

**Universiteti i Prishtinës**

Fakulteti i Inxhinierise Elektrike dhe Kompjuterike

Departamenti:Inxhinieri Kompjuterike

***Dizajni dhe Analiza e Algoritmeve***

**Decision Trees**

**[Pemët Vendimarrëse]**

**Profesor**:Prof. Avni Rexhepi

**Asistent**: Dardan Shabani

**Prishtinë***, maj* *2021*

* **Menaxhimi i Dokumentit**

**Ky dokument është punuar nga:**

Natyra Vitija

Tahir Temaj

Teuta Ukshini

**Për krijimin e dokumentit janë përdorur veglat:**

Microsoft Word

Google Docs

Python

Pycharm

Jupyter Notebook

Visual Studio Code

Përmbajtja

[1. Hyrje......................................................................................................................................4](#_Toc46605562)

[2. Qellimi...................................................................................................................................4](#_Toc46605563)

[3. Decision Trees.....................................................................................................................5](#_Toc46605564)

[3.1.Algoritmat për ndërtimin e Decion Tree....................................................................6](#_Toc46605566)

3.1.1.[ID3 Algoritmi...................................................................................................7](#_Toc46605567)

[3.2. Metrikat për zgjedhjen e Atributeve.........................................................................7](#_Toc46605568)

[3.2.1.Entropy...........................................................................................................7](#_Toc46605569)

[3.2.2 Gini Index...................................................................................................... 8](#_Toc46605570)

[3.2.3 Information Gain............................................................................................ 8](#_Toc46605571)

[4. Implementimi i Decision Tree në Python...........................................................................9](#_Toc46605574)

4.1 Paraqitja e Decision Tree për modelin e dhënë........................................................13

**5. Implementimi i Decision Tree ne aplikacion.......................................................................15**

5.1 Startimi i aplikacionit................................................................................................17

6. Referencat…………………………………………………………………………………………………………………………………19

**1.Hyrje**

Probleme të shumta të ditëve të sotme kanë zgjidhje në lëminë e Data Analytics dhe Machine Learning.

Machine Learning përdoret kudo nga automatizimi i detyrave të zakonshme deri te ofrimi i njohurive inteligjente.Kryesisht aplikohet në zgjidhjen e problemeve për të cilat nuk ekzistojnë zgjidhje tradicionale. Në shumicën e rasteve zgjidhja përmes aplikimit të algoritmeve të Machine Learning rezulton në kod me qartësi dhe performancë më të lartë.   
Për shembull, një gjurmues i fitnesit që mund të vishet si Fitbit, ose një ndihmës inteligjent i shtëpisë si Google Home.1

Ka shumë më tepër shembuj të ML në përdorim:

* Parashikimi,
* Njohja e figurës,
* Njohja e të folurit,
* Diagnostifikimet mjekësore,
* Industria financiare dhe tregtimi.

Ideja gjenerale e shumicës së algoritmeve të Machine Learning është që të ndërtohet një model matematikor nga një mostër e të dhënave e pastaj të shfrytëzohet ky model për bërjen e parashikimeve.

Një ndër algoritmet më të përdorura në lëminë e Machine Learning është Decision Tree.

Decision Tree është një nga algoritmet më të lehta dhe të njohura të klasifikimit për të kuptuar dhe interpretuar.1

**2. Qëllimi**

Ky projekt është dizajnuar për lënden Dizajni dhe Analiza e Algoritmeve me temën Decision Tree dhe ka për qëllim përdorimin e algoritmit Decision Tree në zgjidhjen e një problemi specifik nga jeta e përditshme. Decision Tree është krijimi i një modeli që do të parashikojë vlerën e një synimi bazuar në ndryshoret e dhëna.2

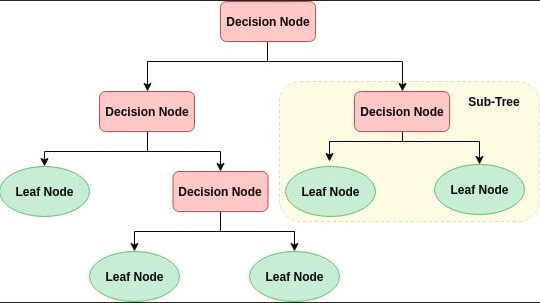
**3.Decision Trees**

Decision Tree përdoren për të përfaqësuar vizualisht vendimet dhe për të treguar ose informuar vendimmarrjen. Kur punoni me machine learning dhe data mining, decision trees përdoren si një model parashikues. Këto modele hartojnë vëzhgimet rreth të dhënave deri në përfundime rreth vlerës së synuar të të dhënave. Në modelin parashikues, atributet e të dhënave që përcaktohen përmes vëzhgimit përfaqësohen nga degët, ndërsa konkluzionet rreth vlerës së synuar të të dhënave përfaqësohen në fletë.2



Fig 1. Paraqitja e Decision Tree

Çdo nyje në pemë vepron si një rast prove për disa atribute, dhe secila skaj që zbret nga ajo nyje korrespondon me njërën nga përgjigjet e mundshme për rastin e provës. Ky proces është rekursiv dhe përsëritet për çdo nënpemë të rrënjosur në nyjet e reja.



Branch

Root Node

Fig 2. Struktura e Decision Tree

1. **Nyja rrënjë(Root node):** Nyja kryesore që edhe ka korrelacion më së shumti me atributin target. Kjo mund të ndahen në më shumë nyje tjera.
2. **Splitting:** Është proces i ndarjes së një nyjeje në dy apo më shumë nën-nyje.
3. **Nyja e vendimit(Decision node):** Kur një nën-nyje ndahet edhe më tej në nën-nyje atëherë quhet Decision node.
4. **Nyja Gjethe(Leaf/Terminal node**): Nyjet që nuk ndahen(split) më tutje.
5. **Pruning:** Kur zvogëlojmë madhësinë e pemëve të vendimit duke hequr nyjet (përballë Splitting), procesi quhet pruning.
6. **Dega / Nënpema:** Një nën-seksion i pemës së vendimit quhet degë ose nën-pemë.
7. **Nyja Prind dhe Fëmijë(Parent** **and Child node):** Një nyje që ndahet në nën-nyje quhet Nyje prind e nën-nyjeve ndërsa nën-nyjet janë fëmijët e nyjës prind.

Në pemën e thjeshtuar të vendimit më lart, një shembull klasifikohet duke e renditur atë përmes pemës në nyjën e duhur të gjethes. Në bazë të krahasimit, ne ndjekim degën që korrespondon me atë vlerë dhe kalojmë në nyjen tjetër.3

**3.1 Algoritmat për ndërtimin e Decision Trees**

Disa nga algoritmat që përdoren për ndërtimin e Decision Tree janë:  
  
**CHAID** (Chi-square automatic interaction detection) → Është algoritmi më i vjetër i Decision tree. Chi-square tek akronimi i këtij algoritmi është një metrikë që matë rëndësinë që ka një atribut për atributin target.5

**CART**(Classification And Regression Trees) → përdoret me shumë kur të dhënat janë kontinuale dhe tek Regresioni.

**ID3**(Iterative Dichotomiser 3) → zgjerim i D3. Ne vazhdim do të përdorim këtë algoritëm për të ndërtuar pemën.  
  
**CHAID** përdor **chi-square** për zgjedhjen e atributeve që të vendosen në decision tree.  
**CART** e përdor **Gini index** dhe **ID3** përdor **Information Gain.**  
**ID3** algoritmi është shumë I ngjajshëm me **CART** vetëm se thamë që ID3 përdor Information Gain që faktikisht llogaritet nga Gini Index.4

**3.1.1 ID3 Algoritmi**

ID3 algoritmi ndërton pemën duke përdor lartë-poshtë greedy algoritmin.  
Algoritmi greedy(lakmitar) gjithmonë e bënë zgjidhjen që duket të jetë më e mira për momentin.

Hapat e ID3 algoritmit:

1. Fillon me të dhënat trajnuese që në fillim të gjitha atributet e tyre konsiderohen si nyje rrënjë(root node) S.
2. Në secilin iterim(përsëritje) të algoritmit, iteron nëpër secilin atribut jo të përdorur(S) më parë dhe llogarit Gini Index(G) dhe Information Gain(IG) të atij atributi.
3. Pastaj zgjedh atributin me Gini Index-in më të vogël apo IG më të lartë.
4. Atributi i zgjedhur ndahet për të formuar nyjet fëmijë(child nodes).
5. Algoritmi përsëritet derisa të kalon nëpër të gjitha atributet dhe të bëhet vendosja e tyre në pemë.4

**3.2 Metrikat për zgjedhjen e Atributeve**

Nëse një dataset përbëhet prej N atributeve atëherë vendimi se cilin atribut ta vendosim në nyjen rrënjë dhe nyjet tjera është një proces i komplikuar. Vendosja e një atributi të rastësishëm nuk e zgjedh problemin.   
Metrikat posaqërisht të formuluara për zgjedhjen e atributeve dhe vendosjen e tyre në pemë janë:

* Chi-Square
* Entropy
* Gini Index
* Information Gain

**3.2.1 Entropia**

Entropia është metrikë që llogarit se sa informata na jep një atribut në dataset në lidhje me atributin target.

Sa më e lartë entropia aq më e vështirë është me nxjerr konkludime nga ai atribut.

Entropia e një atributi matematikisht shprehet si:

ku S – atributi i tanishëm, pi – probabiliteti i një ngjarje i.4

**3.2.2 Gini Index**

Gini Index është metrikë që ka të njëjtin qëllim sikurse Entropia. Vetëm se Gini Index preferohet me shumë kur kemi të bëjmë me të dhëna kategorike(diskrete) e ndërsa Entropia preferohet për të dy llojet e të dhënave(për shkak të formulës që përmban funksionin logaritmik përdoret edhe për të dhëna kontinuale)  
  
Gini Index matematikisht shprehet:

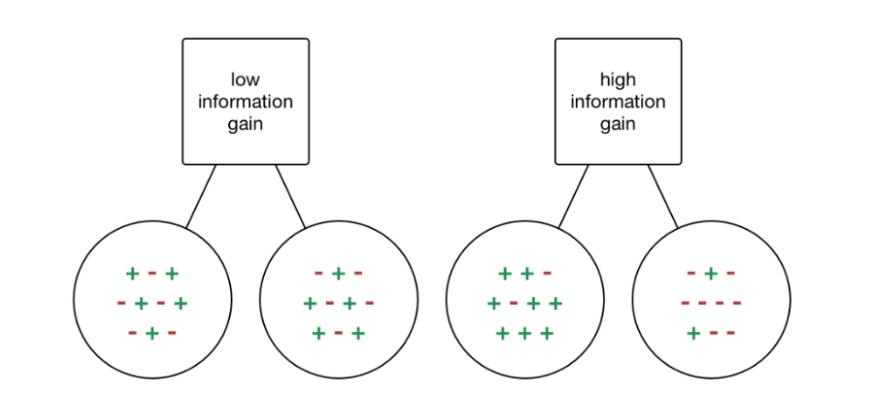
Mesatarja e Gini Index tek binary split apo edhe tek multiarray split llogaritet si mesatare e Gini-t të secilit element të atributit që është bërë split.

Pasi që atributi target me të dhënat që do të ndertohet pema është kategorik(me dy vlera) dhe për shkak të llogaritjeve më lë lehta do të përdorim Gini Index për llogaritje.4

**3.2.3 Information Gain**

Information Gain është një lloj statistike që mat se sa mirë një atribut i caktuar i ndan të dhënat trajnuese sipas atributit target.

Krejt esenca tek ndërtimi i pemës është që të zgjedhen atributet me Information Gain më të lartë dhe Gini index më të ulët.5

  
Fig 3. Information Gain

Information Gain shprehet si:

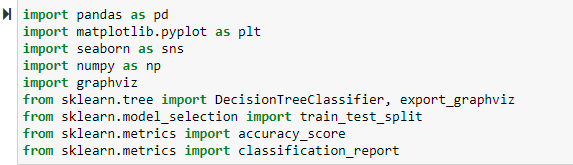
**4.Implementimi i Decision Tree në Python**

Për implementimin e Deciosion tree në Python janë përdorur të dhëna rreth medikamenteve mjekësore (barnave), të dhënat (dataset) janë marrur nga webfaqja *Kaggle* dhe ka për qëllim të parashikohet cilat barnat përdoren për moshën e caktuar, të emërtuara DrugA,DrugB,DrugC,DrugX apo DrugY.

Dataset-i përbëhet nga disa atribute të predikimit mjekësor dhe një atribut i synuar(target).

Atributet përfshijnë: Moshën,Gjininë,Presioni i gjakut,Kolesteroli,Natriumi në Kalium. Nuk do të diskutohet për pjesën e pastrimit dhe pre-procesimit e te dhënave pasi që kjo pjesë i përket lëmisë se Machine Learning mirë po të kalohet drejt e tek aplikimi i algoritmit në të dhëna.

Importimi i paketave të nevojshme për ekzekutim.



Leximi i të dhënave (dataset) dhe 5 rreshtat kryesorë



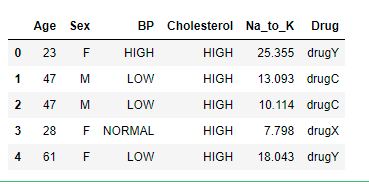


Fig 4. 5 rekordet e para me të dhëna

Në figurë janë paraqitur të dhënat që janë të gatshme per aplikimin të algoritmit.

Këtu bëhet kontrollimi i marrëdhënieve midis barnave dhe dhe moshës së pacientit.

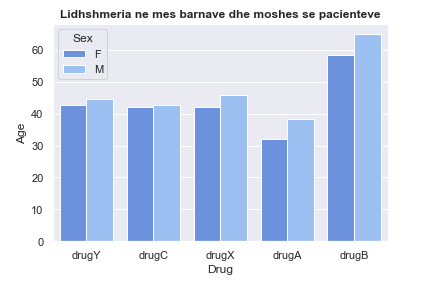


Fig 5.Lidhshmëria në mes barnave dhe moshës së pacienteve

Këtu bëhet krahasimi i përdorimit të barnave me të shfrytezuara.

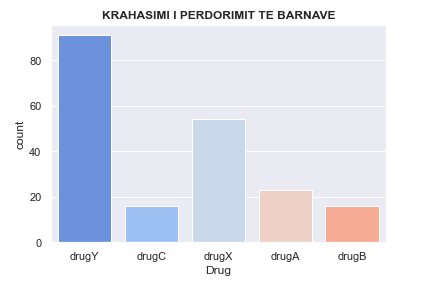


Fig 6. Krahasimi i perdorimit te barnave

Në projekt kemi bërë edhe zëvendësimin e të dhënave në mënyrë që të lehtësohet përdorimi i të dhënave

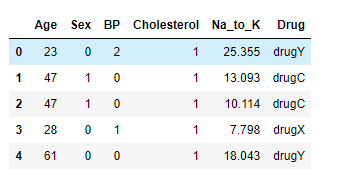
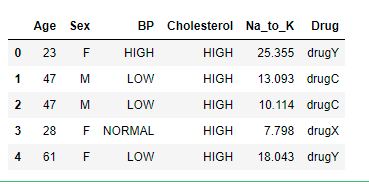


Fig 7. Zëvendësimi i të dhënave

Para se të aplikohet algoritmi së pari ndahen atributet në ato predikuese dhe në atributin e synuar.

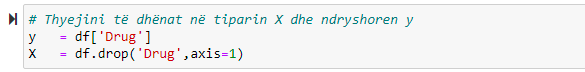


Fig 8. Ndarja e te dhenave



Fig 9. Ndarja ne Train dhe Test

Këtu tek parametrat e funksionit **train\_test\_split**, **test\_size** e kemi caktu **0.25** që do të thotë se **25%** të të dhënave do të jenë të dhëna testuese e ndërsa pjesa tjetër(respektivisht **75%**) të të dhënave janë trajnuese që përdoren për trajnimin e modelit.

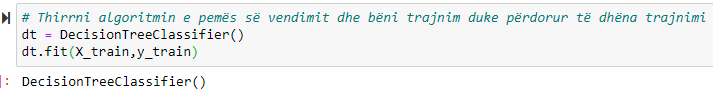


Fig 10. Fillimi i thirrjes se algoritmit

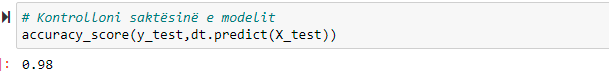


Fig 11. Saktesia e modelit

Shihet qe sakt*ë*sia finale e modelit është98%.

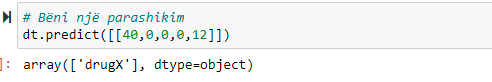


Fig 12.Predikimi

Këtu bëhet kodi për predikimin e një shembullin adekuat.

**4.1 Paraqitja e Decision Tree për modelin e dhënë**

Më anë të funksionit **export\_graphviz** dhe librarinë **pydotplus** për vizualizim është paraqitur pema vendimarrëse.

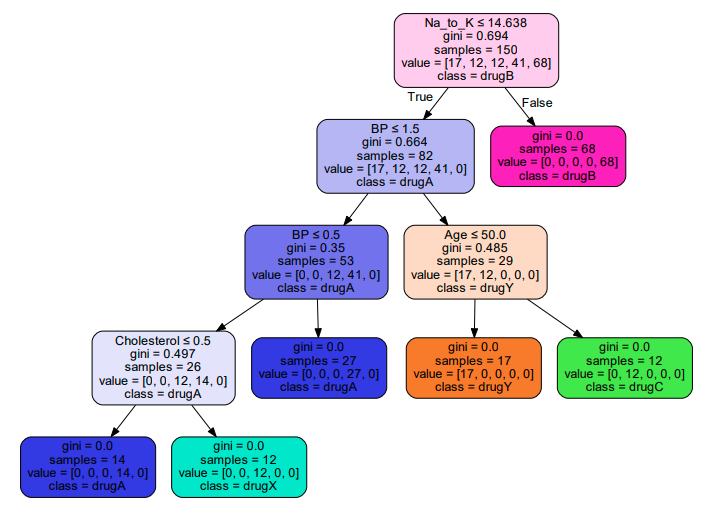


Fig. 13. Paraqitja e Decision Tree përmes graphviz

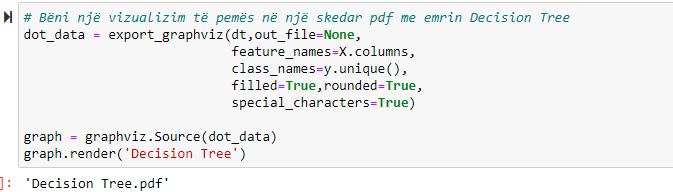
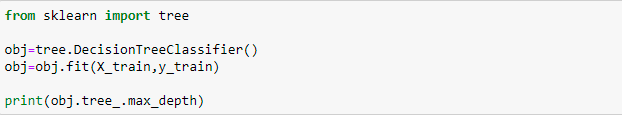


Fig 14.Paraqitja e Decision Tree ne pdf file.

Paraqitja e vizualizimit të pemës ne formë të një skedari pdf të emërtuar Decision Tree.

Gjithashtu **Python** ofron edhe funksione tjera për studim të pemës si llogaritja e thellësisë së pemës, llogaritja e nyjeve etj.

  
 Fig 15. Llogaritja e thellesise se pemes

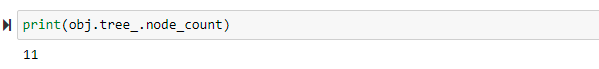


Fig 16. Llogaritja e nyjeve

1. **Implementimi i Decision Tree në një aplikacion**

Aplikacioni është zhvilluar në gjuhën programuese Python më saktësisht me kornizën

Flask.

Flask është ndër korrnizat më të njohura e cila e bënë zhvillimin në ueb tejet të lehtë.

Së pari është bërë serializimi i fajllit në Jupyter Notebook në mënyrë që të mund ta

përdorim atë në Flask dhe pastaj të bëjmë lidhjen me ndërfaqjen grafike.



Fig 17. Serializimi i modelit me anë të Pickle

Më pas në editorin Pycharm është filluar procesi i lidhjes së modelit të zhvilluar më ndërfaqen e aplikacionit me anë të Flask.

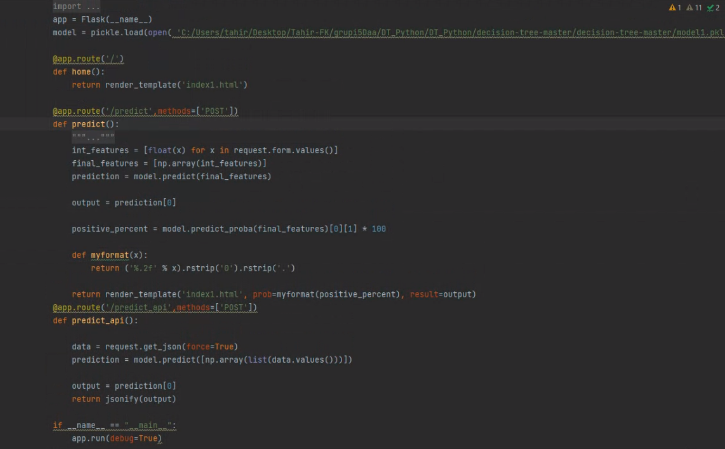


Fig 18.Kodi ne Flask

**5.1 Startimi i aplikacionit**

Aplikacioni është i hostuar lokalisht në kompjuter me anë të editorit PyCharm.

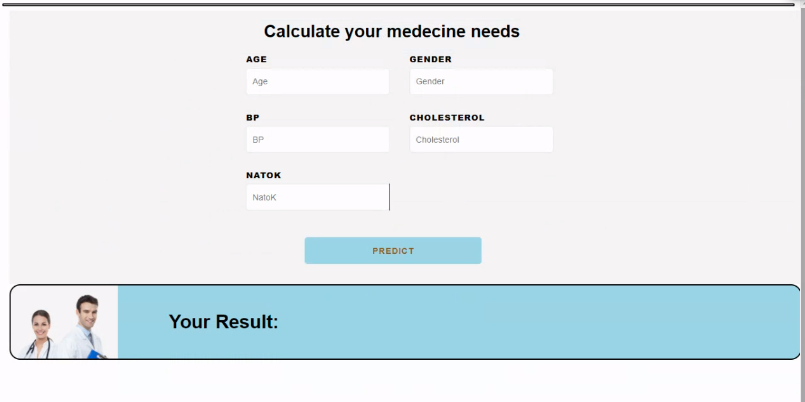
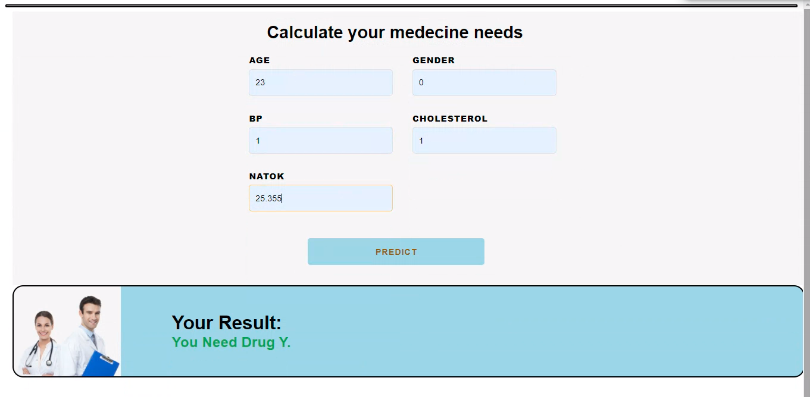


Fig 19. Nderfaqja e aplikacionit

Përdoruesi jep të dhënat kryesore që ndikojnë në predikimin e medikamenteve qe ju nevojiten,sic janë:

* Mosha
* Gjinia
* Presioni i gjakut
* Kolesteroli
* Natriumi në Kalium

Në vijim është shfaqur rezultati që jep aplikacioni pasi që i kemi japur të dhëna.



1. **Referencat**
2. <https://sefiks.com/2020/03/18/a-step-by-step-chaid-decision-tree-example/>
3. <https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning#:~:text=2017%20259.3k-,Introduction,of%20artificial%20intelligence%20(AI).&text=Because%20of%20this%2C%20machine%20learning,has%20benefitted%20from%20machine%20learning>.
4. <https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08>
5. <https://deveshpoojari.medium.com/machine-learning-basics-descision-tree-from-scratch-part-i-4251bfa1b45c#:~:text=A%20decision%20tree%20is%20a,based%20on%20the%20attribute%20value>.
6. <https://becominghuman.ai/decision-trees-in-machine-learning-f362b296594a?gi=7dbb4f4d0e2e>