

MATEMATIČKI FAKULTET

MIHAJLO VIĆENTIJEVIĆ

DAVID NEDELJKOVIĆ

Social Power NBA

Profesor
Nenad Mitić

Asistent
Mirjana Maljković

Sadržaj

1	Analiza podataka	2
1.1	O podacima	2
1.2	Elementi van granica	3
2	Pravila pridruživanja	8
3	Klasterovanje	11
3.1	Cilj klasterovanja	11
3.2	Priprema podataka za klasterovanje	11
3.3	Tehnike klasterovanja	12
3.3.1	K-sredina	12
3.3.2	Hijerarhijsko klasterovanje	15
4	Klasifikacija	17
4.1	Cilj klasifikacije	17
4.2	Priprema podataka za klasifikaciju	17
4.2.1	Analiza atributa	17
4.2.2	Podela podataka	19
4.3	Tehnike klasifikacije	19
4.3.1	Metode zasnovane na drvetima odlučivanja	19
4.3.2	Statistički zasnovane metode	25
4.3.3	Metode zasnovane na instancama	26
4.3.4	Neuronske mreže	28
4.3.5	Metode zasnovane na podržavajućim vektorima	28
4.4	Rezime	33
4.4.1	KNIME	33
4.4.2	SPSS	34

Poglavlje 1

Analiza podataka

1.1 O podacima

Skup podataka „Social Power NBA” ¹ se zasniva na sledećim podacima:

- Osnovne informacije o igraču
- Karakteristike u igri
- Statistički podaci o igraču
- Twitter aktivnosti igrača

U nastavku je data tabela sa opisom atributa. Nisu opisani svi atributi, već neki atributi koje ćemo koristiti za dalje istraživanje podataka.

¹<https://www.kaggle.com/noahgift/social-power-nba>

PLAYER_NAME	Ime i prezime
AGE	Godine igrača
GP	Broj odigranih utakmica
MIN	Broj odigranih minuta
AST_PCT	Procenat uspešnih asistencija
AST_RATIO	Kontrola lopte
OREB_PCT	Procenat ofanzivnih skokova
DREB_PCT	Procenat defanzivnih skokova
EFG_PCT	Procenat davanja koševa iz igre
USG_PCT	Procenat poseda lopte u odnosu na posed svog tima
FGM	Ukupno postignutih koševa iz igre
FGA	Ukupno bacanja na koš
SALARY_MILLIONS	Zarada u milionima
PTS	Prosečan broj postignutih koševa
TWITTER_FOLLOWER_COUNT	Broj Twitter pratilaca u milionima

Table 1.1: Opis podataka nba_2016-2017_100.csv

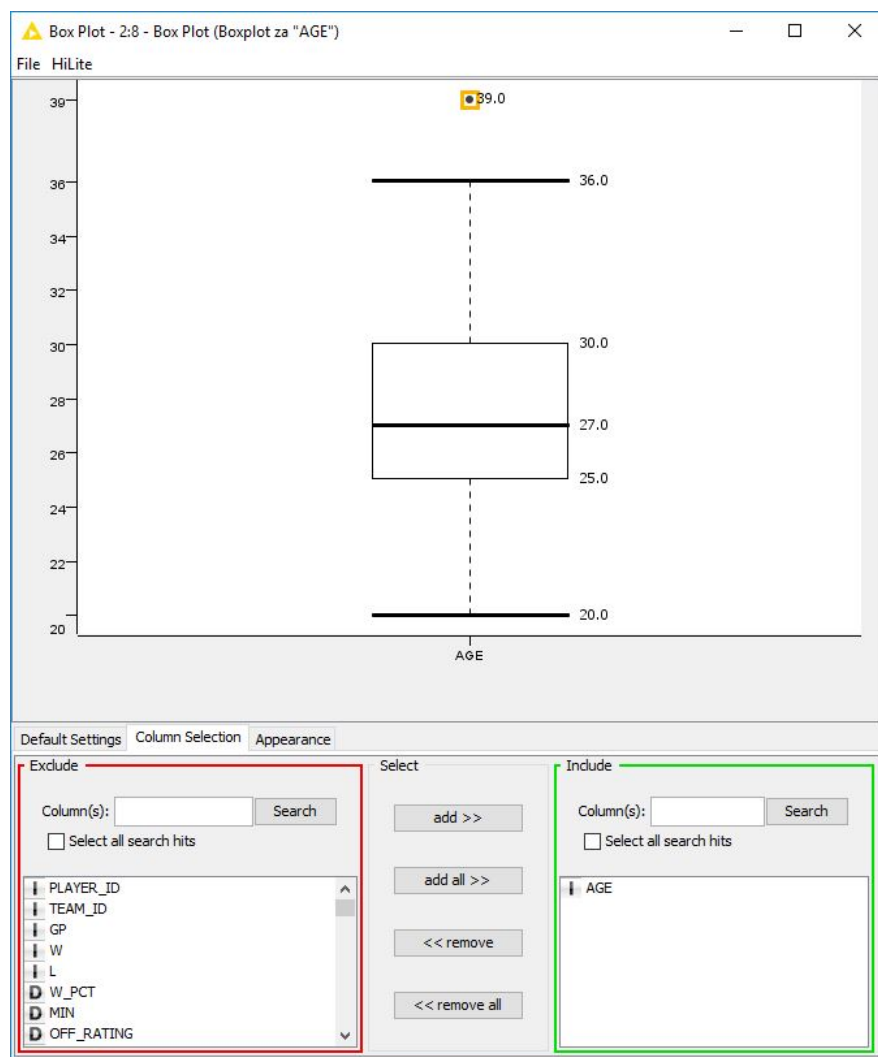
1.2 Elementi van granica

Elementi van granica su objekti koji se po svojim karakteristikama znatno razlikuju od ostalih objekata. Nas je konkretno zanimalo da li se neki igrač izdvaja po nekom atributu. Daljom analizom podataka primetili smo da za atribut *AGE* postoji igrač koji se znatno razlikuje od ostalih. Preciznije, primenom *Boxplot* čvora može se uočiti vrednost 39 kao vrednost koja znatno odstupa od ostalih (slika 1.1).

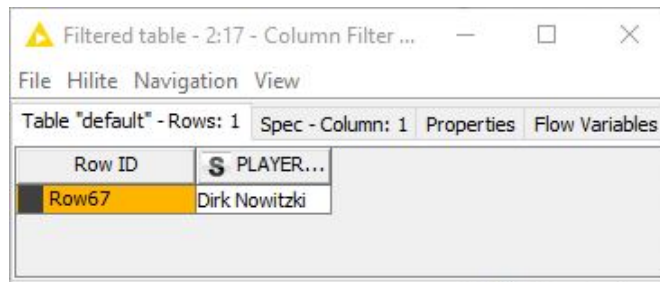
Daljom obradom podataka izvlačimo ime tog igrača tj. igrača koji je znatno stariji od drugih. To je „Dirk Nowitzki” (slika 1.2).

Analizom podataka smo došli i do drugih zanimljivih pojava. Tako smo atribut *SALARY_MILLIONS* kategorizovali u 3 klase tj. platnih razreda:

- Niska
- Srednja
- Visoka



Slika 1.1: Vrednost van granice

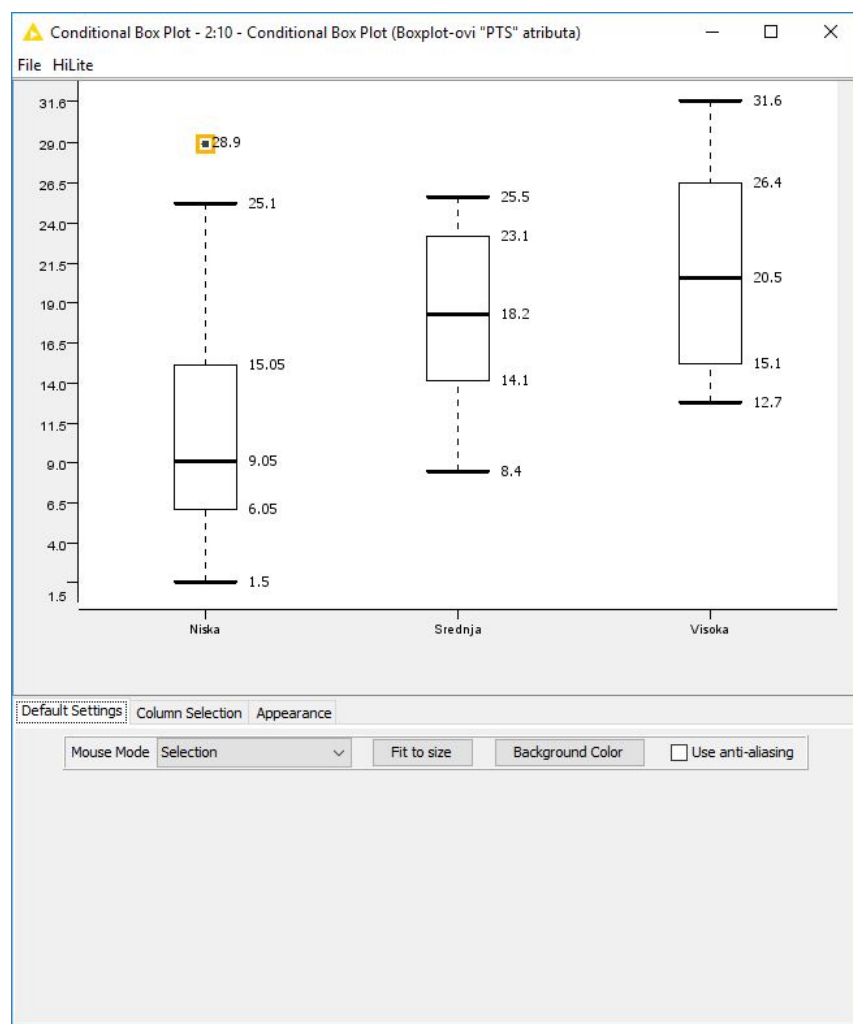


Row ID	PLAYER...
Row67	Dirk Nowitzki

Slika 1.2: Izvučen igrač

Koristeći *Conditional Boxplot* čvora za novodobijeni kategorički atribut uvideli smo da postoji igrač koji postiže mnogo koševa po utakmici a pripada niskom platnom razredu. Na slici (slika 1.3) mozemo da uočimo tu zanimljivost.

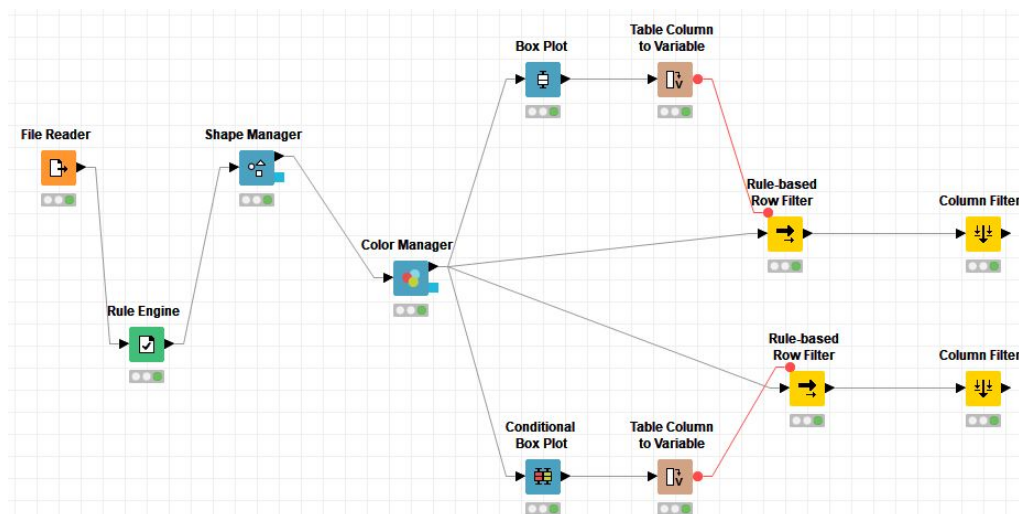
Kao i u prethodnom slučaju daljom transformacijom podataka dobijamo ime tog igrača. To je „Isaiah Thomas” (slika 1.4).



Slika 1.3: Element van granice u odnosu na kategoriju

Filtered table - 2:18 - Column Filter (...)	
File Hilite Navigation View	
Table "default" - Rows: 1 Spec - Column: 1 Properties Flow Variables	
Row ID	PLAYER...
Row17	Isaiah Thomas

Slika 1.4: Izvučen igrač



Slika 1.5: KNIME: Elementi van granica

Poglavlje 2

Pravila pridruživanja

Na osnovu podataka koje posmatramo postavlja se pitanje da li možemo nesto da zaključimo? Da li neke karakteristike igrača zavise od drugih njegovih karakteristika? Da bi to ispitali potrebno je da razumemo pravila pridruživanja. Pravila pridruživanja opisuju relacije između skupova stavki u podacima, i oblika su

$$A \Rightarrow B$$

gde su A i B skupovi stavki predstavljeni u skupu podataka. U našem slučaju skupove stavki predstavljaju skupovi karakteristika igrača. U *KN-IME* alatu (slika 2.1) smo koristili čvor *Association Rule Learner* koji kao ulazne parametre prima podatke koje smo prethodno normalizovali i kategorizovali u 4 sledeće kategorije:

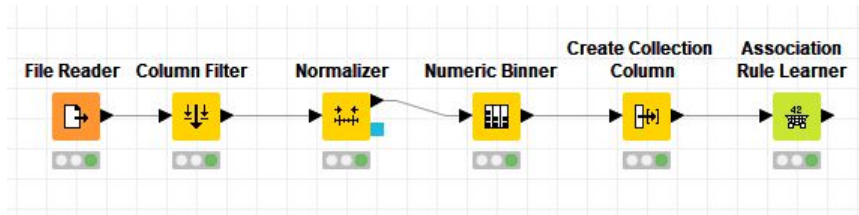
1. Loš
2. Dobar
3. Vrlo dobar
4. Odličan

Kako bi izračunali pravila pridruživanja potrebno je da definišemo *minimalnu podršku* (eng: *min support*) i *minimalnu pouzdanost* (eng: *min confidence*). Podrška nam govori koliko je pravilo korisno i izračunava se po sledećoj formuli:

$$support(A \Rightarrow B) = \frac{\#(A \cup B)}{N}$$

Pouzdanost nam govori koliko je pravilo precizno a njena formula je:

$$confidence(A \Rightarrow B) = \frac{\#(A \cup B)}{\#(A)}$$



Slika 2.1: KNIME: Čest skup stavki

gde $\#(A)$ predstavlja broj pojavljivanja skupa stavki u kompletnom skupu a N broj redova u kompletnom skupu.

U našem slučaju vrednost *minimalne podrške* smo inicijalizovali na 0.05 (5%) a *minimalne pouzdanosti* na 0.8 (80 %). Analizom *Lift* vrednosti koja predstavlja meru interesantnosti pravila izraženu kroz formulu:

$$Lift = \frac{confidence(A \Rightarrow B)}{support(B)}$$

došli smo do sledećih zanimljivih pravila (Slika 2.2):

- Pravilo 398: Igrači koji imaju visok procenat ofanzivnih skokova, mali procenat uspešnih asistencija i dobar procenat davanja koševa iz igre su nesebični tokom igre.
- Pravilo 32: Igrači koji manje dobacuju a više gube loptu, igraju dugo i imaju visok broj defanzivnih skokova su vrlo dobro plaćeni.

▲ Frequent itemsets/Association rules - 2:18 - Association Rule Learner

File Hilite Navigation View

Table "default" - Rows: 1177 Spec - Columns: 6 Properties Flow Variables

Row ID	D S...	D C...	D ▼ Lift	S Conseq...	S Implies	(...) Items
rule398	0.06	0.857	6.122	USG_PCT_1	<---	[OREB_PCT_3,AST_PCT_1,EFG_PCT_2]
rule207	0.05	0.833	5.952	USG_PCT_1	<---	[AST_PCT_1,EFG_PCT_2,MIN_2,SALARY_1]
rule399	0.06	1	5.556	EFG_PCT_2	<---	[OREB_PCT_3,USG_PCT_1,AST_PCT_1]
rule712	0.08	0.889	4.938	EFG_PCT_2	<---	[USG_PCT_1,AST_PCT_1]
rule314	0.05	1	4.762	OREB_PCT_3	<---	[PACE_3,AST_RATIO_1,AST_PCT_1,USG_PCT_2,DREB_PCT_3,SALARY_1]
rule517	0.06	1	4.762	SALARY_2	<---	[OREB_PCT_1,DREB_PCT_1,USG_PCT_2,MIN_4,EFG_PCT_1]
rule384	0.06	0.857	4.762	EFG_PCT_2	<---	[AST_RATIO_1,USG_PCT_1,AST_PCT_1]
rule389	0.06	0.857	4.762	EFG_PCT_2	<---	[USG_PCT_1,AST_PCT_1,SALARY_1]
rule392	0.06	0.857	4.762	MIN_2	<---	[USG_PCT_1,AST_PCT_1,SALARY_1]
rule208	0.05	0.833	4.63	EFG_PCT_2	<---	[USG_PCT_1,AST_PCT_1,MIN_2,SALARY_1]
rule209	0.05	0.833	4.63	MIN_2	<---	[USG_PCT_1,AST_PCT_1,EFG_PCT_2,SALARY_1]
rule457	0.06	0.857	4.082	OREB_PCT_3	<---	[PACE_3,USG_PCT_2,DREB_PCT_3,SALARY_1]
rule32	0.05	1	4	SALARY_3	<---	[AST_RATIO_1,MIN_4,DREB_PCT_4]
rule126	0.05	1	4	SALARY_3	<---	[OREB_PCT_2,AST_RATIO_1,MIN_4,EFG_PCT_1]
rule52	0.05	1	3.448	DREB_PCT_3	<---	[OREB_PCT_3,USG_PCT_2,MIN_3]
rule302	0.05	1	3.448	DREB_PCT_3	<---	[AST_RATIO_1,OREB_PCT_3,AST_PCT_1,USG_PCT_2,EFG_PCT_1,SALARY_1]
rule308	0.05	1	3.448	DREB_PCT_3	<---	[PACE_3,AST_RATIO_1,OREB_PCT_3,AST_PCT_1,USG_PCT_2,SALARY_1]
rule454	0.06	1	3.448	DREB_PCT_3	<---	[PACE_3,OREB_PCT_3,USG_PCT_2,SALARY_1]
rule531	0.06	1	3.448	DREB_PCT_3	<---	[AST_RATIO_1,OREB_PCT_3,AST_PCT_1,EFG_PCT_1,SALARY_1]
rule535	0.06	1	3.448	DREB_PCT_3	<---	[OREB_PCT_3,AST_PCT_1,USG_PCT_2,EFG_PCT_1,SALARY_1]
rule540	0.06	1	3.448	DREB_PCT_3	<---	[AST_RATIO_1,OREB_PCT_3,AST_PCT_1,USG_PCT_2,SALARY_1]
rule662	0.07	1	3.448	DREB_PCT_3	<---	[OREB_PCT_3,AST_PCT_1,EFG_PCT_1,SALARY_1]
rule667	0.07	1	3.448	DREB_PCT_3	<---	[OREB_PCT_3,AST_PCT_1,USG_PCT_2,SALARY_1]
rule743	0.08	1	3.448	DREB_PCT_3	<---	[OREB_PCT_3,USG_PCT_2,SALARY_1]
rule341	0.06	0.857	3.429	SALARY_3	<---	[OREB_PCT_2,MIN_4,EFG_PCT_1]
rule219	0.05	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,PACE_3,MIN_4,EFG_PCT_1,SALARY_3]
rule222	0.05	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,USG_PCT_3,MIN_4,EFG_PCT_1,SALARY_3]
rule254	0.05	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,PACE_3,OREB_PCT_1,USG_PCT_2,EFG_PCT_1]
rule433	0.06	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,PACE_3,USG_PCT_2,EFG_PCT_1]
rule497	0.06	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,OREB_PCT_1,MIN_4,EFG_PCT_1,SALARY_3]
rule591	0.07	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,PACE_3,USG_PCT_2]
rule629	0.07	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,MIN_4,EFG_PCT_1,SALARY_3]
rule728	0.08	1	3.333	AST_PCT_2	<---	[AST_RATIO_2,EFG_PCT_1,SALARY_3]

Slika 2.2: Pravila pridruživanja

Poglavlje 3

Klasterovanje

Klasterovanje ima za cilj da pronađe grupe objekata (klastera) tako da su rastojanja između objekata unutar klastera minimizovana dok su rastojanja između klastera velika.

Među važnijim tehnikama klasterovanja izdvajamo:

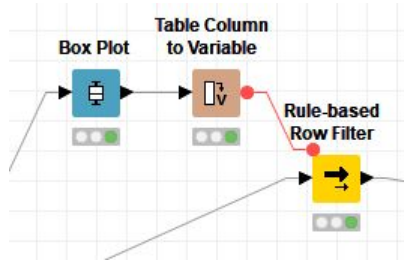
- K-sredina (*k-means clustering*)
- Hijerarhijsko klasterovanje (*Hierarchical clustering*)
- DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*)

3.1 Cilj klasterovanja

Usresredićemo se na „Social Power NBA” tj. društveni aspekt NBA igrača. Pokušaćemo da na osnovu „Twitter” pratilaca i postignutih koševa iz igre grupišemo igrače koji su međusobno slični a da su igrači iz različitih grupa međusobno različiti. Za tu svrhu koristimo klaster analizu.

3.2 Priprema podataka za klasterovanje

Najpre izdvajamo potrebne attribute upotrebom čvora *Column Filter* a zatim i normalizujemo podatke uz pomoć čvora *Normalizer*. Važno je eliminisati vrednosti van granica koje bi mogle negativno da se odraze na izračunavanje centra klastera (Slika 3.1).

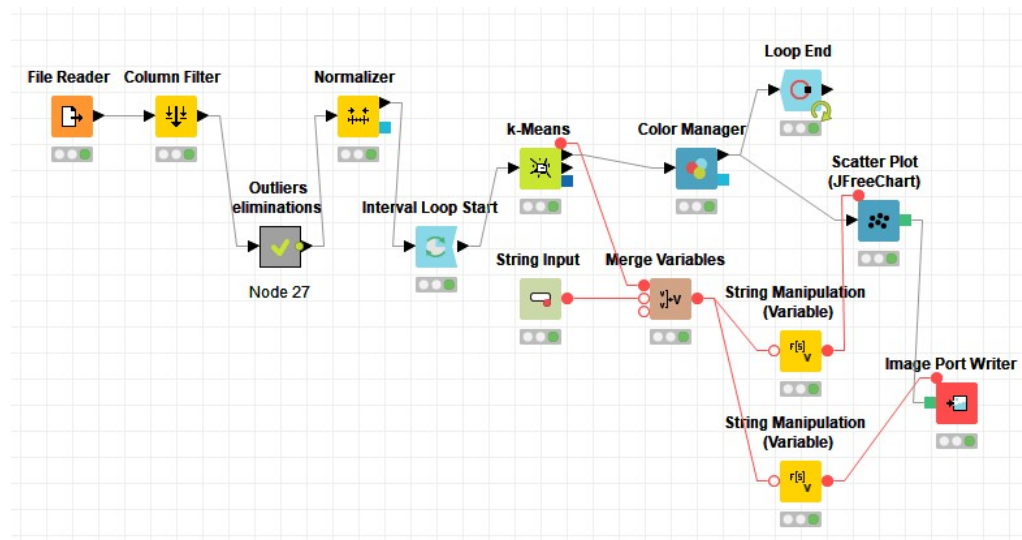


Slika 3.1: Eliminisanje vrednosti van granica

3.3 Tehnike klasterovanja

3.3.1 K-sredina

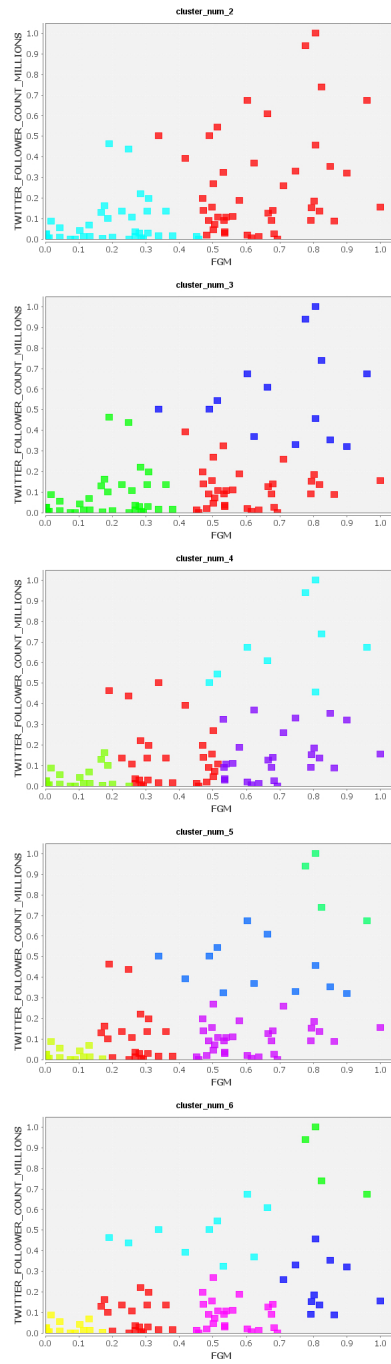
Algoritam *k-sredina* se zasniva na centru koji je reprezentativni predstavnik klastera. Broj klastera ćemo iterativno povećavati i analizirati. U daljoj obradi koristimo čvor *k-Means* koji pripadnost klasteru određuje na osnovu Euklidskog rastojanja (Slika 3.2).



Slika 3.2: KNIME: Primena *k-sredina* algoritma

Prodiskutovaćemo rezultat gde su igrači podeljeni u 6 klastera obojenih različitim bojama (Slika 3.3). Ovde se može uočiti jedan zanimljiv klaster:

- Klasteru tamno plave boje pripadaju igrači koji su postigli mnogo koševa iz igre a imaju mali broj „Twitter” pratilaca. Tom klasteru pripada igrač „Karl-Anthony Towns” koji ima najviše postignutih koševa iz igre a vrlo malo „Twitter” pratilaca.



Slika 3.3: Razlicit broj klastera

3.3.2 Hijerarhijsko klasterovanje

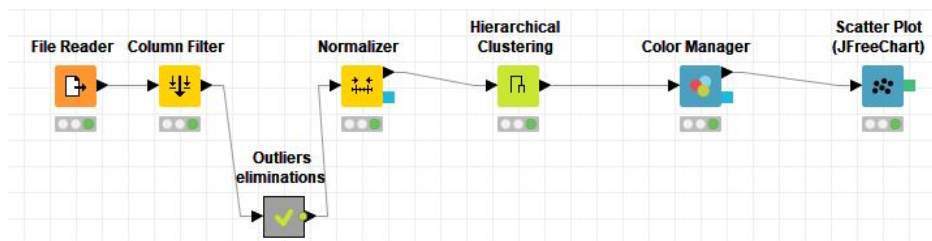
Hijerarhijsko klasterovanje delimo na *klasterovanje spajanjem* i *klasterovanje razdvajanjem*. *Klasterovanje spajanjem* spaja najbliže klustere sve dok se ne dobije jedan sveobuhvatni klaster. Na početku svaka tačka predstavlja po jedan klaster dok *klasterovanje razdvajanjem* ima obratni tok.

Čvor *Hierarchical clustering* (Slika 3.4) implementira *klasterovanje spajanjem* i njega primenjujemo na isti skup podataka koji smo prethodno pripremili. U postavkama čvora željeni broj klastera postavljamo na 5 dok za funkciju razdaljine biramo Euklidsko rastojanje.

Pored prethodne dve postavke postoji i postavka *tip povezivanja* (*Linkage type*) gde definišemo sličnost klastera. Biramo jedan od 3 sledećih vrednosti:

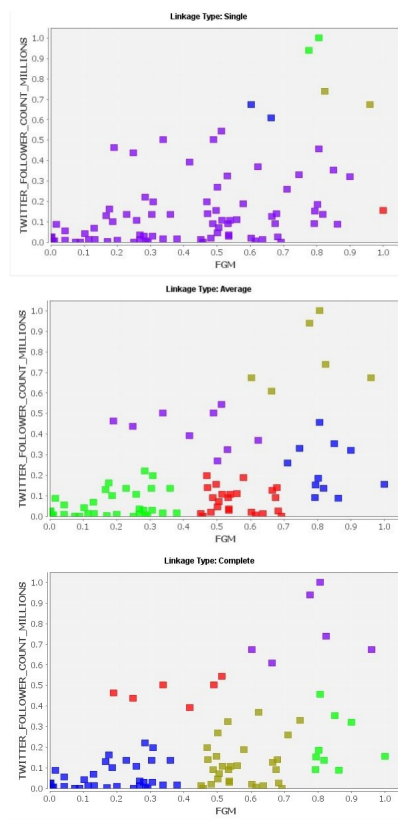
- Single Linkage - *MIN* (sličnost između klastera C1 i klastera C2 definišemo kao minimalnu razdaljinu između bilo koje 2 tačke x i y gde x pripada C1 klasteru a y C2 klasteru)
- Complete Linkage - *MAX* (sličnost između klastera C1 i klastera C2 definišemo kao maksimalnu razdaljinu između bilo koje 2 tačke x i y gde x pripada C1 klasteru a y C2 klasteru)
- Average Linkage - *Prosek* (sličnost između klastera C1 i klastera C2 definišemo kao prosečnu razdaljinu između svih parova tačaka iz C1 i C2)

Ove vrednosti ćemo smenjivati i uporediti (Slika 3.5).



Slika 3.4: KNIME: Primena hijerarhijskog klasterovanja

Ovde se izdvaja prvi rezultat klasterovanja gde je uzet MIN kao definicija sličnosti klastera.



Slika 3.5: Različit tip povezivanja

Poglavlje 4

Klasifikacija

Problem klasifikacije intuitivno možemo da predstavimo na sledeći način: Na osnovu datih slogova iz skupa podataka za trening, pri čemu je svakom slogu pored osnovnog skupa atributa pridružena i oznaka klase, treba odrediti oznaku klase za slogove iz skupa podataka za test koji prethodno nisu viđeni. Preciznije, potrebno je da nađemo ciljnu funkciju, tj. model klasifikacije koji preslikava osnovni skup atributa u specijalni atribut za oznaku klase. Ulazne podatke delimo na:

- Podatke za trening - pomoću kojih se formira model
- Podatke za testiranje - koji se koriste za proveru ispravnosti modela

4.1 Cilj klasifikacije

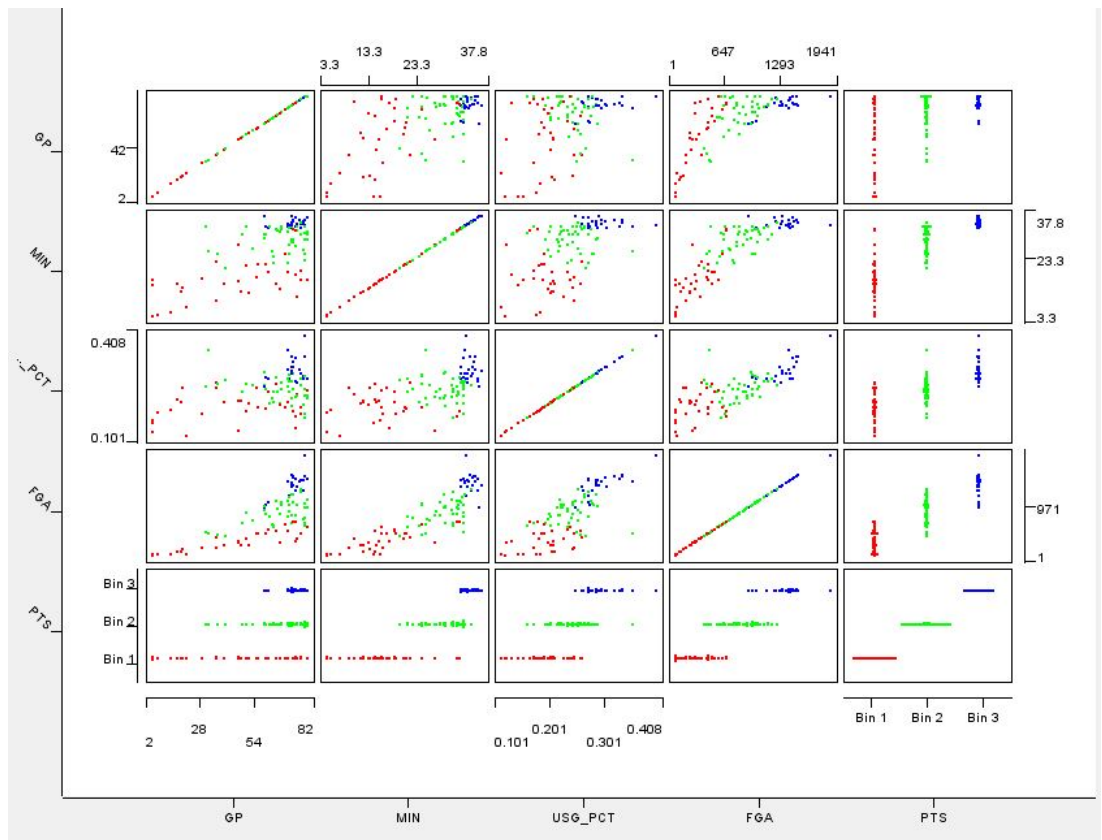
Cilj klasifikacije na našem skupu podataka je da napravimo predikciju prosečnih poena po utakmici - *PTS* na osnovu odabranih atributa.

4.2 Priprema podataka za klasifikaciju

4.2.1 Analiza atributa

Najpre ćemo koristeći čvor *Auto-Binner* kategorisati atribut *PTS* u 3 kategorije (Bin 1, Bin 2, Bin 3). Dalje je potrebno izvršiti selekciju atributa tj. odabrati podskup relevantnih atributa. Cilj nam je da eliminišemo attribute koji nam ne donose dodatne informacije ili nam nisu od značaja.

Detaljnijom analizom atributa zapazili smo sledeća 4 relevantna atributa (Slika 4.1):



Slika 4.1: Selekcija atributa

- *GP* - broj odigranih utakmica
- *MIN* - prosečno odigranih minuta po utakmici
- *USG_PCT* - posed lopte igrača u odnosu na posed lopte celog tima
- *FGA* - broj bacanja na koš iz igre

4.2.2 Podela podataka

Ciljnu funkciju tj. model klasifikacije gradimo na osnovu skupa podataka za trening a zatim proveravamo ispravnost modela na skupu podataka za testiranje. Za podelu podataka koristimo *Partitioning* čvor u *KNIME* alatu i *Partition* čvor u *SPSS* alatu kako bi skup podataka podelili na gore navedene trening i test skupove podataka u odnosu 70 naprema 30.

4.3 Tehnike klasifikacije

4.3.1 Metode zasnovane na drvetima odlučivanja

Ova tehnika je zasnovana na drvetu kroz koje se krećemo odgovarajući na pitanja sve dok ne dođemo do lista drveta koji predstavlja oznaku klase. Za konstruisanje drveta koriste se neki od navedenih algoritama:

- Hantov algoritam
- CART (*Classification And Regression Trees*)
- ID3 (*Iterative Dichotomiser 3*)
- C4.5
- SLIQ
- SPRINT (*Scalable PaRallelizable INduction and decision Tress*)

U *KNIME* alatu postoji čvor *Decision Tree Learner* koji konstruiše drvo odlučivanja. Zasniva se na C4.5 i SPRINT algoritmu. Na ulaz se prosleđuje skup podataka za trening dok se inicijalizacija sastoji od parametara od kojih ćemo izdvojiti neke najvažnije:

- Ciljni atribut (*Class column*)
- Mera podele (*Quality measure*)

- Minimalni broj slogova u čvoru (*Min number records per node*)

Za ciljni atribut postavljamo kategorizovanu klasu *PTS*.

Pitanja na koja odgovaramo prilikom kretanja kroz drvo su uslovi test atributa. Pri određivanju uslova možemo posmatrati homogenost klasa kod čvorova dece tj. meru nečistoće.

Decision Tree Learner nam nudi *Gini index*

$$Gini_{split} = \sum_{i=1}^k \frac{n_i}{n} Gini(i)$$

$$Gini(t) = 1 - \sum_j [p(j|t)]^2$$

i *Gain ratio*

$$GainRatio_{split} = \frac{Gain_{split}}{SplitInfo}$$

$$Gain_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

$$SplitInfo = - \sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

$$Entropy(t) = - \sum_j p(j|t) \log_2 p(j|t)$$

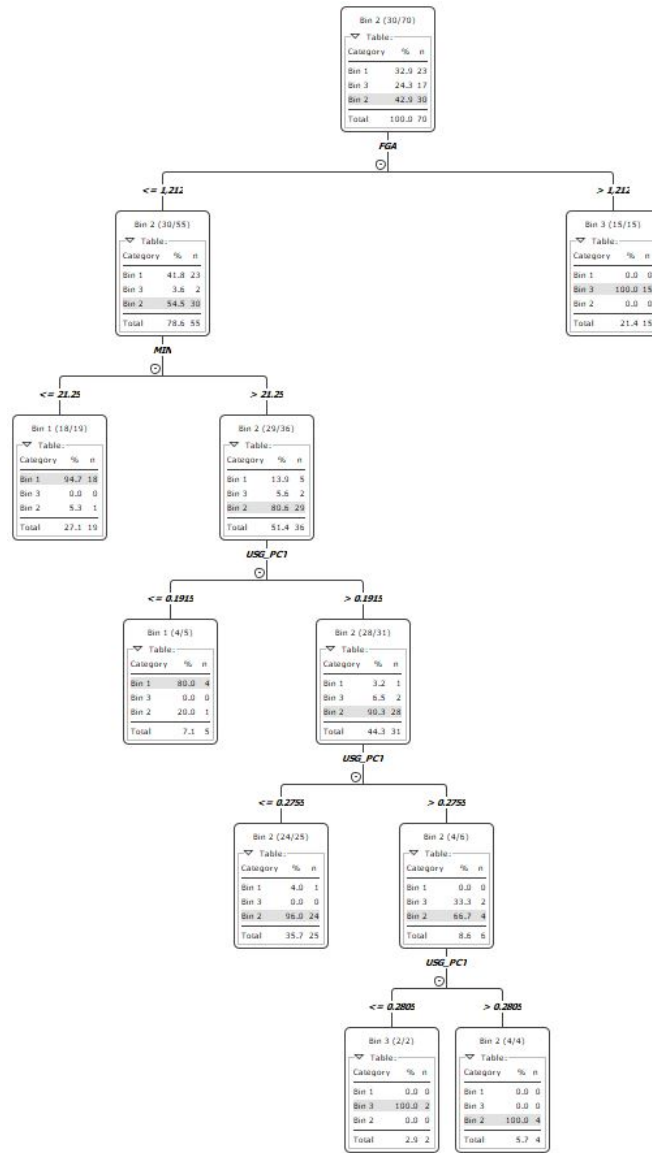
gde $p(j|t)$ predstavlja relativnu frekvenciju klase j u čvoru t , n_i broj slogova u dete čvoru i , n broj slogova u čvoru p a k broj dece čvora p .

Za vrednost ovog parametra uzimamo *Gini index*.

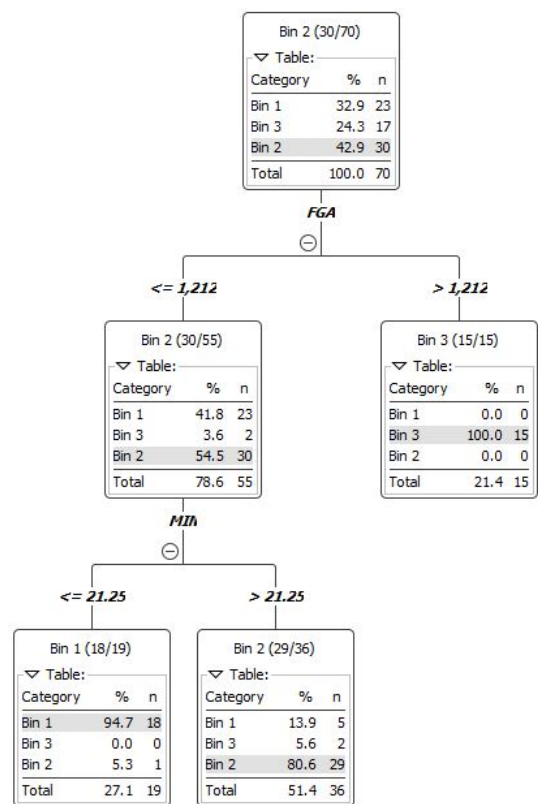
Minimalni broj slogova u čvoru utiče na veličinu drveta (Slika 4.2 i 4.3). Ovaj parametar ćemo iterativno povećavati u intervalu od 2 do 10 za korak 1 i na kraju uporediti sve rezultate.

Model koji dobijamo kao rezultat dalje primenjujemo na skup podataka za test. U tu svrhu koristimo čvor *Decision Tree Predictor*. Kao rezultat ovog čvora dobijamo podatke sa predikcijom klase. Pomoću čvora *Score* izvlačimo informacije o uspešnosti modela predstavljene kroz matricu konfuzije i meru preciznosti.

Matrica konfuzije se sastoji od četiri vrednosti: TP (*True Positive*), FP



Slika 4.2: Minimalni broj slogova u čvoru je 2



Slika 4.3: Minimalni broj slogova u čvoru je 10

(*False Positive*), TN (*True Negative*) i FN (*False Negative*). U našem slučaju ako za *minimalni broj slogova u čvoru* uzmemo vrednost 10 dobijamo matricu (Slika 4.4) sa sledećim tumačenjem:

- TP - broj slogova kojima je prepoznata klasa C a koji pripadaju toj klasi
- FP - broj slogova kojima je prepoznata klasa C a koji ne pripadaju toj klasi
- TN - broj slogova kojima nije prepoznata klasa C a koji ne pripadaju toj klasi
- FN - broj slogova kojima nije prepoznata klasa C a koji pripadaju toj klasi

Ako za klasu C uzmemo Bin 3 možemo videti da je toj klasi dodeljeno 7 slogova, pri čemu je 5 dodeljeno tačno ($TP = 5$) dok 2 nisu tačno dodeljena ($FP = 2$). Od ostalih 23 nedodeljenih slogova 2 nisu dodeljena a trebalo je biti ($FN = 2$) dok ostalih 21 ne pripadaju toj klasi i nisu joj dodeljeni ($TN = 21$).

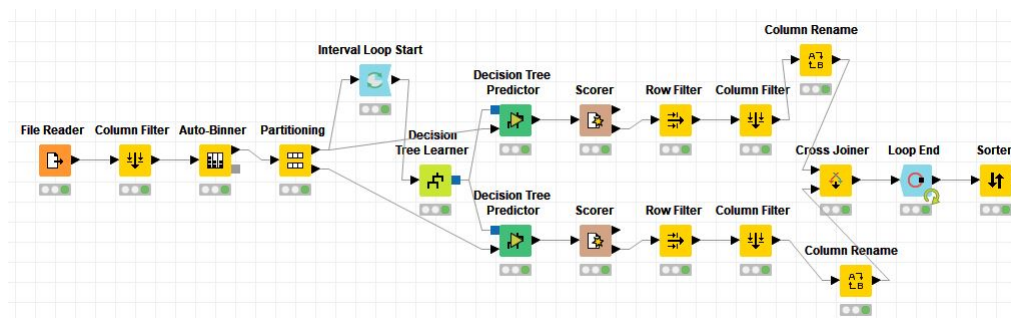
Row ID	Bin 1	Bin 2	Bin 3
Bin 1	10	0	0
Bin 2	0	11	2
Bin 3	0	2	5

Slika 4.4: Drvo odlučivanja: Matrica konfuzije

Na osnovu matrice konfuzije računamo preciznost (*accuracy*) modela po sledećoj formuli:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Pokretanjem algoritma za sve iteracije (Slika 4.5) dobijamo tabelu sa preciznošću modela za svaku iteraciju (Slika 4.6). Pored skupa podataka za test, model smo primenili i na skup podataka za trening. Razlog tome jeste što u našem skupu podataka ima malo slogova (100) pa zbog toga može doći do preprilagođavanja modela.



Slika 4.5: KNIME: Drvo odlučivanja

Row ID	D AccuracyTraining	D AccuracyTest	I Iteration
Overall_Over...	0.957	0.8	0
Overall_Over...	0.943	0.733	1
Overall_Over...	0.929	0.8	2
Overall_Over...	0.929	0.8	3
Overall_Over...	0.914	0.8	4
Overall_Over...	0.9	0.8	5
Overall_Over...	0.886	0.867	6
Overall_Over...	0.886	0.867	7
Overall_Over...	0.886	0.867	8

Slika 4.6: Drvo odlučivanja: Preciznost

4.3.2 Statistički zasnovane metode

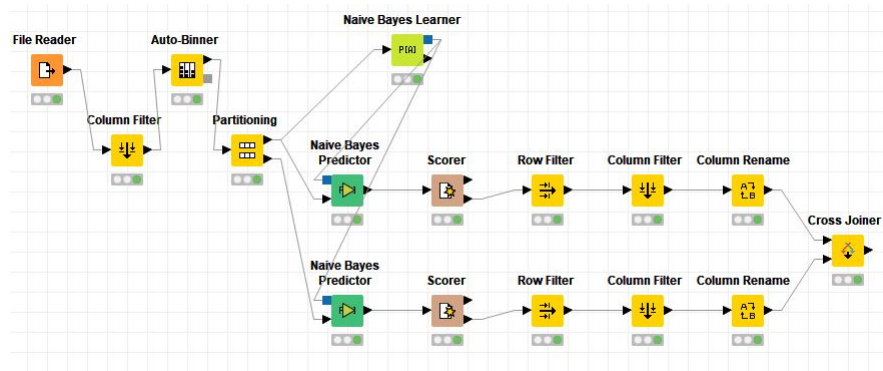
Bajesov klasifikator

Bajesov klasifikator se zasniva na Bajesovoj teoremi:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

gde je $P(Y|X)$ skraćenica za $P(Y = y|X = x)$ i predstavlja uslovnu verovatnoću tj. verovatnoću da će Y uzeti vrednost y onda kada X ima vrednost x .

U *KNIME* alatu postoji čvor *Naive Bayes Learner* koji ćemo koristiti za kreiranje modela. U sledećem koraku koristimo *Naive Bayes Predictor* gde primenjujemo model na skup podataka za trening i na skup podataka za test (Slika 4.7).



Slika 4.7: KNIME: Bajesov klasifikator

Analogno kao i kod drvetva odlučivanja analiziramo rezultat (Slika 4.8 i 4.9).

Row ID	Bin 1	Bin 2	Bin 3
Bin 1	8	2	0
Bin 2	3	9	1
Bin 3	0	1	6

Slika 4.8: Bajesov klasifikator: Matrica konfuzije

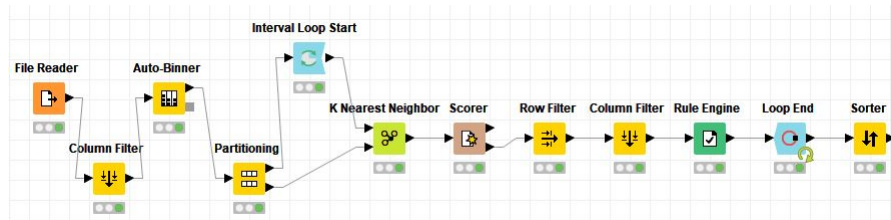
Row ID	D AccuracyTraining	D AccuracyTest
Overall_Overall	0.886	0.833

Slika 4.9: Bajesov klasifikator: Preciznost

4.3.3 Metode zasnovane na instancama

Klasifikacija pomoću najbližeg suseda

Klasifikacija se vrši na osnovu unapred difinisan broj najbližih suseda koji inkrementiramo za 1 u granicama od 3 do 30. Koristimo čvor *K Nearest Neighbor* (Slika 4.10).



Slika 4.10: KNIME: KNN

Row ID	Bin 1	Bin 2	Bin 3
Bin 1	8	2	0
Bin 2	3	10	0
Bin 3	0	1	6

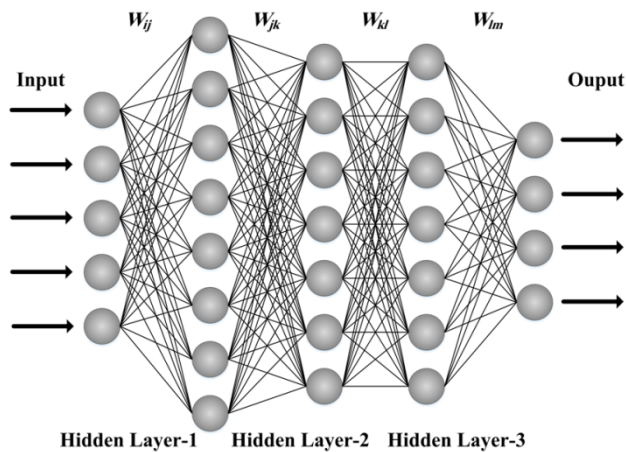
Slika 4.11: KNN: Matrica konfuzije

Row ID	D Accuracy	k	Iteration
Overall#2	0.8	5	2
Overall#3	0.8	6	3
Overall#4	0.8	7	4
Overall#5	0.8	8	5
Overall#6	0.8	9	6
Overall#7	0.8	10	7
Overall#8	0.8	11	8
Overall#12	0.8	15	12
Overall#13	0.8	16	13
Overall#14	0.8	17	14
Overall#15	0.8	18	15
Overall#16	0.8	19	16
Overall#17	0.8	20	17
Overall#18	0.8	21	18
Overall#19	0.8	22	19
Overall#20	0.8	23	20
Overall#21	0.8	24	21
Overall#24	0.8	27	24
Overall#25	0.8	28	25
Overall#26	0.8	29	26
Overall#27	0.8	30	27
Overall#0	0.767	3	0
Overall#1	0.767	4	1
Overall#9	0.767	12	9
Overall#10	0.767	13	10
Overall#11	0.767	14	11
Overall#22	0.767	25	22
Overall#23	0.767	26	23

Slika 4.12: KNN: Preciznost

4.3.4 Neuronske mreže

Ideja neuronske mreže je u oponašanju strukture ljudskog mozga. Tako su neuroni predstavljeni čvorovima dok su dendriti veza između tih čvorova. Najjednostavniji model neuronske mreže jeste perceptron (4.13). Mi ćemo koristiti neuronsku mrežu sa propagacijom unapred.

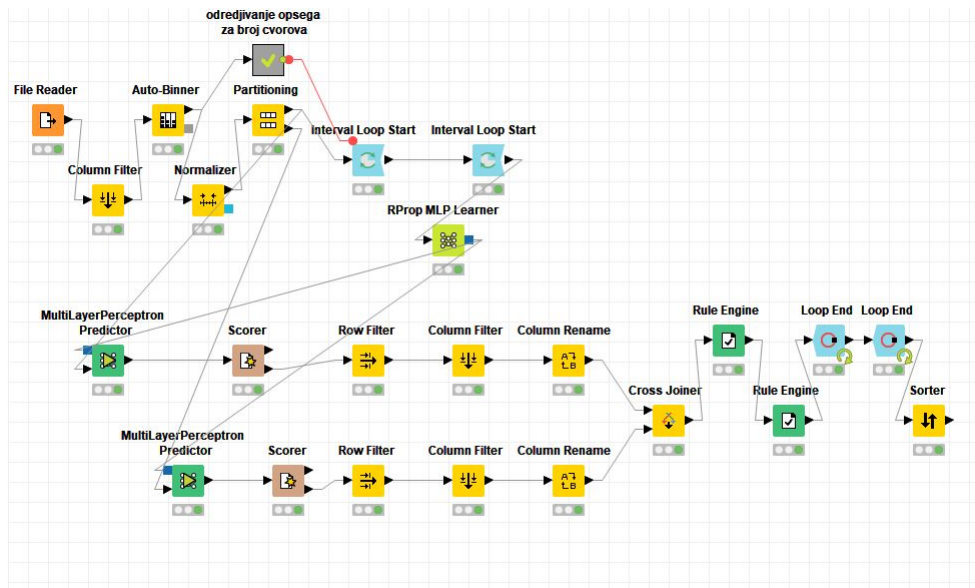


Slika 4.13: Višeslojni perceptron

Čvor *Rprop MLP Learner* u *KNIME* alatu implementiramo upravo taj tip neuronske mreže. Za podešavanje ovog čvora imamo sledeće bitnije parametre:

- Maksimalan broj iteracija (*Maximum number of iterations*)
- Broj skrivenih slojeva (*Number of hidden layers*)
- Broj skrivenih neurona po sloju (*Number of hidden neurons per layer*)

Broj skrivenih slojeva i skrivenih neurona po sloju ćemo iterativno povećavati za korak 1. Granice u prvom slučaju će nam biti od 1 do 5 dok ćemo u drugom slučaju koristiti formulu sa vežbi. Za razliku od prethodnih algoritama, ovaj algoritam zahteva normalizovane ulazne podatke pa smo u tu svrhu dodali čvor *Normalizer*.



Slika 4.14: KNIME: RProp MLP Learner

Row ID	Bin 1	Bin 2	Bin 3
Bin 1	10	0	0
Bin 2	2	11	0
Bin 3	0	0	7

Slika 4.15: NN: Matrica konfuzije

Sorted Table - 0:51 - Sorter

File Hilite Navigation View

Table "default" - Rows: 25 Spec - Columns: 6 Properties Flow Variables

Row ID	D Accura...	D Accura...	layers	nodes	Iteration	Iteratio...
Overall_Over...	0.929	0.967	4	4	3	1
Overall_Over...	0.929	0.967	5	4	4	1
Overall_Over...	0.9	0.967	1	5	0	2
Overall_Over...	0.929	0.967	2	5	1	2
Overall_Over...	0.943	0.967	3	5	2	2
Overall_Over...	0.886	0.967	1	7	0	4
Overall_Over...	0.9	0.933	2	3	1	0
Overall_Over...	0.886	0.933	3	3	2	0
Overall_Over...	0.957	0.933	1	4	0	1
Overall_Over...	0.914	0.933	2	4	1	1
Overall_Over...	0.929	0.933	3	4	2	1
Overall_Over...	0.886	0.933	2	6	1	3
Overall_Over...	0.914	0.933	3	6	2	3
Overall_Over...	0.9	0.933	3	7	2	4
Overall_Over...	0.929	0.933	5	7	4	4
Overall_Over...	0.886	0.9	4	3	3	0
Overall_Over...	0.943	0.9	5	3	4	0
Overall_Over...	0.886	0.9	1	6	0	3
Overall_Over...	0.971	0.9	4	6	3	3
Overall_Over...	0.886	0.867	1	3	0	0
Overall_Over...	0.914	0.867	4	5	3	2
Overall_Over...	0.9	0.867	5	6	4	3
Overall_Over...	0.914	0.833	5	5	4	2
Overall_Over...	0.886	0.833	2	7	1	4
Overall_Over...	0.914	0.833	4	7	3	4

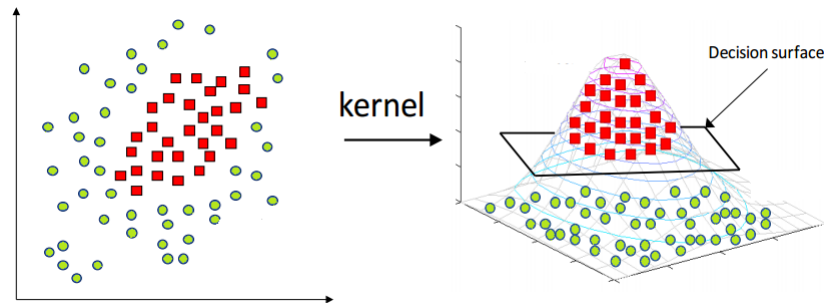
Slika 4.16: NN: Preciznost

4.3.5 Metode zasnovane na podržavajućim vektorima

Metode zasnovane na podržavajućim vektorima (*SVM - Support Vector Machine*) je još jedna od metoda za klasifikaciju podataka koja je bazirana na ideji vektorskih prostora. Model je predstavljen kroz formulu. Cilj je naći hiper-ravan tako da su svi podaci iz iste klase sa iste strane ravni. Ovaj algoritam je osnova za binarnu klasifikaciju.

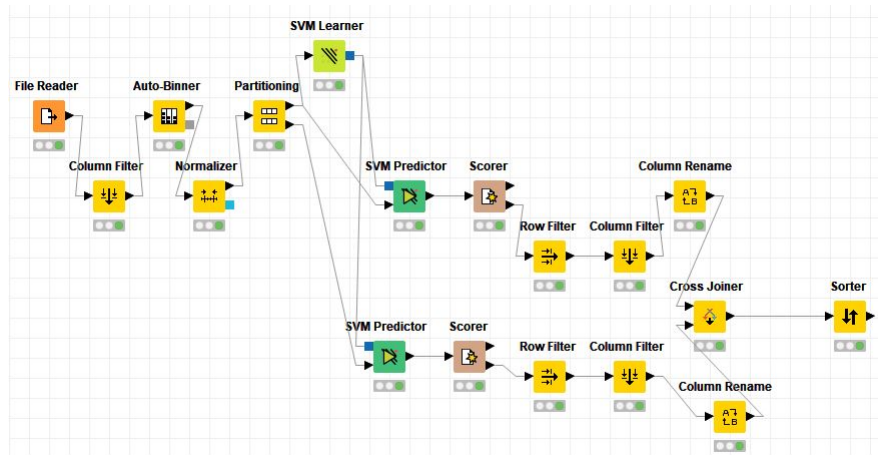
KNIME nam za ovu vrstu klasifikacije nudi čvor *SVM Learner* (slika 4.18) kod koga je potrebno odabrati jedan od ponuđenih kernela. Ideja kernel funkcije je da preslika podatke u prostor veće dimenzije gde je moguće linearno razdvojiti klase (Slika 4.17).

- Polinomijalan kernel (*Polynomial*): $K(X_i, X_j) = (X_i X_j + c)^q$
- Sigmoid kernel (*HyperTangent*): $K(X_i, X_j) = \tanh(\alpha x_i x_j - b)$
- Gausov kernel (*RBF*): $K(X_i, X_j) = e(-\frac{\|X_i - X_j\|^2}{2\sigma^2})$



Slika 4.17: Kernel

Analizom rezultata (Slika 4.19-4.21) možemo uporediti preciznost modela za odabran kernel. Vidimo da Gausov kernel daje najveću preciznost tj. najbolje klasifikuje podatke.



Slika 4.18: KNIME: SVM

Row ID	D AccuracyTraining	D AccuracyTest
Overall_Overall	0.871	0.833

Slika 4.19: Polinomijalan kernel: preciznost

Row ID	D AccuracyTraining	D AccuracyTest
Overall_Overall	0.6	0.6

Slika 4.20: Sigmoid kernel: preciznost

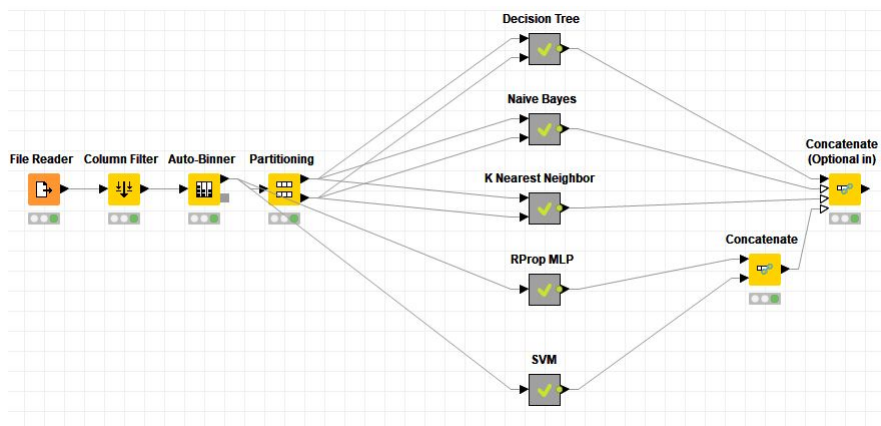
Row ID	D AccuracyTraining	D AccuracyTest
Overall_Overall	0.957	0.867

Slika 4.21: Gausov kernel: preciznost

4.4 Rezime

4.4.1 KNIME

Kako smo gore obradili metode klasifikacije, sada želimo da te metode uporedimo i vidimo koja nam daje najbolji rezultat (Slika 4.22). Kako primenom svih ovih metoda imamo više čvorova *Partitioning* potrebno je da podesimo isti *random seed* kako bi razdvajanje na trening i test podatke bilo isto za svaki metod.



Slika 4.22: Upoređivanje tehnika klasifikacije

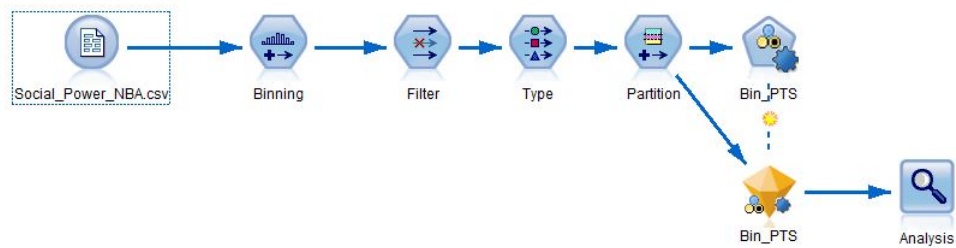
Analizom vrednosti *preciznost* za test podatke, u tabeli navedena kao *AccuracyTest* (Slika 4.23), zaključujemo da metode uglavno daju sličnu preciznost. Bajesov klasifikator, klasifikator zasnovan na najbližim susedima i neuronske mreže daju jednaku preciznost (0.967). Za nijansu su lošiji klasifikator zasnovan na drvetima odlučivanja (0.933) i klasifikator zasnovan na podržavajućim vektorima (0.867)

Row ID	S Metod	D AccuracyTraining	D AccuracyTest	Iteration	k	layers	nodes	Iteratio...
Overall_Overall	Naive Bayes	0.829	0.967	?	?	?	?	?
Overall#6	KNN	?	0.967	6	9	?	?	?
Overall_Over...	RProp MLP	0.929	0.967	3	?	4	3	0
Overall_Over...	Decision Tree	0.9	0.933	4	?	?	?	?
Overall_Over...	SVM	0.943	0.867	?	?	?	?	?

Slika 4.23: Gausov kernel: preciznost

4.4.2 SPSS

Alat *SPSS* ima čvor *Auto Classifier* koji objedinjuje različite metode klasifikacije. U podešavanju biramo prethodno navedene metode. U našem slučaju izabrali smo *C5*, *Bayesian Network*, *KNN*, *SVM* i *Neural Net*. Analogno kao i u *KNIME* alatu, izvršili smo selekciju atributa, ciljni atribut kategorisali u 3 klase a zatim u čvoru *Type* taj atribut definisali kao *Target* (Slika 4.24).



Slika 4.24: SPSS: Auto Classifier

Za rezultat dobija se lista već pomenutih metoda sortirana po vrednosti *preciznost*. Kao i u *KNIME* alatu neuronske mreže se pokazuju kao najbolje (Slika 4.25).

Use?	Graph	Model	Build Time (mins)	Overall Accuracy (%)
<input checked="" type="checkbox"/>		Neural Net 1	< 1	90.323
<input checked="" type="checkbox"/>		C5 1	< 1	87.097
<input checked="" type="checkbox"/>		SVM 1	< 1	87.097
<input checked="" type="checkbox"/>		KNN Algorithm 1	< 1	83.871
<input checked="" type="checkbox"/>		Bayesian Network...	< 1	64.516

Slika 4.25: Rezultati