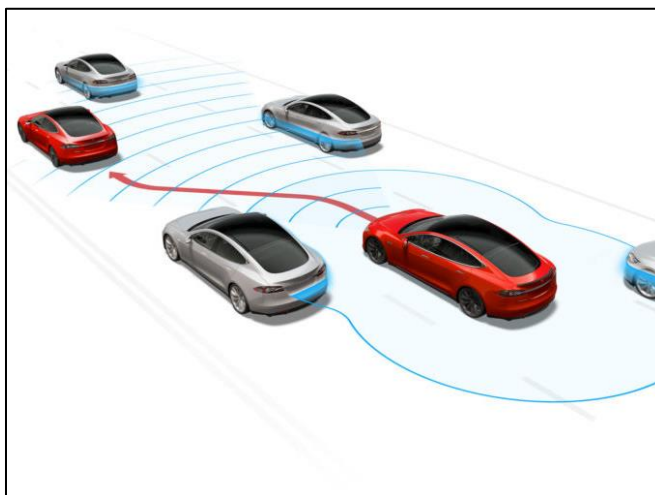
вјештачке интелигенције да „размишља“ и комуницира на начин који је до сада био могућ само у научној фантастици.

Његова могућност да генерише кохерентне и често импресивне текстове на основу корисничких упита и наредби (слика 2.2) отворила је многе етичке, филозофске и техничке расправе. Критичари су брзо скренули пажњу на потенцијалне опасности које вјештачка интелигенција може представљати, позивајући на опрез и строже регулативе.



Слика 2.2. ChatGPT препознаје фотографију и пише пјесму о њој

Неки од типичних примјера у којима се машинско учење показало као одговарајући алат су аутономна возила (слика 2.4), препоручена претрага код интернетских претраживача, добијање повратне информације о томе шта купци мисле о компанији, препознавање нежељених порука, откривање превара.



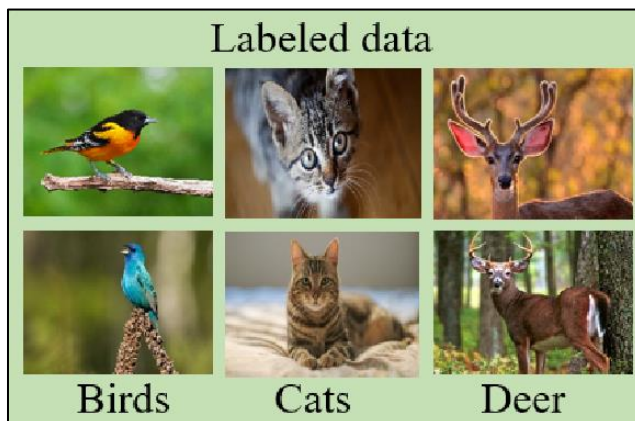
Слика 2.4. Приказ аутономне промјене возне траке возила компаније Tesla Motors ¹¹

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.

¹¹ <https://www.tesla.com/>

2.1. Надгледано учење

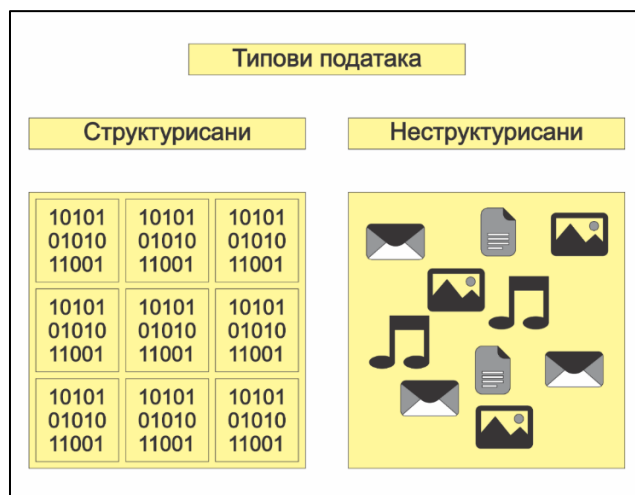
Надгледано учење (*енг. Supervised Learning*) је врста машинског учења у коме алгоритам проучава функцију (модел) која пресликава улазне податке (слика 2.5) на излазне како би научио функцију којом може да предвиди излазе за нове, непознате улазне податке. [4]



Слика 2.5. Примјер скупа улазних података

[3] 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.

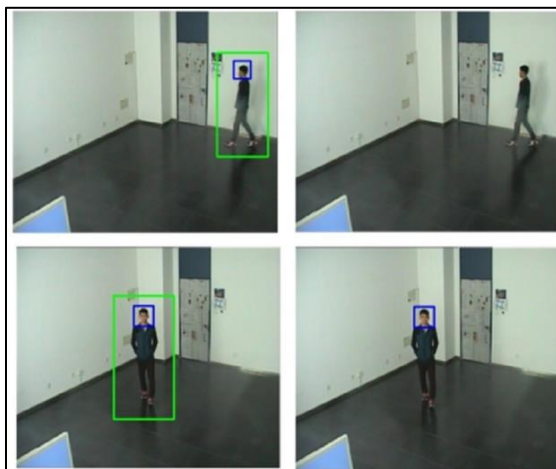
Пошто улаз модела чине подаци, они се могу јавити у структурисаном, неструктурисаном и полуструктурисаном облику (слика 2.6).



Слика 2.6. Примјер означеног скупа података

Структурирани подаци су сви подаци који се могу интерпретирати у неком фиксном формату (физичка мјерења, човјечанству, бројеви телефона, број особа, пол), док су неструктурирани подаци они подаци за које не постоји фиксна структура (разне текстуалне датотеке, фотографије, видео записи, звучни записи, веб странице, резултати претраживања на вебу).

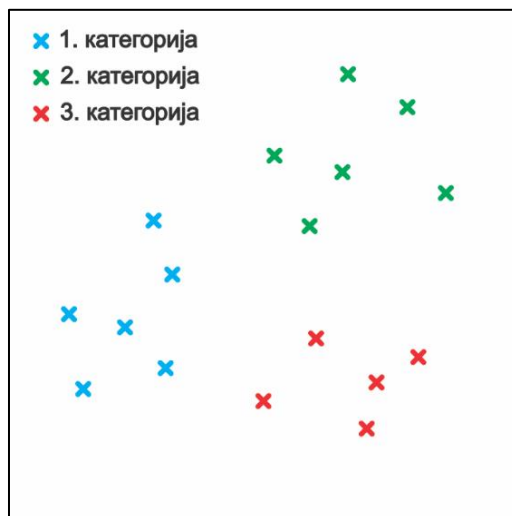
Надгледано учење је најчешћа врста машинског учења која се користи код проблема класификације, регресије и рангирања. Проблем откривања нежељене поште је један од најпознатијих примјера код којих се користи надгледано учење. [5] Надгледано машинско учење се често користи и у апликацијама попут кредитне процјене, медицинске дијагностике, временске прогнозе, препознавања лица (слика 2.7) и персонализације реклама. [6]



Слика 2.7. Препознавање лица у реалном времену [7]

2.1.1. Класификација

Класификација представља технику додјеливања категорије свакој ставци из скупа података (слика 2.8).



Слика 2.8. Примјер класификације у три класе

На примјер, класификација докумената састоји се од додјеливања категорија као што су политика, посао, спорт или вријеме сваком документу, док се класификација слика састоји од додјеливања 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. [5]

Постоје разне технике класификације, укључујући логистичку регресију, K-nn, 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.

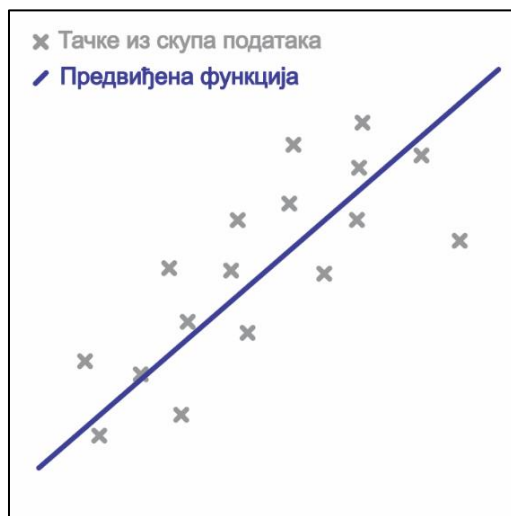
Класификација може бити надгледана или ненадгледана. У надгледаној класификацији, подаци садрже позитивне примјере за сваку категорију, док се у ненадгледаној класификацији тражи да алгоритам сам пронађе категорије, а може му се и спецификовати колико тачно категорија има у улазном скупу података. Да би класификација била квалитетно одрађена, потребно 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 (*енг. Accuracy*), прецизности (*енг. Precision*) и F-мјере (*енг. F-measure*). Овај процес помаже у одређивању колико добро алгоритам функционише у стварним примјенама и омогућава 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. [6]

2.1.2. Регресија

Регресија у машинском учењу представља технику коришћену за прогнозирање нумеричке вриједности на основу постојећих података (слика 2.9).



Слика 2.9. Примјер линеарне регресије

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 параметара.

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. [5]

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.

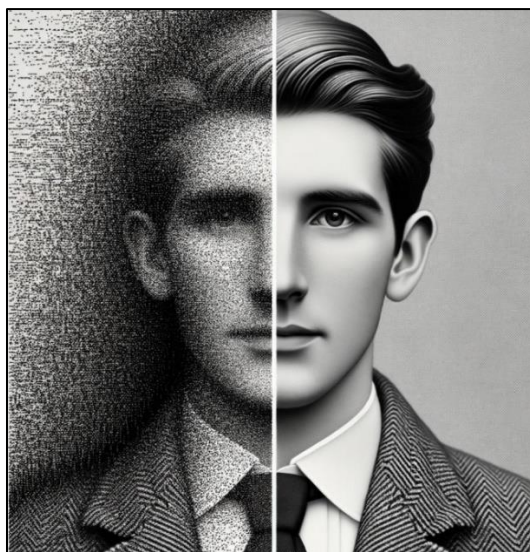
Постоји више врста регресионих модела, укључујући линеарну регресију, полиномијалну регресију и регресију са сигмоидном функцијом. Коришћењем различитих модела, може се доћи до различитих резултата, што захтијева одабир модела који најбоље одговара задатом проблему.

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 промјених.

2.2. 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 покушава сам пронаћи структуру у подацима. Груписање и смањење димензионалности су примјери проблема код којих се ненадгледано учење показало као најбоље. [4].

Груписањем се групишу слични подаци у исте групе како би се створиле категорије. Смањење димензионалности покушава смањити број димензија података како би се добила слика о главним вриједностима и структури података (слика 2.10).



Слика 2.10. Примјер уклањања шума са фотографије помоћу аутоенкодерске неуронске мреже генерисан помоћу DALE-E алата компаније OpenAI¹²

У ненадгледаном учењу, систем користи алгоритме попут кластерисања како би груписао сличне податке и створио категорије. Овакав приступ је користан када постоји

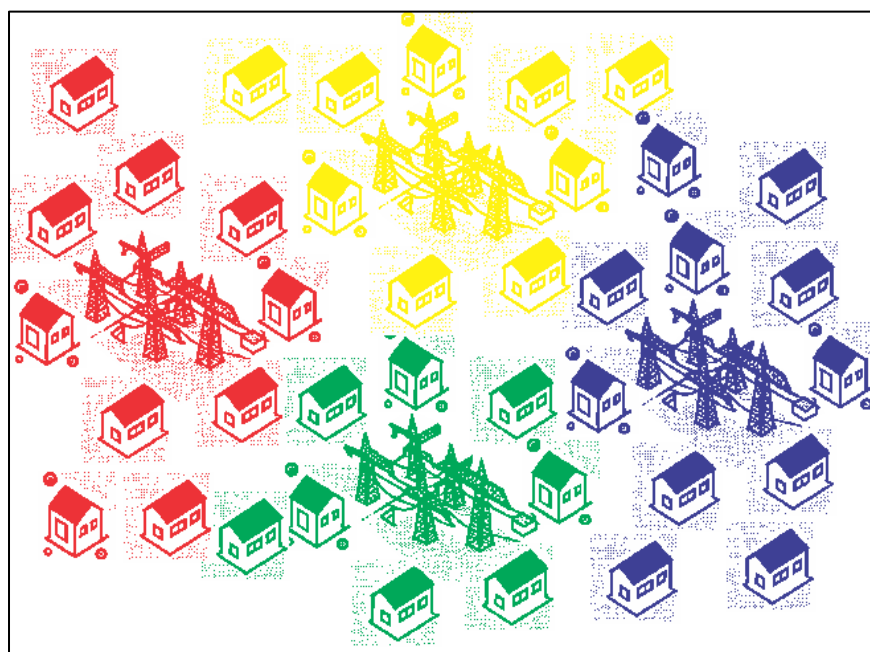
¹² <https://labs.openai.com/>

велики број података са сложеном структуром, а непознато је које су категорије важне за дати проблем.

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.

2.2.1. 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 (слика 2.11).



Слика 2.11. Једноставан примјер груписања домаћинстава према удаљености од електричног разводника

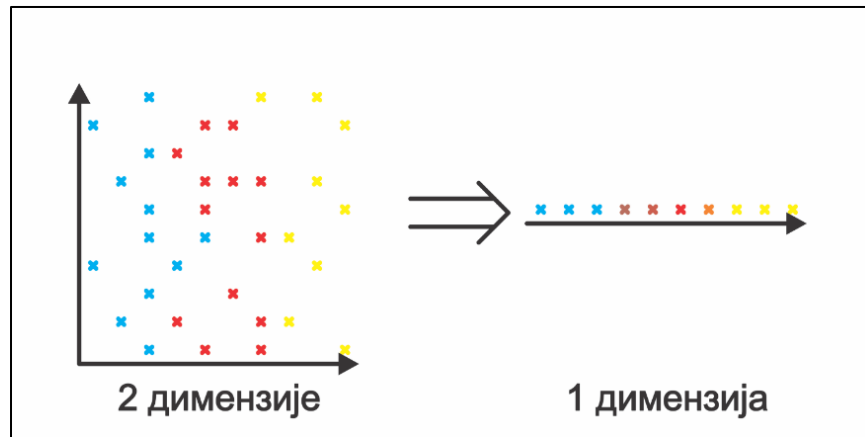
Груписање се описује као једна од метода за апстраховање информација из података без да се користе подаци о ознакама објеката. Груписање се може провести кориштењем различитих алгоритама, као што су хијерархијско груписање (енг. *Hierarchical clustering*), груписање методом к-средњих вриједности (енг. *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. [6]

2.2.2. Смањење димензионалности

Смањење димензионалности је техника у машинском учењу која се користи за смањивање броја атрибута у подацима. Смањење димензионалности се често користи у комбинацији са другим техникама машинског учења, како би се створили бољи модели (слика 2.12). Смањење димензионалности може да буде од помоћи и за избјегавање преприлагођавања.



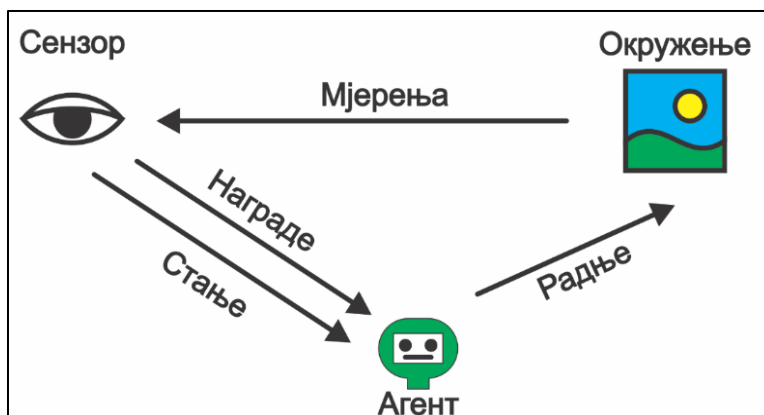
Слика 2.12. Примјер смањења димензионалности ненадгледаним учењем

Различите технике смањења димензионалности укључују анализу главних компоненти (енг. *Principal Component Analysis - PCA*), линеарну дискриминантну анализу (енг. *Linear Discriminant Analysis – LDA*) и мултидимензионално скалирање (енг. *Multidimensional Scaling - MDS*). Редукција димензионалности није рјешење за све проблеме у машинском учењу и треба се користити са опрезом како се не би изгубиле важне информације. [6]

Алгоритми који спадају у ову врсту учења су : К средњих вриједности, PCA (енг. *Principal Component Analysis*), t-SNE (енг. *T-Distributed Stochastic Neighbor Embedding*) и Правило удруживања (енг. *Association rule*).

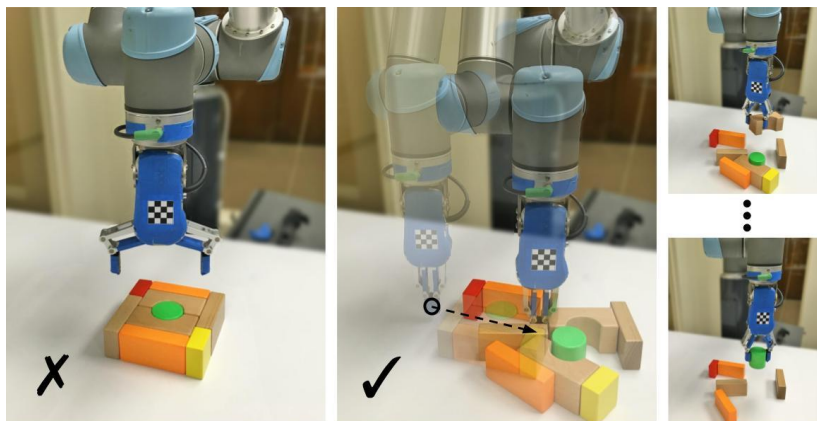
2.2.3. Учење уз подстицај

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 (слика 2.13).



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 (слика 2.14). [5]



Слика 2.14. Примјер роботске руке која учи уз подстицаје да подигне зелени предмет¹³

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.

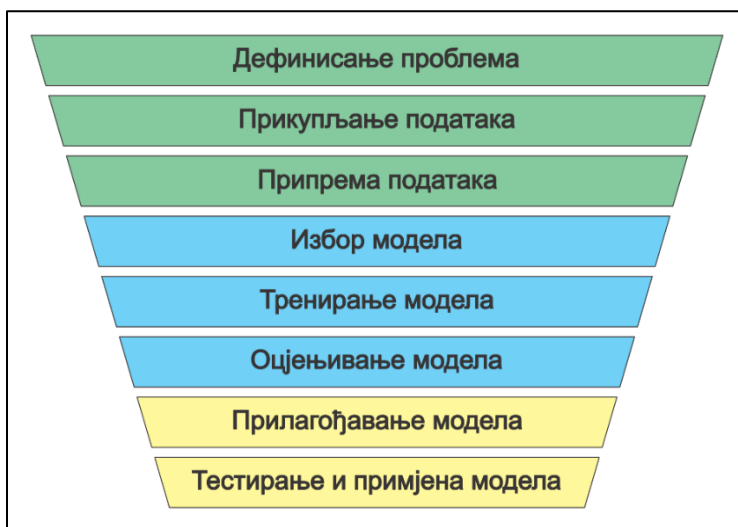
¹³ <https://vpg.cs.princeton.edu/>

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.

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. [6]

3. Процес машинског учења

Процес машинског учења (слика 3.1) почиње дефинисањем проблема. Послије тога, наставља се са прикупљањем података, који су најчешће велики број мјерења неке појаве (са великим бројем мјерених вриједности). Затим се припремају подаци тако што их се чисти, нормализује и претвара 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 продукционом окружењу како би се предвидјели нови случајеви на којима модел није трениран. [8]



Слика 3.1. Фазе процеса машинског учења

3.1. 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. [9]

3.2. Прикупљање података

Фаза прикупљања података у машинском учењу представља процес узимања података из различитих извора, њихову идентификацију и сакупљање у јединствену колекцију података који ће се користити за тренирање и тестирање машинских модела. [10] Подаци морају бити релевантни за проблем који се покушава ријешити и требали би бити квалитетни и служити као представник скупа података из стварног свијета. Велика пажња се посвећује прикупљању квалитетних података, јер лоши подаци доводе до лоших модела и неадекватних резултата.

Подаци могу бити прикупљени из једног извора података или из више извора (преузимањем, екстракцијом из база 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. [11].

3.3. Припрема података

Фаза припреме података 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 провјерио његову перформансу на другом скупу.

Без добро припремљених података, модел може бити неуспјешан у њиховој предикцији и генерализацији на нове податке. [12]

3.4. Избор модела

Избор модела у машинском учењу подразумијева одабир најприкладнијег алгорита и имплементацију тог алгорита 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 који су на располагању.

Избор модела се објашњава кроз неколико фаза. Прва ствар коју треба учинити је дефинисати проблем који се жели ријешити. [8] Ова дефиниција омогућава одабир адекватног типа модела. Након што се дефинише проблем, сљедећи корак је преглед релевантних модела и препорука од стране стручњака у пољу.

Након што се дефинише проблем и препоручи неколико модела, сљедећи корак је евалуација ових модела у складу са дефинисаним критеријумима, као што су прецизност, брзина тренирања и примјене, стабилност, робустност, интерпретабилност и сл. Сљедећи корак је стварни избор модела на основу испитивања, искуства или мјешавине та два приступа.

Погрешан избор модела може резултовати 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.

3.5. Тренирање модела

Тренирање модела у машинском учењу представља процес у коме се моделу дају подаци како би се научио како да предвиди резултате за нове примјере. Тренинг модела се може описати као процес оптимизације функције губитка (*енг. Loss Function*) или функције циља (*енг. Objective Function*). [8] Функција губитка одређује колико је добар модел у предвиђању, а функција циља се користи за оптимизацију модела у смислу да се максимизира та функција.

Процес тренирања се обично спроводи коришћењем алгорита 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 кроз које се тренира модел може варирати, а одлука о томе када прекинути тренирање зависи од многих фактора, као што су сложеност модела, величина скупа података, брзина оптимизације итд.

Процес тренирања модела није увијек једноставан и може захтијевати велике ресурсе, посебно ако се користе комплексни модели или велики скупови података, па је важно искористити га на одговарајући начин, како би се постигла адекватна предикција и генерализација модела на нове податке.

3.6. 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 (*енг. Cross-Validation*), провјеру са непознатим скупом података (*енг. Holdout Validation*) и процјену грешке (*енг. Error Estimation*). Ове процедуре оцјењивања користе се за одређивање прецизности модела, капацитета генерализације и сличних мјера квалитета.

Постоје различити критеријуми за оцјењивање модела, укључујући прецизност (*енг. Accuracy*), прецизност учења (*енг. Learning Accuracy*), прецизност генерализације (*енг. Generalization Accuracy*), F1 скор и ROC крива. 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 који се желе постићи, па је важно разумјети њихове предности и недостатке при одабиру метрика за оцјењивање модела, јер могу да утичу на перформансе модела.

3.7. 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.

На примјер, ако се модел обучава на скупу података са сликама мачака и паса, а потом се примјењује 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.

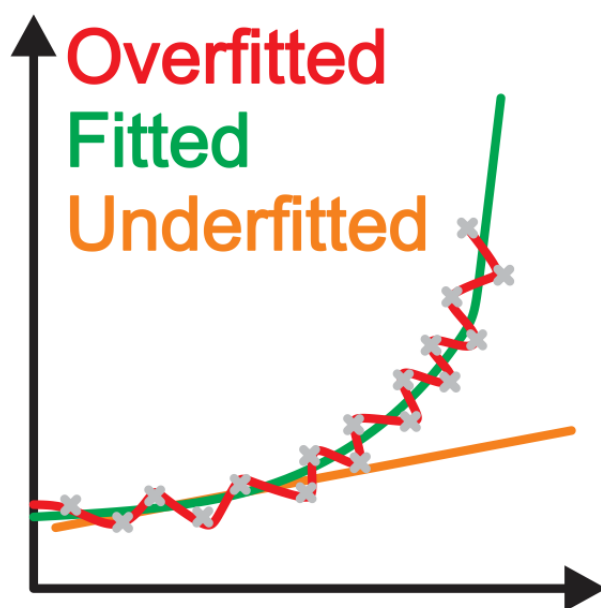
Постоје два основна начина за прилагођавање модела: надгледано и ненадгледано учење. Надгледано учење користи означене податке са жељеним излазима и користи их за изградњу модела који се потом оптимизује користећи алгоритме који подешавају параметре модела на најбољи начин. Ненадгледано учење користи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

Преприлагођавање (*енг. Overfitting*) у машинском учењу је проблем који настаје када модел претјерано добро учи детаље и шум из тренинг скупа до тачке гдје његова способност да генерализује нове податке постаје ограничена. То значи да модел има високу тачност на тренинг скупу али се показује лоше када се суочи са непознатим подацима. Преприлагођавање се често јавља када је модел прекомплексан у односу на количину и природу тренинг података, када садржи превише параметара који могу ухватити шум у подацима уместо стварних трендова.

Да би се избјегло преприлагођавање, истраживачи примењују различите технике као што су рано заустављање (гдје се тренинг прекида када се перформансе на валидационом скупу почну погоршавати), регуларизација (која додаје казнене термине у функцију губитка

да би се ограничила сложеност модела), и увећавање података (које умножава тренинг скуп кроз различите трансформације, повећавајући његову разноврсност и обим). Такође, коришћење крос-валидације може помоћи у процјени способности модела да генерализује на нове податке. Ове стратегије помажу у стварању робуснијих модела који су боље опремљени за решавање реалних проблема у различитим доменима.

Недовољно прилагођавање (енг. *Underfitting*) у машинском учењу је ситуација када модел не успијева адекватно да научи структуру или обрасце у тренинг подацима, што резултје лошим перформансама и на тренинг и на тест скупу података. Ово се обично дешава када је модел превише једноставан (са премало параметара) у односу на сложеност података, недовољно обучен, или када не постоји довољно података за обуку. Underfitting води ка томе да модел не може да генерализује нове податке ефикасно, што ограничава његову употребљивост у стварним применама.



Слика 3.2. Приказ преприлагођености и недовољне прилагођености

3.8. 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 генерализујући и способан радити на цијелом скупу података. Ако се модел покаже 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.

4. Избор алгоритма

Избор алгоритма у машинском учењу одређује квалитет модела и његову ефикасност. Различити фактори играју кључну улогу у избору алгоритма:

- Природа података: карактеристике података, као што су величина и врста, могу утицати на избор алгоритма. На примјер, ако су подаци дискретни, неки алгоритми као што су стабло одлучивања (*енг. Decision Tree*) могу бити погоднији од алгоритама заснованих на регресији.
- Циљ проблема: циљ проблема, као што је класификација, регресија или кластерисање, 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.

4.1. Логистичка регресија

Логистичка регресија (*енг. Logistic Regression*) је статистички алат који има за циљ да моделира биномни резултат с једном или више објашњивих промјењивих. У машинском учењу, овај алат често се користи за рјешавање проблема, у којима је циљ да се предвиди припадност објекта некој од двије класе. Основа алгоритма је линеарна регресија, али умјесто да се користи континуална зависна промјењива, логистичка регресија користи логистичку функцију (*енг. Logistic Function*) како би се моделирао однос између и бинарне зависне варијабле. Логистичка функција враћа излаз између 0 и 1 који се може интерпретирати као вјероватноћа припадности објекта једној од двије класе.

У процесу тренирања, алгоритам користи градијентни спуст (*енг. Gradient Descent*) 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. Loss Function*) и, користећи их, ажурирају се коефицијенти у логистичкој функцији.

Након што се модел тренира, може се користити за предвиђање припадности нових објеката једној од двије класе на основу вриједности њихових ознака. У овом случају, логистичка функција се користи за израчунавање вјероватноће припадности објекта некој класи, а коначна класификација се врши тако што се вјероватноћа пореди са неким прагом (*енг. Threshold*), 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 атрибута, величину скупа података и врсту проблема.

4.2. Gaussian Naive Byes Classifier

Gaussian Naive Byes Classifier (GNB) је алгоритам класификације базиран на Бајесовој теорему и претпоставци да свака карактеристика у подацима има нормалну расподелу. Алгоритам се користи за класификацију у којима постоји n -категорија и n -карактеристика за сваку инстанцу у подацима.

Процес функционисања GNB алгоритма почиње са рачунањем Gaussian Probability Density Function (PDF) за сваку карактеристику у свакој категорији. Овај PDF описује како се карактеристике расподељују у датој категорији. Након тога, Бајесова теорема се користи за рачунање вјероватноће за сваку категорију за дату инстанцу података. Вјероватноће се комбинују за све карактеристике у датој инстанци и на крају се израчунава коначна вјероватноћа за сваку категорију. Инстанца се класификује у категорију са највећом вјероватноћом.

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

GNB је једноставан и ефикасан алгоритам за класификацију, а посебно се добро показује у случајевима са великим бројем категорија и малим бројем карактеристика.

4.3. K Nearest Neighbors Classifier

K Nearest Neighbors Classifier (KNN) алгоритам је један од најједноставнијих и најчешће коришћених алгоритама у машинском учењу, коришћен за класификацију. Овај алгоритам користи концепт k -најближих сусједа за класификацију нових објеката.

Алгоритам KNN представља једноставну верзију алгоритма класификације, гдје се узима у обзир k -најближих сусједа објекта који се покушава класификовати.

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

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

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

4.4. Decision Tree Classifier

Decision Tree Classifier (DT) алгоритам се користи за разврставање података. Овај алгоритам користи стабло одлучивања као своју главну структуру, гдје се разматрају различити атрибути унутар скупа података и доносе се одлуке о томе како да се подаци разврстају.

Стабло одлучивања користи двије врсте чворова: чвор са члановима и лист. Чвор са члановима разматра неки атрибут и одлучује да ли подаци треба да иду у један од његових подчворова, који се затим поново разматрају. Овај процес тумачења се понавља све док се не дође до листа, који означава коначно мјесто код разврставања података.

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

Овај алгоритам је једноставан за разумијевање и имплементирање, што га чини погодним за многе примјене укључујући финансије, трговину, медицинске и друге индустрије. Међутим, постоје неки недостаци као што склоност преприлагођавању и потреба за одређивањем границе када се разматрају атрибути.

4.5. Random Forest Classifier

Random Forest Classifier (RF) је алгоритам који се користи и за класификацију и за регресију. Овај алгоритам користи много одвојених стабала одлучивања (насумичну шуму стабала одлучивања) 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 процјене, а затим се користи гласање да се добије финална процјена.

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 вјероватноће преприлагођавања (*енг. Overfitting*). Такође, овај алгоритам има боље перформансе у случајевима када има много атрибута, што често може довести до проблема у класичном стаблу одлучивања.

Коришћење RF 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 примјене у различитим областима.

4.6. Gradient Boosting Classifier

Gradient Boosting Classifier (GBT) 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. Користи технику бустинг (*енг. Boosting*) 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.

GB такође користи технику Gradient Descent за одређивање правца у ком ће се класификатори обучавати. Gradient Descent користи функцију губитка (*енг. Loss Function*) да би се утврдила удаљеност између тренутног стања и жељеног стања. Ова функција губитка се користи да би се пронашло оптимално рјешење, које ће смањити укупан број грешака.

Овај алгоритам обично даје боље резултате од других, једноставнијих класификатора, попут логистичке регресије или наивног Бајес класификатора. Међутим, GB је такође склон преприлагођавању, па је потребно пронаћи оптималну комбинацију броја класификатора и хиперпараметара.

4.7. Support Vector Machine Classifier

Support Vector Machine (SVM) Classifier је алгоритам машинског учења који користи математичке моделе како би предвидио категорију новог примјера. SVM користи теорију оптимизације да би формирао једну или више хиперсуперфиција (тј. линија или хиперраван) које најбоље раздвајају примјере у скупу података у одговарајуће категорије.

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

SVM има два главна мода рада: Линеарни SVM који користи линеарне хиперсуперфиције за раздвајање примјера у категорије, и нелинеарни SVM који користи нелинеарне хиперсуперфиције. Нелинеарни SVM се може користити у ситуацијама када линеарни SVM не може довољно добро да раздвоји примјере у категорије. Такође, подржава и технике регуларизације, што га чини погодним за рад са великим скуповима података и тешким проблемима класификације. Овај алгоритам се често користи за анализу текста, слика и геномских података.

4.8. LightGBM Classifier

LightGBM је алгоритам за машинско учење који се користи за класификацију и регресију. Развијен је као оптимизована верзија Gradient Boosting Machine (GBM) алгоритма, користећи технике које убрзавају процес тренинга. LightGBM користи технологију усмереног успоравања учења, што значи да се фокусира на тешке примјере у сваком кораку учења, како би се што брже извршио процес учења.

Он се разликује од стандардних GBM алгоритма по томе што користи хистограм за расподелу особина у сваком чвору. Овај приступ значајно смањује потребан број итерација у процесу учења, а такође, олакшава и убрзава приступ подацима. Такође, користи бинарно претраживање, што даље смањује вријеме потребно за извршење процеса учења. Овај алгоритам има способност учења на различитим типовима података, укључујући и велике податке и скупове података са спојеним подацима. Такође, он има добре перформансе на проблемима са небалансираним подацима, када има више примјера једне класе него друге.

LightGBM 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.

4.9. K Nearest Neighbors Regressor

K Nearest Neighbors Regressor алгоритам 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 функционише тако што се за сваку инстанцу у скупу података за учење проучава к најближих сусједа и израчунава се просјек вриједности промјењиве циља за те сусједе. На крају се добијени просјек 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 крос-валидација. Такође, укључивање тежина за

сусједи такође може утицати на квалитету предвиђања, гдје су сусједи који су ближе инстанци за предикцију тежи.

Међутим, овај алгоритам има и неколико недостатака, укључујући слабе перформансе у случајевима гдје постоји много димензија и висок степен интеракције међу промјенивима, што може резултовати проблемима с пренаучености.

KNN Regressor алгоритам је једноставан и брз за тренирање те често даје добре резултате у случајевима гдје су подаци хомогени и дистрибуција варијабле циља није превише сложена. Међутим, потребно је пажљиво одабрати параметре и провјерити перформансе модела на скупу података за тестирање како би се избјегли проблеми са пренаучењем.

4.10. LightGBM Regressor

LightGBM је врста регресијског алгоритма у машинском учењу који користи технику усмјерену према GB-у. Главна разлика између LightGBM и других алгоритама GB-а је у томе што користи двоструко бинарно подјелу за брзо рјешавање проблема скалирања.

У раду LightGBM-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 LightGBM-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.

LightGBM такође користи технику смањења димензионалности како би се избјегло превише сложености и преприлагођавања. То се постиже одабиром најзначајнијих атрибута које ће се користити за сваки модел.

LightGBM је популаран због своје брзине и ефикасности, посебно када се ради са великим скуповима података. Он је такође способан радити с различитим врстама података, укључујући и податке с различитим скалама и категоријске податке. LightGBM је један од најбољих регресионих алгоритама и користи се за предвиђање цијена, кредитне процјене и персонализацију препорука.

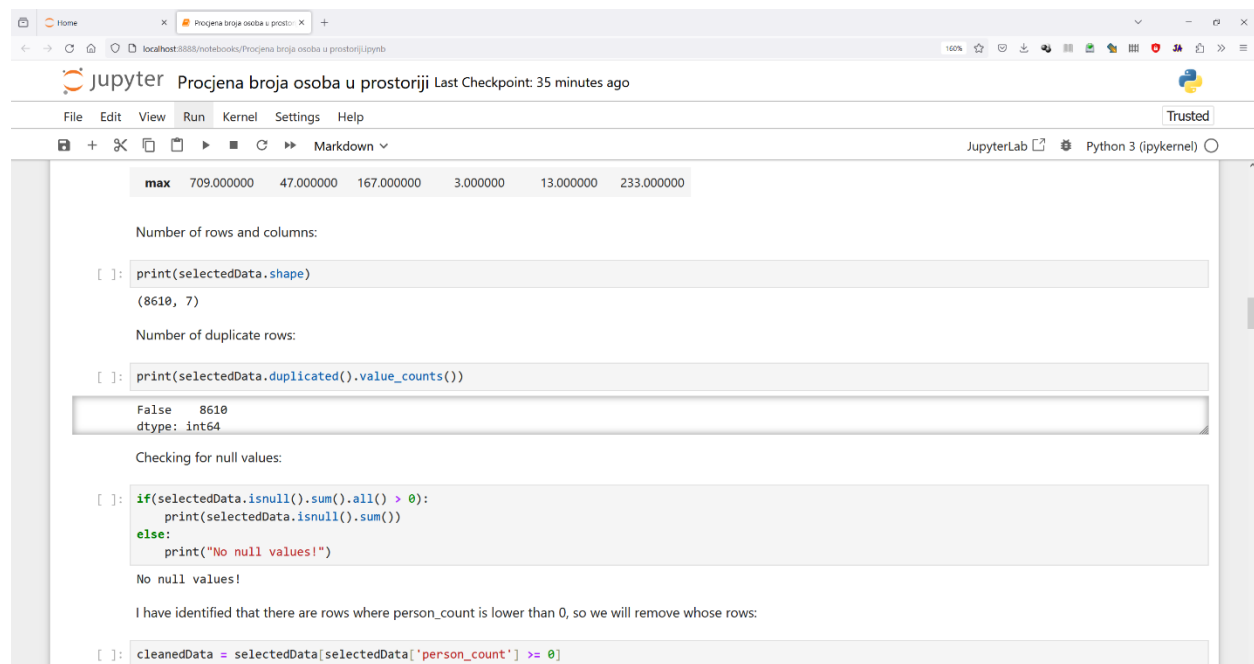
5. Практични рад

Задатак је да се анализирају модели за предвиђање броја особа у просторији креирани кориштењем неколико алгоритама (класификационих и регресионих). Моделе је потребно тренирати подацима који садрже температуру, влажност, ниво угљен-диоксида и др. За реализацију је кориштен Jupyter Notebook¹⁴ алат (који је приказан на слици 5.1.). Извршена је компаративна анализа 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. Карактеристике га способност комбиновања кода, текста, графике и формула у једном документу. [13]

Документ се састоји од ћелија које могу садржавати различите типове садржаја и најчешће се инсталира кроз дистрибуцију Anaconda¹⁵, која садржи многе научне библиотеке за Python. Поред класичног Jupyter 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 дијелење и презентацију резултата. [14]



Слика 5.1. Изглед радног окружења Jupyter Notebook алата

Библиотеке које су кориштене у практичном раду су: **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**

¹⁴ <https://jupyter.org/>

¹⁵ <https://anaconda.org/>

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.

У току развоја рјешења кориштени су сљедећи параметри: **THRESHOLD** праг за прорачун тачности модела, **DRAW** за избор да ли матрице конфузије треба да се исцртавају и **PRINT** за избор да ли желимо да исписујемо параметре на стандардни излаз.

Како проблем био ријешен што јасније, дефинисане су три помоћне функције: **calculate_accuracy** за рачунање тачности модела, **draw_confusion_matrix** исцртавање матрице конфузије и **fit_and_analyse** за тренирање и анализу модела.

5.1. Прикупљање и припрема података

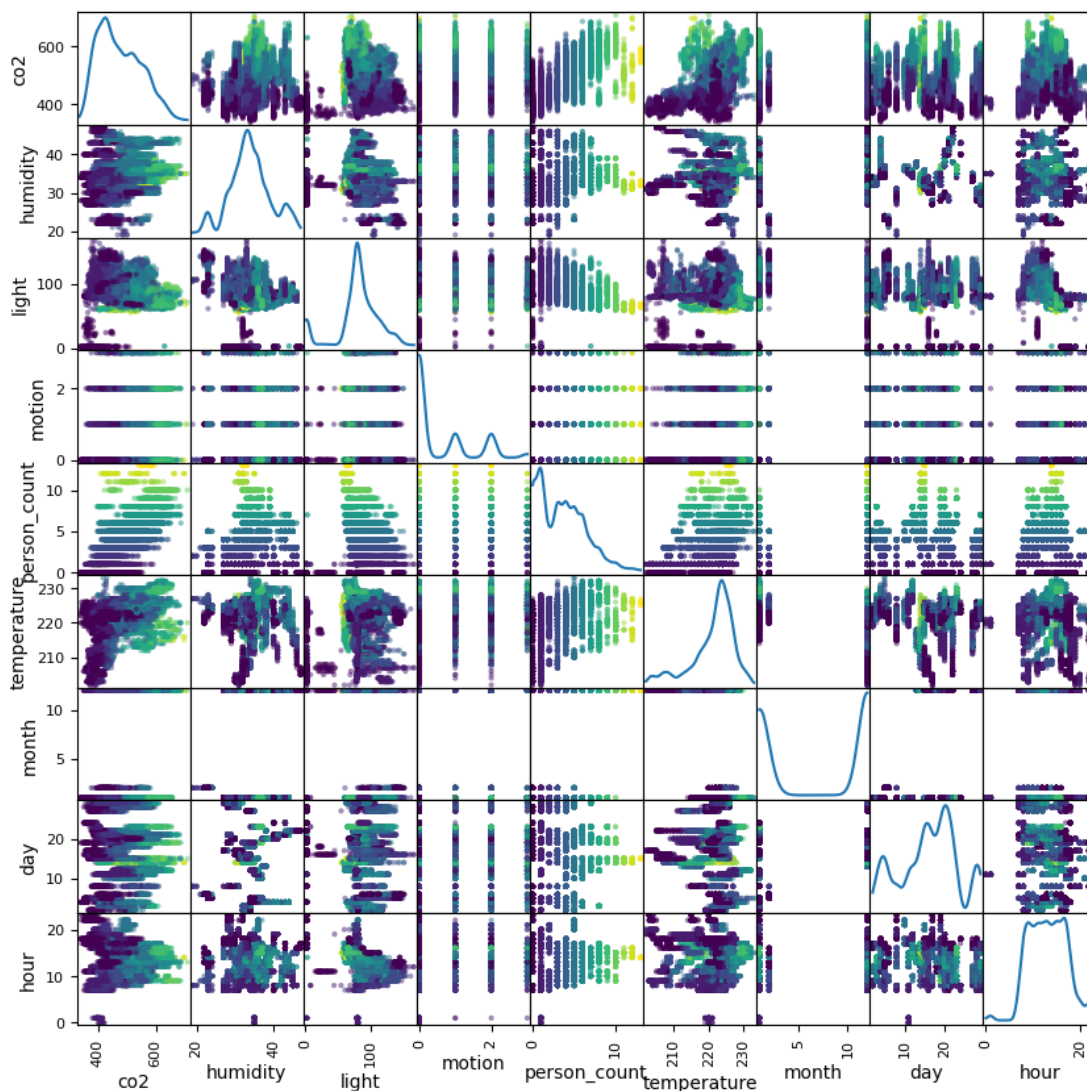
Скуп података је преузет са отвореног BitLab репозиторијума [15] у CSV формату. Колоне које садржи скуп података су: *name* - име, *time* - вријеме, *app_id* – идентификатор апликације, *battery* – ниво батерије уређаја, *co2* – ниво угљендиоксида у просторији, *dev_id* – идентификатор уређаја, *hardware_serial* – серијски број уређаја, *humidity* – влажност ваздуха, *light* – освијетљеност просторије, *motion* – количина помјерања у просторији, *person_count* – број особа у просторији, *temperature* – температура у просторији и *time_device* – вријеме/датум. Изабрано је сљедећих седам колона које ће бити кориштене за тренирање модела: **co2**, **humidity**, **light**, **motion**, **person_count**, **temperature** и **time_device** док су остале одбачене због тога што број особа у просторији не би требао имати никакав утицај на њихове вриједности.

Основне статистичке мјере (број, средње вриједности, стандардна девијација, минимум, максимум, квантили) су израчунате за сваку колону у скупу изабраних података. Исписан је облик (димензије) изабраног скупа података. Уклоњени су сви редови који имају **null** вриједности. Уклоњени су сви редови у којима је број особа у просторији мањи од нула.

Пошто датум и вријеме нису у одговарајућем формату, дефинисане су функције **convert_month**, **convert_day**, **convert_hour**, како би се претворила колона *time_device* из низа знакова у формат мјесец, дан и сат, пошто подаци о години, минутима и секундама нису од великог значаја за процјену броја особа у просторији. Функција је извршена једном над скупом података и нове вриједности датума и времена су спремљене у нове колоне **month**, **day** и **hour**.

Након овога, може се израчунати и приказати и матрица распршености скупа података, која је приказана на слици 5.2 Овај графикон омогућава визуелно поређење дистрибуција између атрибута, као и идентификацију евентуалних зависности или образаца међу атрибутима у скупу података.

Пошто нема смисла да се рачуна корелација колоне према самој себи, на главној дијагонали је кривуља дистрибуције података за поједине колоне, чија висина и облик показују густоћу расподеле података, што може бити корисно за разумијевање карактеристика података.



Слика 5.2. Приказ матрице распршености

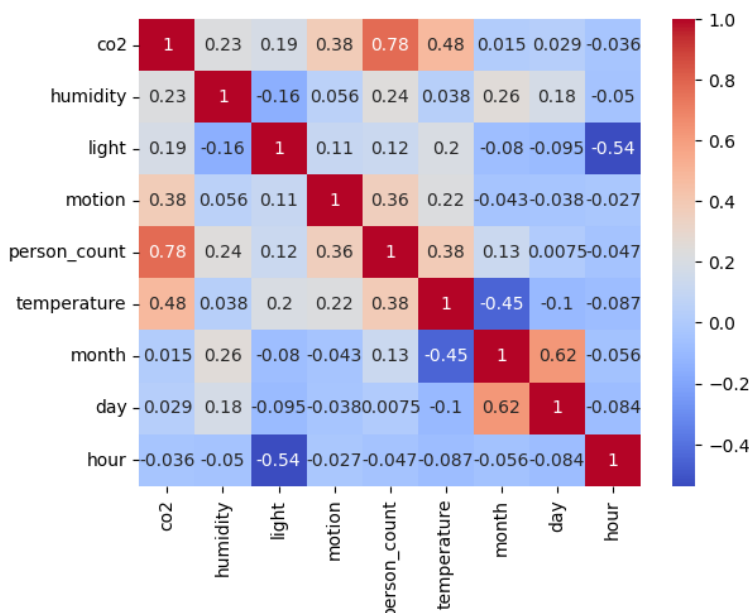
Матрица корелације је квадратна табела која приказује корелационе коефицијенте између различитих промјенљивих у скупу података. Корелациони коефицијенти мјере снагу и правац линеарне везе између двије нумеричке промјенљиве.

Вриједности у матрици корелације могу варирати од -1 до 1, гдје 1 означава савршену позитивну корелацију (како једна промјенљива расте, тако расте и друга), 0 означава да нема линеарне корелације између промјенљивих, а -1 означава савршену негативну корелацију (како једна промјенљива расте, друга опада).

Матрица корелације је корисна у статистичкој анализи за идентификацију односа између промјенљивих, што може помоћи у одабиру карактеристика за моделе машинског

учења, препознавању могућих узрока проблема са колинеарношћу у регресионим анализама и уопштено, за добијање увида у структуру података.

На слици 5.3. се може видјети матрица корелације која је израчуната над скупом података, како би се провјерио ниво зависности унутар самог скупа података.



Слика 5.3. Приказ матрице корелације

Закључци које се могу извући на основу ове матрице корелације су следећи. Постоји јака позитивна корелација између **co2** и **person_count** (0.78), што може указивати на то да већи број људи у просторији води до већег нивоа **co2**. **Temperature** и **co2** такође имају значајну позитивну корелацију (0.48), што може сугерисати да више нивои **co2** могу бити повезани са вишим температурама. **Month** и **temperature** имају умјерену негативну корелацију (-0.45), што указује на то да су ниже температуре повезане са одређеним мјесецима, тј. хладнијим добима године. Велика негативна корелација између **hour** и **light** (-0.54) може указивати на то да има мање свјетлости током одређених сати, вјероватно ноћу. Остале корелације у матрици су углавном слабе до умјерене, што значи да не постоји изразито јака линеарна зависност између тих промјенљивих.

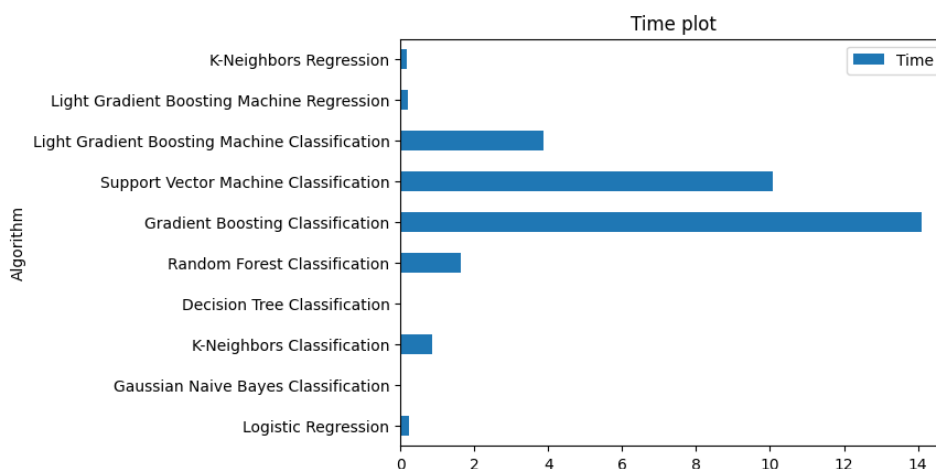
Скуп података је подијељен на подскупе тако што је прво издвојена колона за број особа из скупа података, а онда су креирани следећи скупови: **X** – скуп улазних података за тренирање модела, **X1** – скуп улазних података за тестирање и оцјењивање модела, **y_train** – скуп излазних података за тренирање модела, **y_test** – скуп излазних података за тестирање и оцјењивање модела

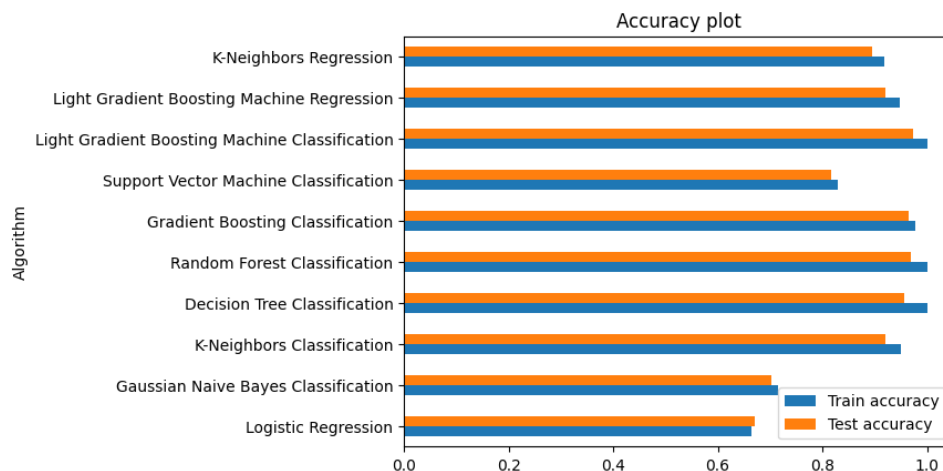
5.2. Избор и тренирање модела

5.3. Наизмјенично су бирани, креирани и записивани резултати за сљедеће алгоритме: LogisticRegression, GaussianNB, KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, SVC, LGBMClassifier, LGBMRegressor, KNeighborsRegressor. За сваки алгоритам су кориштени исти скупови података за тренирање и тестирање, како би компаративна анализа показала што вјеродостојније резултате. За тренирање је кориштена функција `fit_and_analise`, која је поред тога што је кориштена за тренирање, додатно мјерилаи исписивала вријеме извршавања и графички исписивала резултате тренирања, те карактеристике модела и скупова података. Оцјењивање модела и резултати

На крају, потребно је да приказати резултате и перформансе свих модела кориштених за тренирање и тестирање. Приказани су табеларно помоћу Time plot дијаграма све моделе који су кориштени заједно са временом извршавања и тачношћу алгоритма. Овај дијаграм помаже да се визуелно упоређи брзина извршавања сваког модела и идентификују модели који су релативно спори у односу на остале.

Ассурасу plot дијаграм приказује тачност сваког модела на скуповима за тренирање и тестирање.





Слика 5.4. Графички приказ резултата тренирања

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

Logistic Regression је брз модел с временом извршавања од само 0.226 секунди, али, са тачношћу тестирања од 67%, не пружа довољну прецизност за неке захтјевније примјене. Слично томе, Gaussian Naive Bayes Classification је још бржи, али и даље пружа малу тачност од 70%.

Gradient Boosting Classification и Support Vector Machine Classification су спорији модели али пружају врло добре резултате тачности тестирања од 81% и 97%.

Модел попут K-Neighbors Classification, Decision Tree Classification и Light Gradient Boosting Machine Classification комбинују брзину с високом тачношћу, прелазећи 90% на тестирању. Међутим, савршена тачност тренирања на скупу података за тренирање код Decision Tree и Light Gradient Boosting Machine Classification алгоритама може указивати на потенцијално преприлагођавање, што значи да модел може неадекватно генерализовати на невидјеним подацима.

Иако и други алгоритми пружају снажне перформансе, Random Forest Classification се истиче као најпрецизнији са изузетном тачношћу тестирања од 97%. Иако није алгоритам који се извршава најбрже, вријеме извршавања од око 1.6 секунди је прихватљиво за потребе мјерења броја особа у просторији, пошто ће се модел само једном тренирати. Његова савршена тачност тренирања на скупу података за тренирање такође указује на могућност преприлагођавања.

Матрица конфузије (слика 5.4) је табела која се користи за оцјену перформанси модела машинског учења, нарочито у задацима класификације. Она је посебно корисна за визуализацију перформанси када су класе дисбалансиране. Матрица показује број исправних и погрешних предвиђања у односу на стварне класе, односно тачне вриједности. Структура матрице конфузије за бинарну класификацију обично изгледа као на слици 5.4.

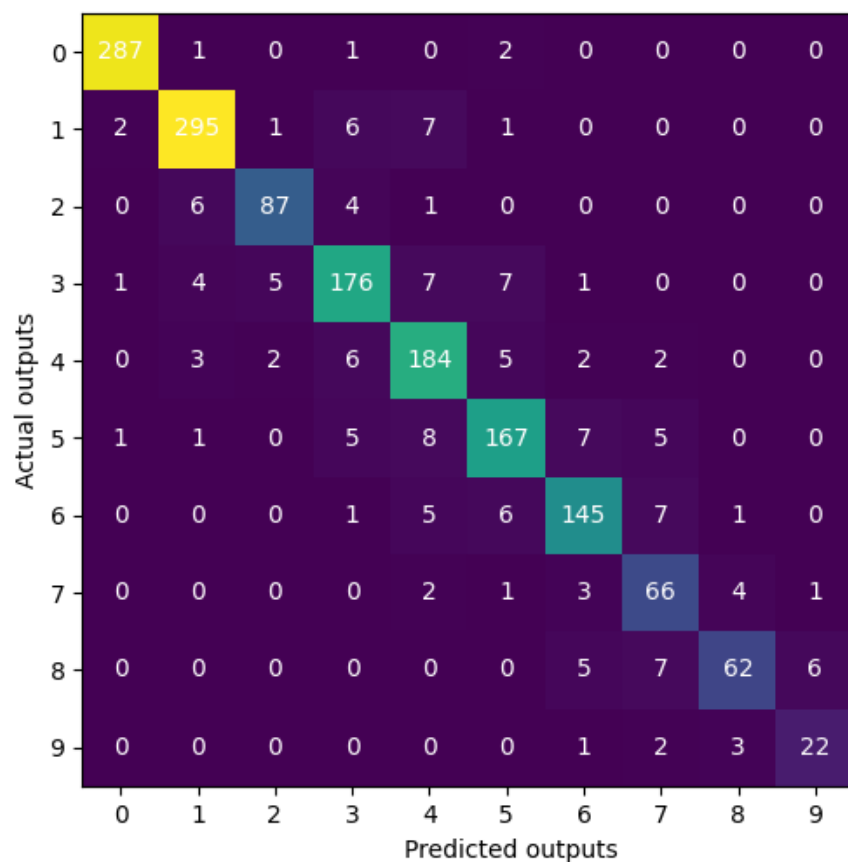
		ПРЕДВИЂЕНИ	
		ПОЗИТИВНИ	НЕГАТИВНИ
СТВАРНИ	ПОЗИТИВНИ	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
	НЕГАТИВНИ	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)

Слика 5.5. Графички приказ матрице конфузије

На основу матрице конфузије на слици 5.5, могу се уочити високе вриједности на главној дијагонали. Прво, модел изузетно добро класификује примјере који припадају класи **co2**, што се види из великог броја тачних предвиђања. Исто тако, класе **humidity**, **light**, **motion** и **person_count** такође имају релативно висок број тачно класификованих примјерака. Ово упућује на то да модел добро хвата карактеристике које су релевантне за ове класе.

Са друге стране, класе као што су **month**, **day** и **hour** имају мањи број тачно класификованих примјерака, што може указивати на то да моделу можда недостају релевантне карактеристике или је потребно додатно усавршавање да би боље разликовао временске периоде.

Модел показује извјесну способност да разликује **temperature**, али се суочава са изазовима у класификацији **month**, **day** и **hour**, што може указивати на сложеност у обради и класификацији временских података. Ово може бити због везе између времена и температуре која захтева софистицираније методе за детекцију узорака.



Слика 5.6. Графички приказ матрице конфузије Random Forest Classification модела

Иако модел има добру основу за препознавање неких класа, оптимизација модела и фино подешавање, као и примјена напредних техника за обраду временских серија, могли би бити потребни да би се побољшала прецизност и свеобухватност у класификацији комплекснијих шаблона као што су временски подаци.

6. 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 предвиђања текста, већ су и отвориле врата за револуционарне могућности у интеракцији са рачунарима и обради природног језика (OpenAI ChatGPT и Google Gemini).

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 би 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 продукционом окружењу, пружајући тачна и брза предвиђања броја особа у просторији. И док сваки модел има своје предности и недостатке, 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 Intelligence A Modern Approach, New Jersey: Pearson Education, Inc., 2010.
- [4] C. M. Bishop, Pattern Recognition and Machine Learning, New York, USA: Springer, 2006.
- [5] M. Mohri, A. Rostamizadeh и A. Talwalkar, Foundations of machine learning, Cambridge, Massachusetts: The MIT Press, 2018.
- [6] T. Mitchell, Machine Learning, New York: McGraw-Hill, 1997.
- [7] M. Weidong, F. Mengdan, L. Jing и H. Quing, Real-time face recognition based on pre-identification and multi-scale classification, London, England: The Institution of Engineering and Technology, 2019.
- [8] E. Alpaydin, Introduction to Machine Learning, Cambridge, Massachusetts: The MIT Press, 2010.
- [9] G. James, D. Witten, T. Hastie и R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, New York City, USA: Springer, 2017.
- [10] E. Amelsen, Building Machine Learning Powered Applications: Going from Idea to Product, Sebastopol, CA: O'Reilly Media, Inc., 2020.
- [11] I. H. Witten, E. Frank и M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Burlington, USA: Morgan Kaufmann, 2011.
- [12] T. Fawcett и F. Provost, Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, Sebastopol, CA: O'Reilly Media, Inc., 2013.
- [13] D. Toomey, Learning Jupyter 5: Explore Interactive Computing Using Python, Java, and More, Birmingham, UK: Packt Publishing, 2018.
- [14] J. Daniel, Data Science at Scale with Python and Dask, Shelter Island, New York: Manning, 2019.
- [15] N. Bockstael и A. Jadin, CO2 based room occupancy detection : an IoT and machine learning application, Louvain-la-Neuve: Université catholique de Louvain, 2018.