Математички факултет

Универзитет у Београду

СЕМИНАРСКИ РАД

У оквиру курса Истраживање података 2

Тема:

Успешност ICD терапије

Професор: Студент:

Ненад Митић Марко Савић, 149/2019

Април, 2023.

САДРЖАЈ

1.Увод

1.1 Предмет истраживања

Напрасна срчана смрт (Sudden Cardiac Death - SCD) је један од најчешћих узорка смрти и дешава се у року краћем од сат времена од појаве првих симптома. Напрасна срчана смрт може бити условљена другим срчаним болестима, али се може догодити и код пацијената без других срчаних обољења. Превенција SCD се врши уградњом ICD (IntraCardiac Defibrilator) уређаја који у одговарајућим тренуцима врши слање електричних импулса како би реактивирао срчани рад. Иако ICD даје добре резултате у превенцији SCD, индикације за уградњу ICD уређаја код пацијената нису у потпуности прецизне. Код пацијената који су већ преживели једну епизоду, индикација за уградњу ICD је јасна, док је предвиђање потребе за уградњу ICD уређаја ради спречавања прве епизоде SCD непрецизно.

Тренутни стандард за предвиђање SCD се заснива на EF (Ejection Fraction) вредности - проценту крви који напушта срце при свакој контракцији. Уколико је EF вредност мања од прага од 30%, доноси се одлука да је уградња ICD уређаја ипак потребна. Ипак, овај начин предвиђања је далеко од савршеног и ***предмет овог истраживања*** је да се пронађу друге клиничке слике пацијената које би потенцијално довеле до бољег предвиђања успешности ICD терапије која се огледа у активацији ICD уређаја у потребном тренутку и исхода преживљавања.

За предвиђање ће бити приказан рад пет различитих алгоритама класификације података.

1.2.Претпроцесирање

Подаци који су добијени представљају веома детаљне клиничке слике пацијената, стога је као први корак извршено претпроцесирање, којим смо од полазних података, избацивањем нерелевантних, ретких атрибута и другим методама, добили податке погодне за коришћење у алгоритмима.

Како су алгоритми које смо користили првобитно намењени нумеричким вредностима, извршили смо претварање категоричких вредности у нумеричке.

Као на пример извршили смо пресликавање често појављиваних вредности : “da” у 1, док смо “ne” пресликали у 0.

Аналогно смо урадили и за колону “pol” : “muski” смо пресликали у 1, док смо “zenski” пресликали у 0.

У колони “NYHA” вредности римских бројева су замењене арапским бројевима.

У колони “AF_pre_ugradnje” смо обе вредности “paroksizmalna” и “permanentna” заменили вредношћу 1, јер се вредности у нашем случају могу посматрати сличним.

Потом смо у процесу претпроцесирања и припреме података уклонили колоне 'Trajanje_bolesti' и 'Uzrok_smrti' због нерелевантности у оквиру истраживања, док смо колоне 'vreme_do_prve_ICDth' и 'vreme_do_prve_VT' уклонили због тога што су подаци унутар њих ретки, имају више недостајућих података, него података са информацијама.

Због релативно мале количине података, и неуравнотежености класа података добијених за анализу и истраживање у фази претпроцесирања извршено је и вештачко генерисање података на основу већ постојећих (‘oversampling’) . Oversampling смо извршили над тренинг подацима применом SMOTE алгоритма из python-ove *imblearn.over_sampling* библиотеке. Више о начину

рада овог алгоритма на https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html .

2. КЛАЦИФИКАЦИОНИ АЛГОРИТМИ

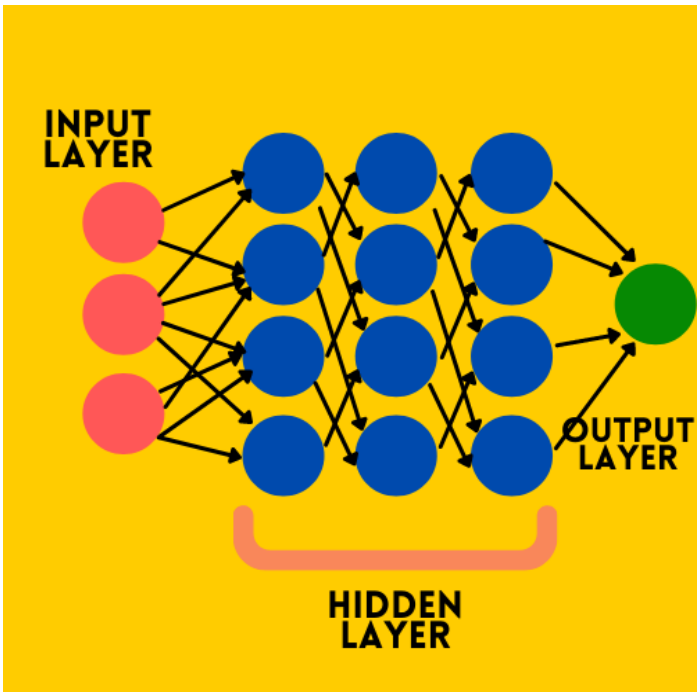
2.1. Artificial Neural Network(ANN)

Неуронске мреже су врста алгоритама машинског учења који су дизајнирани да науче образце и односе у подацима. Моделиране су према структури и функцији људског мозга и састоје се од повезаних слојева чворова или неурона.

Неуронске мреже функционишу тако што прихватају улазне податке и пролазе кроз серију слојева, при чему сваки слој врши низ математичких операција над подацима. Излаз једног слоја служи као улаз у следећи слој, а коначни излаз мреже генерише се последњим слојем.

Током обуке, неуронска мрежа прилагођава тежине и пристраности својих неурона како би минимизирала разлику између својих предвиђених излаза и стварних излаза за дати улаз. То се ради кроз процес називан повратна пропагација грешке, при чему мрежа прилагођава своје тежине и пристраности пропорционално грешци између њеног предвиђеног излаза и стварног излаза.

Када се неуронска мрежа обучи, може се користити за прављење предвиђања или класификацију нових података које није видела раније. То чини неуронске мреже моћним алатом за широк спектар примена, од препознавања слика до обраде природног језика до предиктивне аналитике.



За рад са неуронским мрежама у овом раду коришћена је python библиотека *Tensorflow*, и имплементирана је потпуно повезана мрежа са три слоја.

Као основни параметри потпуно повезаног слоја мреже задају се:

- Units - број неурона на посматраном слоју мреже
- Activation – активациона функција тог слоја

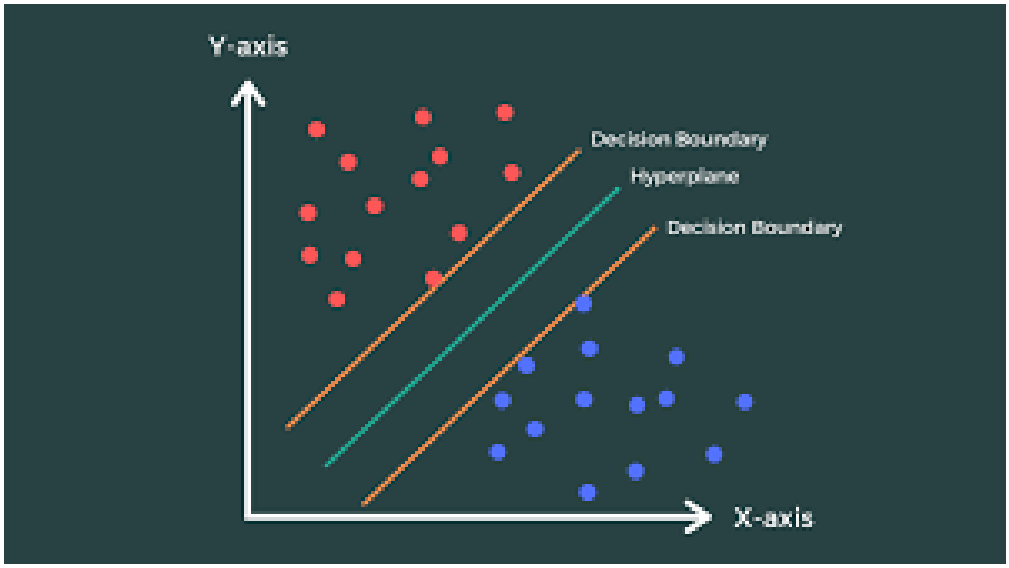
За прва два слоја мреже одабрана је активациона функција 'relu', док се број неурона на њима разликује.

Последњи слој се састоји од једног неурона у коме ће се налазити информација о класи којој инстанца припада, то је омогућено јер вршимо бинарну класификацију и користимо сигмоидну активациону функцију на последњем слоју.

Више о начину имплементације неуронских мрежа коришћењем Tensorflow библиотеке можете погледати на следећем линку: https://www.geeksforgeeks.org/artificial-neural-network-in-tensorflow/ .

2.2. Support Vector Machine

Support Vector Machine(SVM) је популаран алгоритам машинског учења за класификацију и регресију. SVM тражи хиперраван која најбоље раздваја податке у две класе тако да је маргина између та два скупа тачака максимална. SVM се може користити за рад са подацима високе димензионалности и прилагођава се различитим проблемима класификације и регресије.



За имплементацију SVM алгоритма у овом раду коришћена је python библиотека *Scikit learn*, I њена имплементација SVC (Support Vector Classifier) која представља специјализовану верзију SVM за класификацију, коју и вршимо у овом раду. Неки од параметара које задајемо јесу:

- C – регуларизациони параметар
- Kernel – омогућава специјализацију врсте језгра SVM машине коју користимо

Додатно о имплементацији SVC алгоритма у Scikit learn библиотеци, и о њеним параметрима, можете погледати на следећем линку:

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

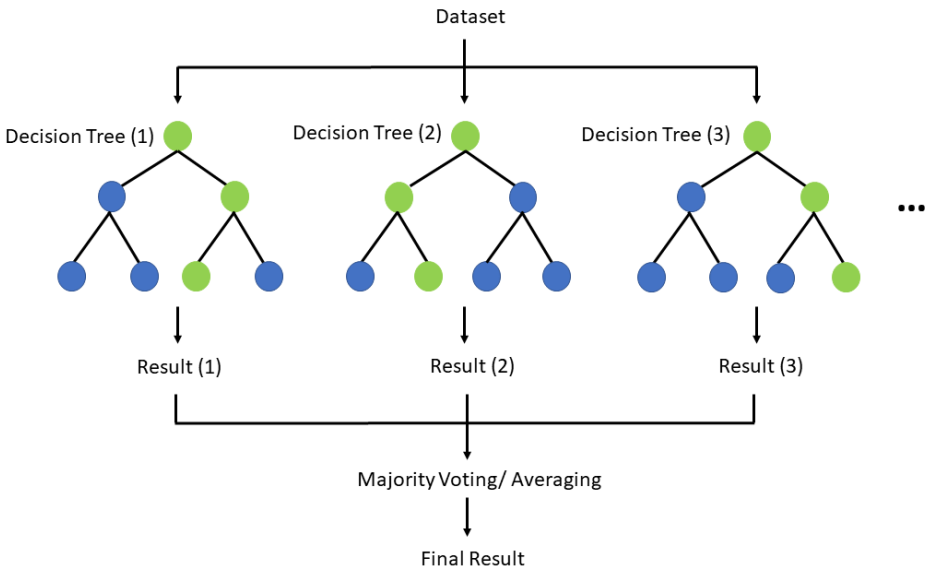
2.3. Random forest

Random Forest је популаран алгоритам машинског учења, који се темељи на ансамбл методи. Ансамбл метода комбинује више модела како би се постигла боља предикција.

Random Forest се састоји од више стабала одлучивања, при чему свако стабло користи насумични узорак података за тренирање. Стабла се затим комбинују како би се донела коначна одлука.

Када се врши предвиђање, Random Forest издваја најважније атрибуте података и користи их за доношење одлуке. Ова техника се може користити за класификацију или регресију и прилагођава се различитим проблемима предвиђања.

Random Forest је популарна техника због своје способности да се носи са високим ступњем шума и недостајућим вредностима података, као и због способности да ради са великим скуповима података.



Random Forest

За имплементацију Random Forest алгоритма у овом раду коришћено је заглавље *ensemble* у оквиру python библиотеке *Scikit learn*, у коме постоји имплементирана функција RandomForestClassifier. Свако дрво унутар ансамбла имплементирано је коришћењем DecissionTreeClassifier

класификатора. Неки од најчешће коришћених параметара RandomForestClassifier алгоритма су:

- n_estimators – број засебних дрвета у ансамблу
- max_depth – задаје максималну дубину дрвета
- criterion – функција која мери квалитет поделе у оквиру дрвета

За детаљнији опис имплементације алгоритма у оквиру библиотеке, као и детаљнији опис хиперпараметара класификатора погледати документацију:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

2.4. Decision Tree

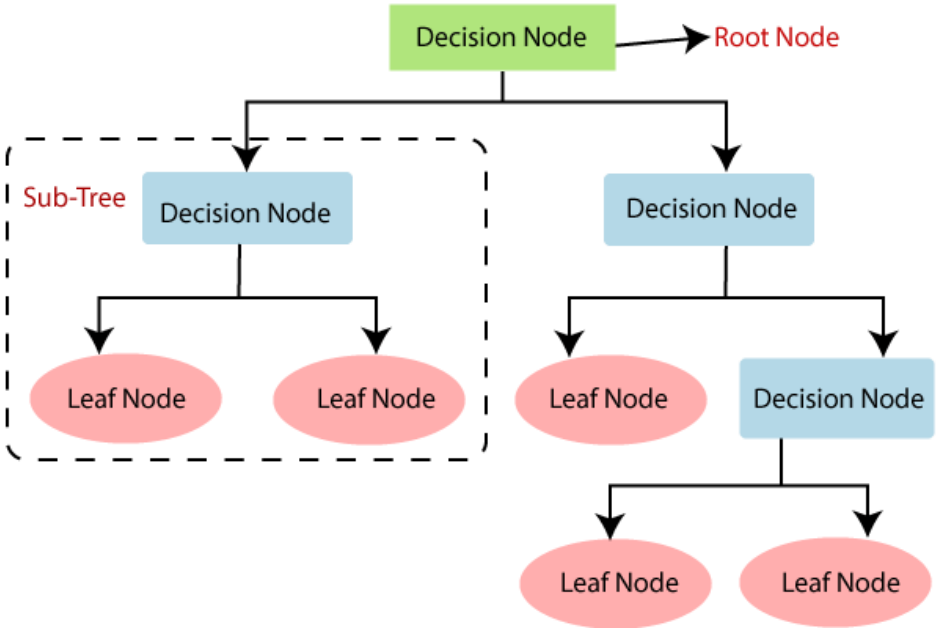
Стабло одлучивања је метода машинског учења која се користи за класификацију и предвиђање. Алгоритам гради стабло са гранама које се деле према различитим вредностима атрибута у подацима. Када се врши предвиђање, улазни подаци се прате кроз стабло и класификацијска одлука се доноси према гранама које воде до краја стабла. Ова метода је веома популарна због своје интерпретабилности и могућности прилагођавања различитим проблемима класификације и предвиђања.

За потребе овог рада за имплементацију Decision Tree алгоритма коришћено је заглавље *tree* у оквиру python библиотеке *Scikit learn* I класификатор DecisionTreeClassifier. Scikit learn библиотека у својој имплементацији стабла одлучивања користи CART алгоритам. Неки од хиперпараметара DecisionTreeClassifier-a су:

- criterion – функција квалитета поделе унутар стабла
- max_depth – максимална дубина стабла
- min_samples_split – минималан број инстанци унутар чвора да би могао бити подељен

За детаљнији опис имплементације класификатора и описа његових хиперпараметара погледати :

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

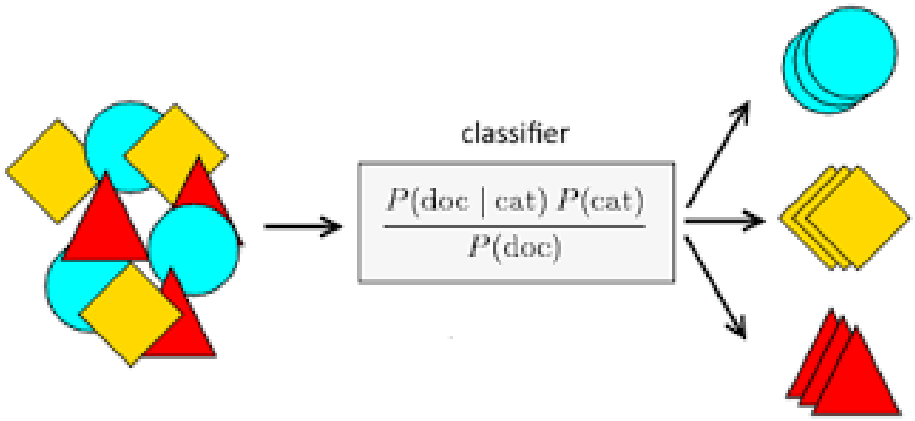


2.5. Naive Bayes

Naive Bayes је још једна често коришћена метода класификације и предвиђања. Овај алгоритам користи теорему Бајеса и претпоставку о независности атрибута у подацима, како би се израчунала вероватноћа припадности података различитим класама.

Када се користи наивни Бајес, улазни подаци се разврставају у категорије на темељу вредности атрибута које поседују. Затим се израчунава вероватноћа припадности података свакој од тих категорија. Класификација се затим врши према категорији са највећом вероватноћом.

Наивни Бајес је популаран због своје једноставности и брзине извршавања. Такође добро ради са великим скуповима података и омогућава изградњу модела из малог броја примера за тренирање. Међутим претпоставка о независности атрибута може бити ограничавајућа у неким случајевима и довести до погрешних предвиђања.



У корист тестирања понашања овог алгоритма на нашем конкретном проблему коришћена је класа GaussianNB из заглавља *naive bayes* библиотеке *Scikit learn*.

Више о имплементацији класе, као и њеним параметрима погледати на следећем линку: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

3.Анализа резултата

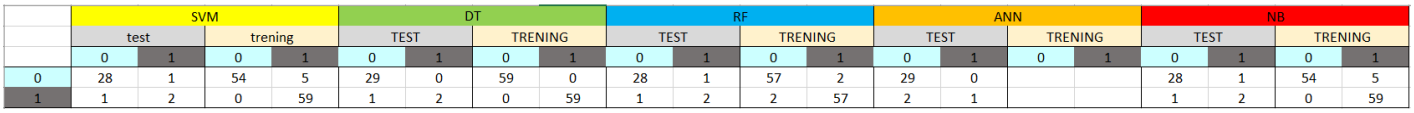
Како бисмо дошли до жељеног циља вршена је анализа резултата наведених алгоритама који су примењивани над улазним подацима на три различита начина.

Како је предмет истраживања успешност ICD терапије, која се огледа у постотку преживљавања тако је прво разматрано решење подразумевало да се алгоритми примењују над подацима из два дела. Први део представља примену алгоритама класификације над улазним подацима са циљем класификације атрибута 'ICD terapija', како би се у другом делу вршила класификација атрибута 'prezivljavanje' .

Први део

Циљ класификације атрибута 'ICD terapija' био је да уочимо како су остали атрибути који представљају клиничку слику повезани са тиме да ли је ICD апарат укључен или не. У том светлу алгоритми класификације су примењивани над подацима са циљем креирања модела за предвиђање укључености ICD апарата на основу клиничке слике пацијента.

Train/test split	SVM	Decision Tree	Random Forest	Neural Network(ANN)	Naive Bayes
70/30	0.9375	0.90625	0.9375	0.875	0.9375
70/30 stratify	0.9375	0.875	0.90625	0.9062	0.875
60/40 stratify	0.8372	0.8604	0.907	0.907	0.9535
50/50 stratify	0.7924	0.8868	0.849	0.793	0.9434



	SVM				DT				RF				ANN				NB			
	test		trening		TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	28	1	54	5	29	0	59	0	28	1	57	2	29	0			28	1	54	5
1	1	2	0	59	1	2	0	59	1	2	2	57	2	1			1	2	0	59

У првој табели приказане су тачности алгоритама примењених над тест подацима у зависности од начина поделе улазних података на тренинг и тест.

Генерално велики проблем у тренирању модела јесте био недостатак података. Тест података има мало и лоше су уравнотежене класе унутар њих, те код неких алгоритама не можемо јасно закључити да ли модели врше класификацију како треба.

У другој табели приказане су матрице конфузије посматраних алгоритама за тренинг и тест податке, у случају када смо извршили поделу података у односу 70/30. На тренинг подацима алгоритми генерално показују лепе резултате. Видимо да алгоритми дају сличне резултате класификације, такође видимо и да у класи 1 имамо само 3 инстанце у тест подацима те да није једноставно закључити да ли алгоритми врше класификације како треба или не. Како смо приметили да нам класе у тренинг и тест подацима нису уравнотежене покушали смо то да решимо увођењем стратификоване поделе података.

	SVM				DT				RF				ANN				NB			
	test		trening		TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING	
30 bez stra	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	28	1	54	5	29	0	59	0	28	1	57	2	29	0			28	1	54	5
1	1	2	0	59	1	2	0	59	1	2	2	57	2	1			1	2	0	59
70/30 strat	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	26	1	58	3	26	1	61	0	27	0	60	1	27	0			24	3	56	5
1	1	4	0	61	3	2	0	61	3	2	1	60	3	2			1	4	0	61

Видимо да се број инстанци тест података које припадају класи 1 повећао, и можемо приметити промене у класификацији наших класификартора. Међутим можемо приметити да сада и Дрво одлучивања и Random Forest и Неуронска мрежа већински греше у класификацији инстанци које припадају класи 1. На основу досада приказаног можемо закључити да SVM класификатор даје најбоље резултате.

Како је 5 инстанци и даље мали број за утврђивање да ли класификатори раде како треба на проблематичној класи алгоритми су покренути над подација где је извршена подела у односу 60/40 и 50/50, како нам се уравнотеженост класа показала као веома битна оставићемо је укључену у наставку истраживања.

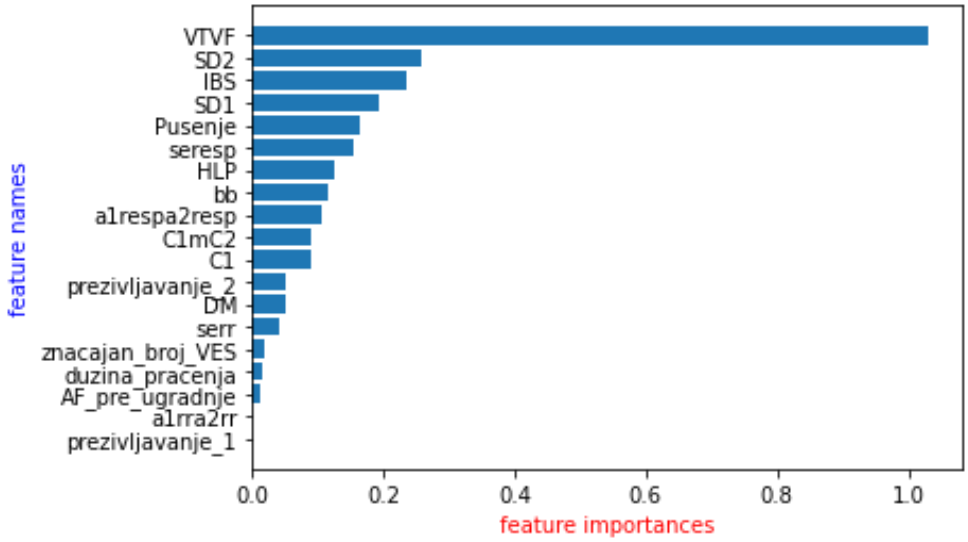
	SVM				DT				RF				ANN				NB			
	TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING	
30 bez stra	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	28	1	54	5	29	0	59	0	28	1	57	2	29	0			28	1	54	5
1	1	2	0	59	1	2	0	59	1	2	2	57	2	1			1	2	0	59
70/30 strat	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	26	1	58	3	26	1	61	0	27	0	60	1	27	0			24	3	56	5
1	1	4	0	61	3	2	0	61	3	2	1	60	3	2			1	4	0	61
60/40	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	32	4	49	3	35	1	52	0	35	1	52	0	36	0			35	1	47	5
1	3	4	0	52	5	2	0	52	3	4	1	51	4	3			1	6	0	52
50/50	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	38	6	44	0	43	1	44	0	42	2	44	0	41	3			42	2	40	4
1	5	4	0	44	5	4	0	44	6	3	0	44	8	1			1	8	0	44

На основу табеле са тачностима можемо приметити да са повећањем тест скупа података код већине алгоритама тачност опада, због смањења скупа података за тренирање. Једини алгоритам чија тачност расте са порастом скупа података за тренирање јесте Наивни Бајес, можемо приметити како на основу тачности тако и на основу матрица конфузије да он даје најбоље резултате класификације за повећани тест скуп (како 60/40 тако и 50/50).

SVM

Видимо да за тренинг податке SVM врши веома добру класификацију, док за тест податке због мањка података у другој класи не знамо да ли модел зна добро да класификује податке. Видимо да са повећањем тест скупа података класификатор приликом класификације малобројне класе 1 све више личи на класификатор који случајним избором одређује класу којој инстанца припада.

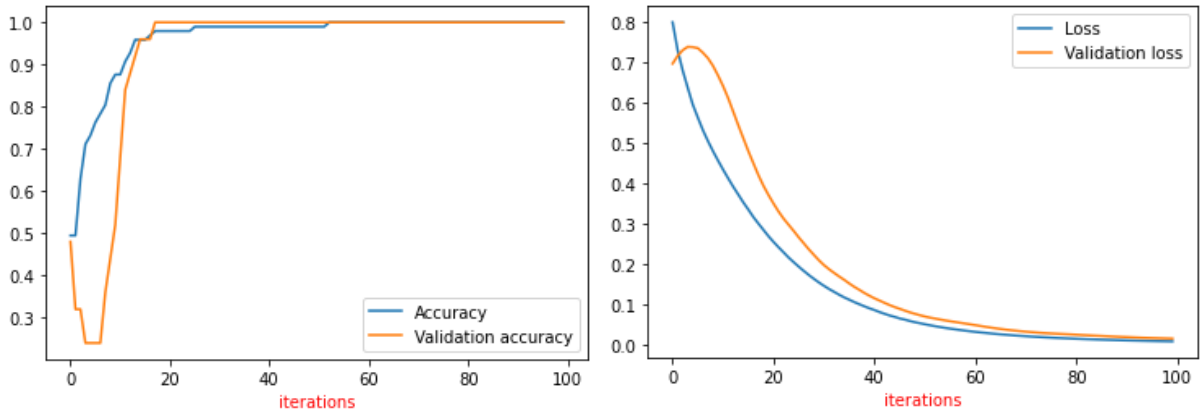
Посматрајући матрице конфузије и тачности предвиђања можемо закључити да SVM алгоритам даје најбоље резултате за поделу 70/30.



На датој слици можемо видети резултат рада SVM алгоритма, и његово предвиђање утицаја атрибута клиничке слике на класификацију укључивања ICD апарата. Можемо приметити да је најутицајнији атрибут VTVF, али такође можемо видети и утицаје осталих атрибута.

ANN

Посматрајући табелу са матрицама конфузије можемо закључити да овако формирана неуронска мрежа не уме да изврши класификацију проблематичне, малобројне класе 1. На наредним сликама можемо видети промену тачности и грешке кроз итерације тренирања мреже за поделу 70/30.

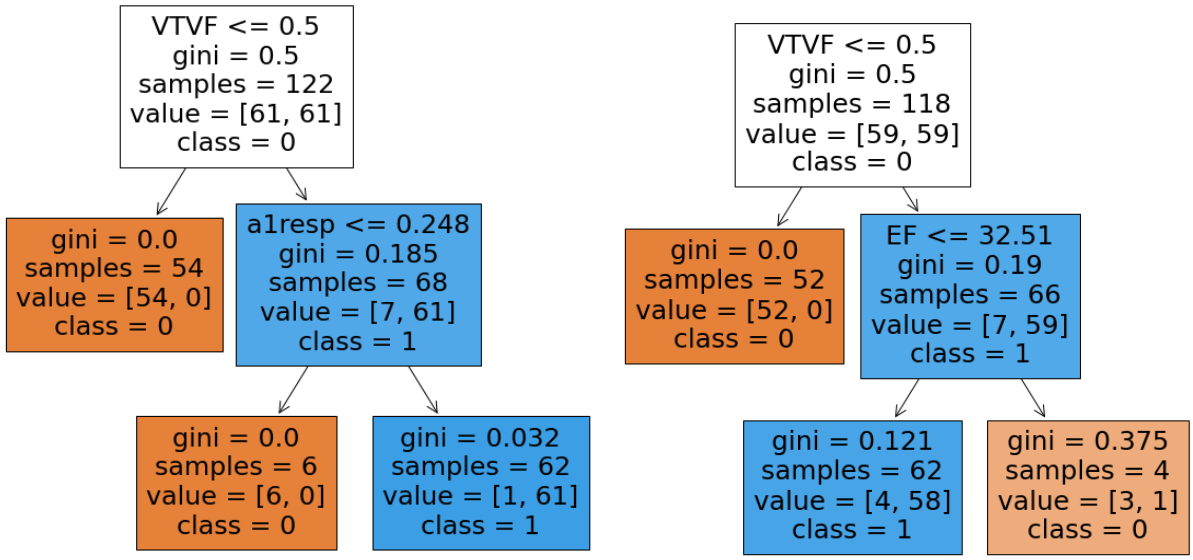


На основу графика видимо да је овако задатом моделу мреже потребно 20-ак епоха да би се добро истренирала мрежа.

ДРВО ОДЛУЧИВАЊА

Тачност класификације дрвета одлучивања такође опада са порастом величине тест скупа. Видимо да на тренингу скупу независно од поделе података дрво одлучивања врши безгрешну класификацију, док на тест скупу код малобројније класе има проблема, не колико неуронска мрежа, али и даље класификација инстанци класе 1 није прецизно одређена.

Дрво одлучивања нам као и SVM алгоритам омогућава приказ важности атрибута клиничке слике.



Слика са леве стране представља дрво одлучивања добијено за стратификовану поделу 70/30 , док слика десно представља дрво одлучивања за поделу 70/30 која није стратификована.

На основу резултата дрвета одлучивања можемо видети да је VTVF најутицајнији атрибут, али такође можемо доћи и до закључка :

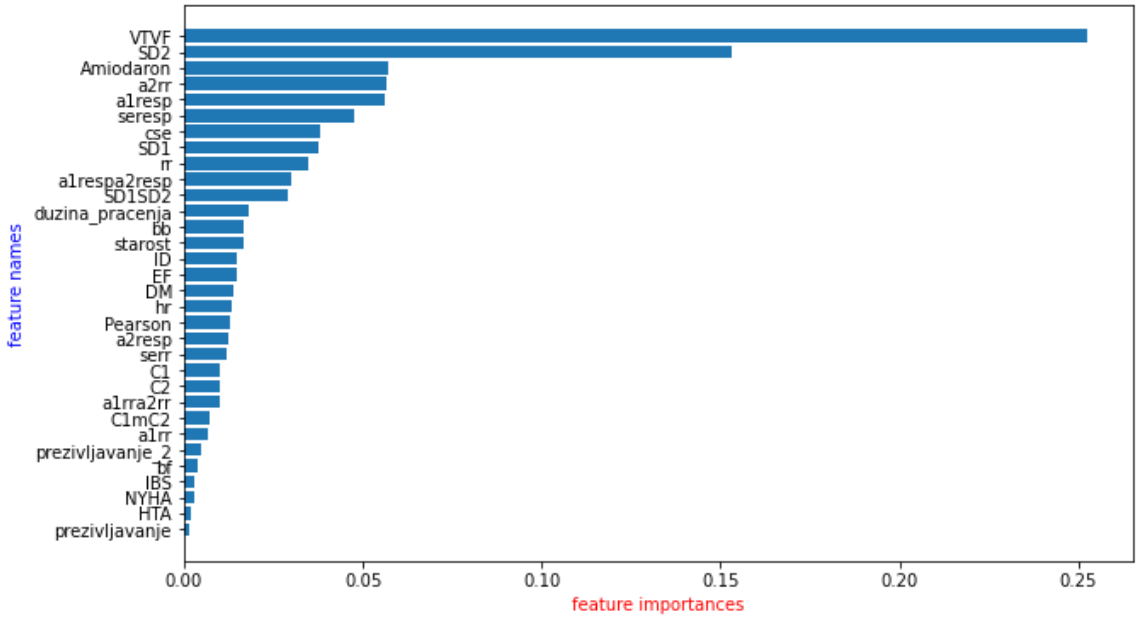
1. *Ако пацијент има VTVF вредност већу од 0.5 и вредност атрибута a1resp већу од 0.248 ICD терапију треба укључити, у супротном не.*
2. *Ако пацијент има VTVF вредност већу од 0.5 и вредност атрибута EF мању или једнаку од 32.51 ICD терапију треба укључити, у супротном не.*

Можемо приметити да наведени закључци имају смисла, јер се и у самом тексту задатка помиње закључак 2. :

- *Тренутни стандард за предвиђање SCD се заснива на EF (Ejection Fraction) вредности - проценту крви који напушта срце при свакој контракцији. Уколико је EF вредност мања од прага од 30%, доноси се одлука да је уградња ICD уређаја ипак потребна.*

<u>Random forest</u>

За Random Forest на основу табеле тачности такође можемо приметити да се са порастом величине тест скупа тачност смањује. Као и код дрвета одлучивања, веома добра класификација тренинг података, независно од поделе, али оно што нам прави проблем јесте као и код осталих алгоритама малобројна класа 1 у тест подацима. Код ансамбл методе Random forest такође можемо приметити важности атрибута клиничке слике:



Naive Bayes

Као што смо већ прокоментарисали како на основу табеле са тачностима тако и на основу матрица конфузије видимо да Наивни Бајес даје веома добре резултате, и једини је чија тачност расте са порастом тест скупа података. Посматрајући матрице конфузије можемо видети да на класификацији тренинг података показује мало лошије резултате него дрво одлучивања или Random forest, али показује значајно боље резултате на тест подацима, поготово на критичној малобројној класи 1.

Други део

У другом делу истраживања задатак је био класификација преживљавања. примењивани су исти класификациони алгоритми само над подацима из којих је избачен атрибут "prezivljavanje" као циљни атрибут.

Поново су алгоритми покренути за различите поделе података на тренинг и тест скуп. Резултате алгоритама можете погледати у наредној табели.

Сви наведени резултати подразумевају да је у првом делу истраживања подела стратификована.

Train/test	SVM	DT	RF	ANN	NB
70/30	0.9375	0.84375	0.78125	0.71875	0.59375
70/30 stratify	0.875	0.96875	0.8125	0.625	0.594
60/40	0.907	0.884	0.833	0.698	0.746
50/50	0.868	0.811	0.793	0.698	0.698

	SVM				DT				RF				ANN				NB			
	TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING	
30 bez stra	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	10	1	23	0	10	1	21	2	7	4	18	5	7	4			1	10	6	17
1	1	20	0	51	4	17	0	59	3	18	0	51	5	16			3	18	0	51
70/30 strat	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	8	2	24	0	9	1	19	5	4	6	18	6	4	6			4	6	19	5
1	2	20	1	49	0	22	1	49	0	22	0	50	6	16			7	15	7	43
60/40	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	11	1	22	0	11	1	18	4	9	3	17	5	8	4			3	9	6	16
1	3	28	0	41	4	27	0	41	4	27	0	41	4	27			3	28	0	41
50/50	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	14	2	18	0	16	0	18	0	9	7	15	3	4	12			2	14	5	13
1	5	32	0	35	10	27	2	33	4	33	0	35	7	30			2	35	0	35

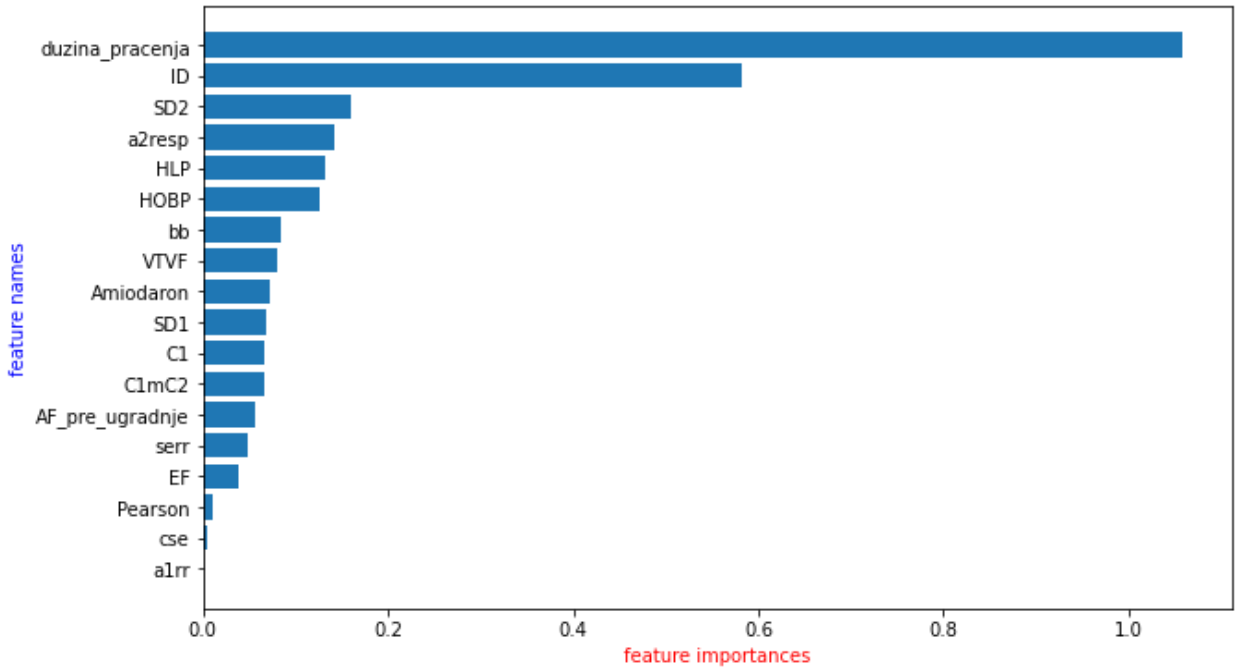
Како нам стратификовање поделе у другом делу не доноси значајне промене занемаримо га, јер класе нису неуравнотежене у мери као у првом делу истраживања.

SVM

На основу табела тачности и матрица конфузије можемо приметити да SVM алгоритам на овом делу истраживања показује веома добре резултате.

На тренинг скупу SVM класификатор врши безгрешну класификацију, док на тест скупу такође показује веома добре резултате, независно од поделе података на тренинг и тест.

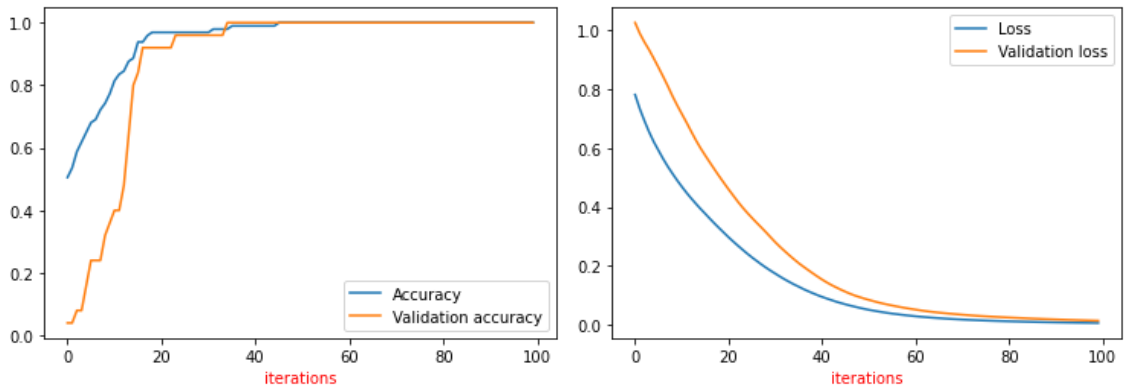
SVM класификатор нам омогућава визуелни приказ утицаја атрибута клиничке слике на класификацију атрибута “prezivljavanje”.



Примећујемо да SVM класификатор посматра атрибут дужина праћења као најзначајнији.

<u>Неуронска мрежа</u>

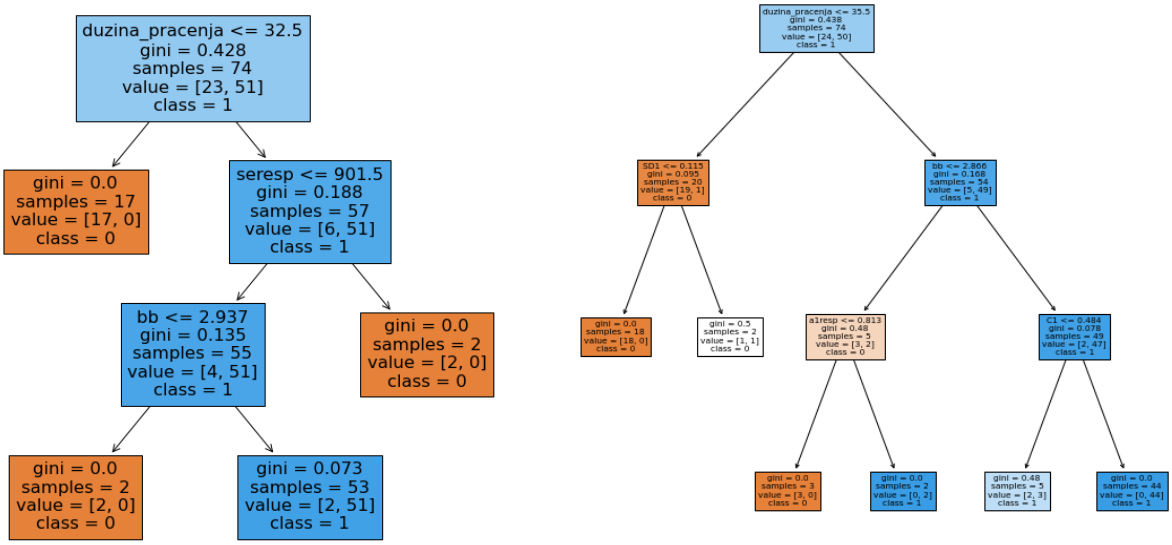
Као што можемо приметити из датих табела неуронска мрежа не показује похвалне резултате. Тачност се са смањењем величине тренинг скупа смањује, што можемо приметити и у матрици конфузије где се промашаји малобројније класе, у овом случају класе 0, значајно повећавају.



На слици са леве стране можемо видети промену тачности на тренинг и валидационом скупу, док на десној слици можемо видети како се грешка мењала кроз итерације.

Дрво одлучивања

Дрво одлучивања нам даје добре резултате на овом нивоу истраживања. Посматрајући матрице конфузије видимо да у већини случајева дрво врши класификацију на исправан начин. Са порастом тест скупа података повећава се и грешка, али тачност остаје на прихватљивом нивоу. Као и у првом делу истраживања дрво нам пружа визуелизацију начина на који он врши класификацију.



Са леве стране видимо дрво одлучивања добијено за поделу 70/30 где није узета у обзир стратификована подела, док са десне стране видимо дрво одлучивања за стратификовану 70/30 поделу која показује боље резултате.

Поново можемо извести закључке:

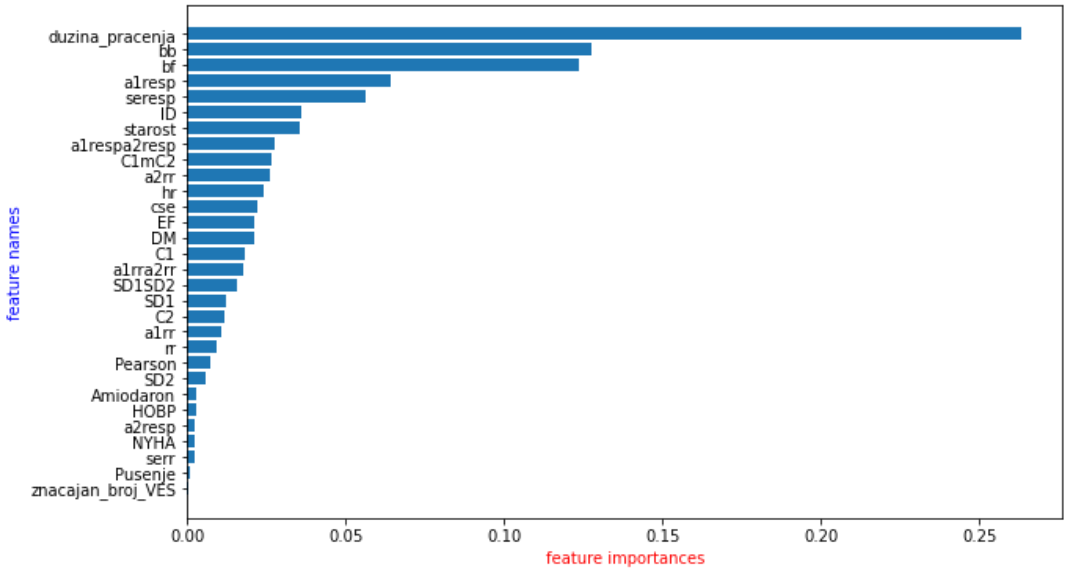
1. *Ако је дужина праћења већа од 32.5, вредност атрибута "seresp" мања или једнака од 901.5 и вредност атрибута "bb" већа од 2.937 пацијент ће преживети, у супротном неће.*

2.1. *Ако је дужина праћења већа од 35.5, вредност атрибута "bb" мања или једнака 2.866 и вредност атрибута "a1resp" већа од 0.813, пацијент ће преживети.*

2.2. Ако је дужина праћења већа од 35.5, вредност атрибута "bb" већа од 2.866 и вредност атрибута "C1" већа од 0.484, пацијент ће преживети.

Random Forest

Ансамбл метода не показује сјајне резултате у класификацији преживљавања. На основу матрица конфузије можемо приметити да је класификација непрецизна. Random forest нам такође омогућава визуелни приказ важности атрибута клиничке слике у предвиђању преживљавања.



Видимо да је најбитнији атрибут дужина праћења, што се поклапа са дрветом одлучивања, као и SVM класификатором, такође наредни по битности јесте атрибут "bb" који се такође појављивао и у анализама дрвета одлучивања и SVM класификатора.

Naive Bayes

Класификатор Наивног Бајеса показује веома лоше резултате на овом нивоу предвиђања. Посматрајући матрице конфузије можемо приметити да класификатор скоро увек погрешно класификује инстанце класе 0. Иако је показао најбоље резултате у првом делу истраживања, можемо рећи да на овом делу класификатор Наивног Бајеса показује веома лоше резултате.

У циљу истраживања, и поређења резултата извршена је и директна класификација преживљавања, без претходног класификовања атрибута “ICD terapija”. Како смо у досадашњем истраживању водили рачуна о атрибуту ICD , као најбитнијем у предвиђању преживљавања, сада је он у потпуности избачен из скупа података како бисмо уочили утицај осталих атрибута на преживљавање. Такође анализом резултата класификације “ICD terapije” закључили смо да је најутицајнији атрибут VTVF тако да ћемо у наставку приказати резултате добијене директном класификацијом преживљавања над два скупа података, где ће из једног као и “ICD terapija” због битности бити избачен и “VTVF”, док ћемо у другом посматрати и њега.

Обе варијанте подразумевају поделу од 70/30 без стратификовања, тако да је у наредној табели, као репер, приказано и регуларно, већ разматрано класификовање преживљавања које је посматрало и ICD атрибут.

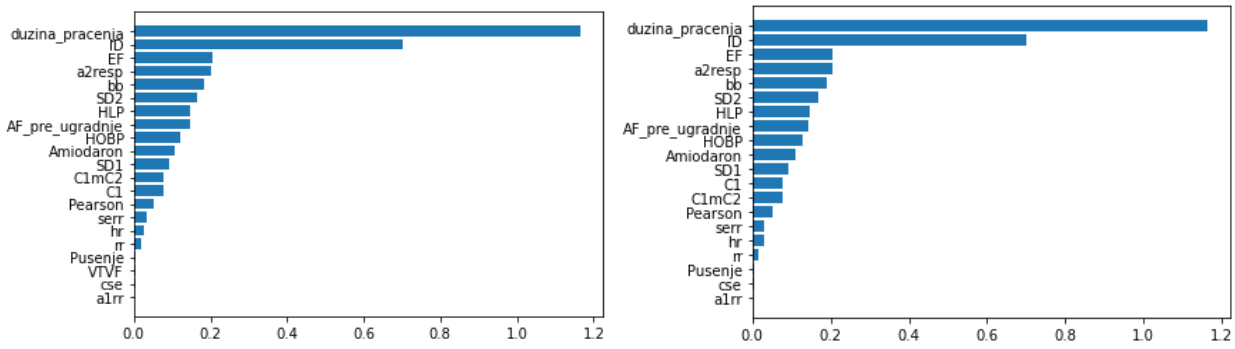
Train/test	SVM	DT	RF	ANN	NB
70/30	0.9375	0.84375	0.78125	0.71875	0.59375
Direktno bez VTVF	0.96875	1	0.8125	0.75	0.594
Direktno sa VTVF	0.96875	1	0.90625	0.78	0.5934

Одмах на почетку можемо приметити да се директном класификацијом добијају бољи резултати.

	SVM				DT				RF				ANN				NB			
	TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING		TEST		TRENING	
70/30	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	10	1	23	0	10	1	21	2	7	4	18	5	7	4			1	10	6	17
1	1	20	0	51	4	17	0	59	3	18	0	51	5	16			3	18	0	51
Direktno bez VTVF	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	11	0	51	0	11	0	51	0	10	1	48	3	8	3			1	10	11	40
1	1	20	1	50	0	21	1	50	5	16	1	50	4	17			3	18	0	51
Direktno sa VTVF	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	11	0	51	0	11	0	51	0	11	0	49	2	7	4			1	10	11	40
1	1	20	1	50	0	21	1	50	3	18	1	50	3	18			3	18	0	51

SVM

На основу табеле са задатим тачностима и матрица конфузије, можемо видети да SVM показује веома добре резултате у класификацији преживљавања у било ком случају. Добијамо исту тачност независно од укључености VTVF атрибута, тако да можемо рећи да он код SVM квалификатора не игра битну улогу у предвиђању преживљавања.



На левој слици можемо видети утицај атрибута на класификацију преживљавања у случају када је VTVF атрибут укључен, док је са десне стране приказ утицаја атрибута када је VTVF занемарен.

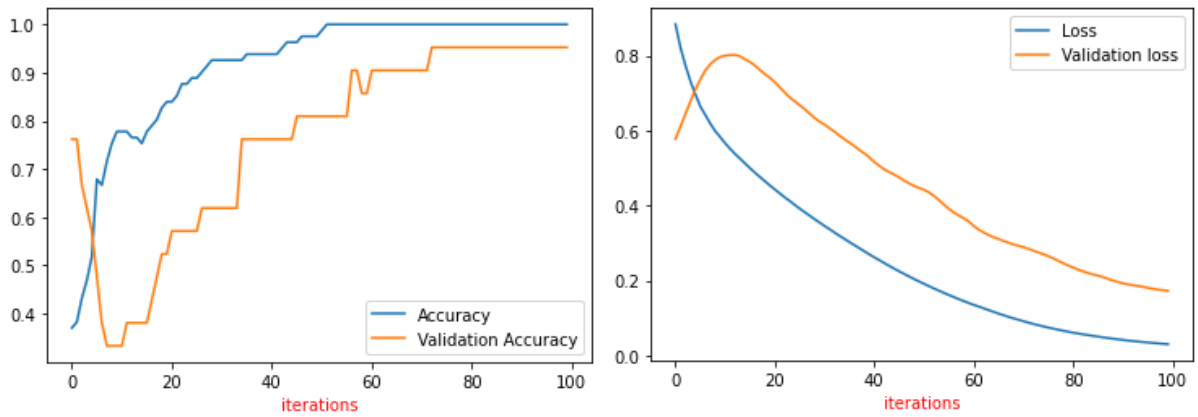
Поново као и на основу анализе тачности можемо закључити да атрибут VTVF не игра битну улогу у предвиђању преживљавања, те да се утицаји атрибута клиничке слике у великој мери поклапају, независно од његове укључености.

Видимо да је атрибут дужина праћења најистакнутији, као и у претходној анализи.

Поређењем са анализом утицаја атрибута SVM алгоритма који је посматрао и ICD терапију, можемо приметити да се доста атрибута поклапа, међутим и да су се некима утицају на предвиђање преживљавања променили. Дужина праћења је у сва три случаја најистакнутија, на другом месту је свуда ID, док је EF који је на трећем месту у оба случаја када не посматрамо ICD, у ситуацији када се ICD узима у обзир на доста нижем месту. Такође можемо приметити да је атрибут VTVF када је био посматран атрибут ICD био доста утицајнији него када смо занемарили ICD.

Неуронска мрежа

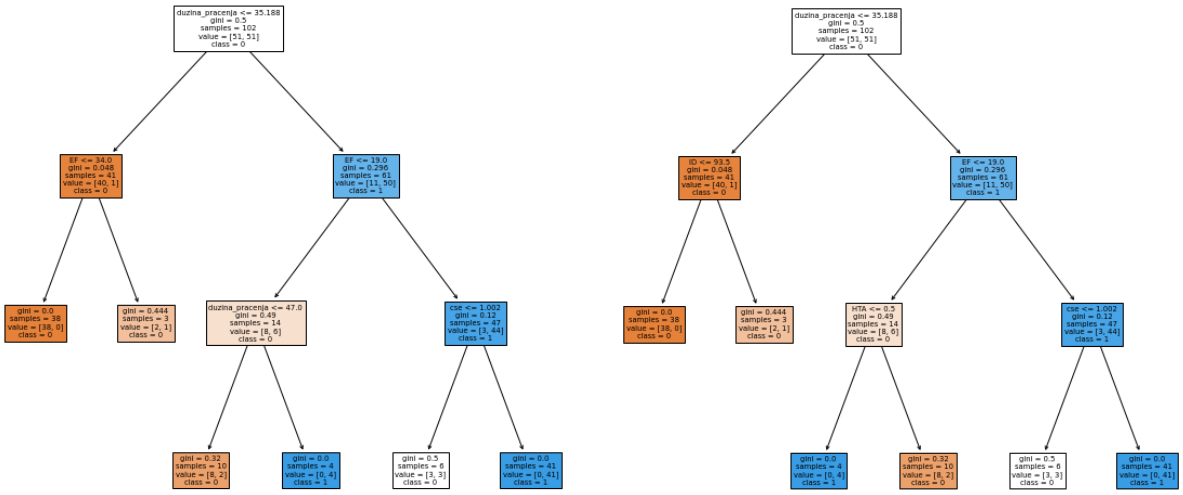
Посматрајући добијене резултате можемо закључити да неуронска мрежа исте архитектуре врши бољу класификацију у директној варијанти. Она поново не даје много боље резултате и и даље "брља" током класификације.



На сликама можемо видети промену тачности и грешке кроз епохе тренирања мреже, те видимо да заправо мрежа не успева у потпуности да се истренира и да јој је вероватно потребно још итерација(епоха).

Дрво одлучивања

Дрво одлучивања нам у обе директне варијанте даје безгрешно класификовање тест инстанци, док на тренинг подацима врши минималну грешку. Свакако показује боље резултате у директној варијанти. На наредним сликама ћемо видети како изгледају дрвета одлучивања која су вршила директну класификацију.



На слици лево је приказано дрво одлучивања класификације са VTVF атрибутом, док је десно без.

Као и до сада, изводимо закључке:

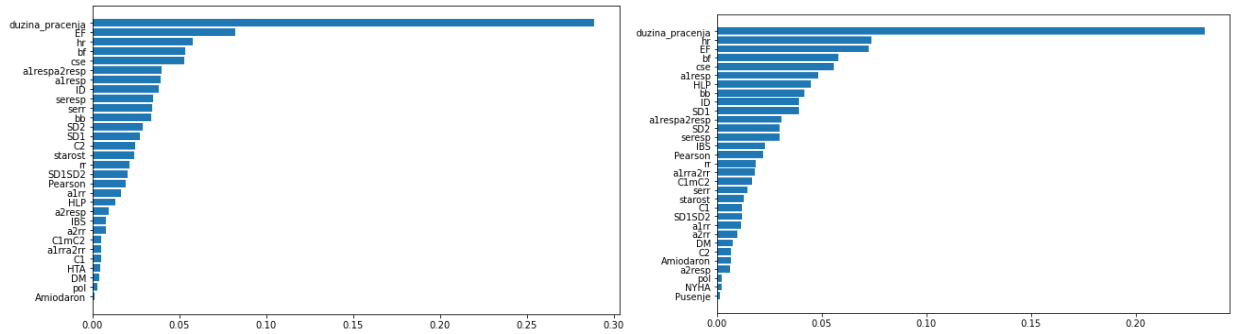
- Из класификатора са VTVF-ом:
  1. *Ако је дужина праћења већа од 47.0 и вредност атрибута EF мања или једнака 19 пацијент ће преживети.*
  2. *Ако је дужина праћења већа од 35.188,EF већи од 19 и вредност атрибута "cse" већа од 1.002 пацијент ће преживети.*
  3. *Ако је дужина праћења мања или једнака 35.188 и вредност EF већа од 34 тада пацијент неће преживети.*
- Из класификатора без VTVF:
  4. *Ако је дужина праћења мања или једнака 35.188 и вредност ID мања или једнака од 93.5 тада пацијент неће преживети.*
  5. *Ако је дужина праћења већа од 35.188, EF вредност мања од 19 и вредност атрибута HTA мања или једнака 0.5 пацијент ће преживети.*

Можемо приметити да се закључак број 3 понавља у оба случаја.

Random Forest

На основу табеле тачности и матрице конфузије видимо да Random forest у директној методи даје веома добре резултате. Тачности су боље него у

раздвојеном приступу, а и по матрицама конфузије видимо да се класификација врши са мањим грешкама. Следи визуелизација утицаја атрибута на преживљавање.



Са леве стране је приказ утицаја атрибута у класификатору који посматра VTVF,док је са десне стране приказан класификатор који не узима атрибут VTVF у обзир. Можемо приметити да постоје разлике у утицају атрибута на преживљавање у зависности од тога да ли је атрибут VTVF укључен или није.

Naive Bayes

Класификатор Наивног Бајеса показује веома лоше резултате у директној класификацији преживљавања, као и у претходној анализи предвиђања преживљавања. Класификатор скоро потпуно маши одређивање класе инстанци које припадају класи 0, стога је показао понашање које је мало боље од случајног(рандом) одабира класе којој ће инстанца припадати.

4.Потребни програми и ресурси

Комплетан код пројекта написан је у Python програмском језику, на оперативном систему Windows, користећи окружење Jupyter Notebook. Коришћене су python библиотеке Scikit Learn I Tensorflow. Рачунар на коме су тестирани алгоритми садржи 8GB RAM меморије. За покретање програма потребно је на рачунару имати инсталирано окружење Jupyter Notebook, као и сам python. Једноставним кликом на дугме Kernel и одабиром Restart & Run All опције унутар Jupyter Notebook-а извршиће се комплетно покретање програма испочетка

5. Закључак

Кроз приказану анализу резултата могли смо да видимо понашање пет посматраних алгоритама класификације на конкретном проблему.

Како је анализа вршена на различите начине, тако су се неки алгоритми у одређеним ситуацијама показали бољи, док су у другим заказали.

Алгоритам Наивног Бајеса се показао сјајно у првом делу истраживања, где се вршило предвиђање ICD терапије, док је у другом делу истраживања показао веома лоше резултате, како у варијанти где смо посматрали ICD тако и када смо га избацили из скупа података. Као што је наведено у објашњењу алгоритама, Бајесовски класификатори врше претпоставку о независности атрибута што може бити ризично и довести до грешака приликом класификације.

Алгоритам који је током читавог истраживања показивао квалитетне резултате је SVM алгоритам. Он је показао добре резултате како током класификације ICD терапије, тако и касније током класификације преживљавања, било директно, било засновано на ICD-у.

Алгоритам Неуронских мрежа се показао као најлошији, током целог истраживања овај алгоритам је имао проблема са класификацијом малобројних класа.

Алгоритми Дрвета одлучивања и ансамбл метода Random forest су се показали као солидни алгоритми за класификацију на овом примеру. У првом делу,током класификације ICD-а оба алгоритма су имала проблема са малобројном класом, пак њихови резултати су били знатно бољи од неуронске мреже. На класификацији преживљавања оба алгоритма показала су добре резултате, дрво одлучивања нешто боље, како у ситуацији где је разматран ICD, тако и када је занемарен.

Видели смо да су нам алгоритми SVM, Random Forest и Дрво одлучивања омогућили визуелни приказ утицаја атрибута на класификацију, и могли смо приметити да се утицајни атрибути нису у многоме разликовали у зависности од алгоритма. То нам је омогућило да одговоримо на захтев и изведемо

одређене клиничке слике на основу којих се може извршити предвиђање уградње ICD апарата и преживљавања.

6. Литература

1. https://www.crcpress.com/Data-Classification-Algorithms-and-Applications/Aggarwal/p/book/9781466586741
2. https://www.springer.com/gp/book/9783319102467
3. https://scikit-learn.org/stable/
4. https://www.tensorflow.org/