

Izveštaj Ekspertskog sistema

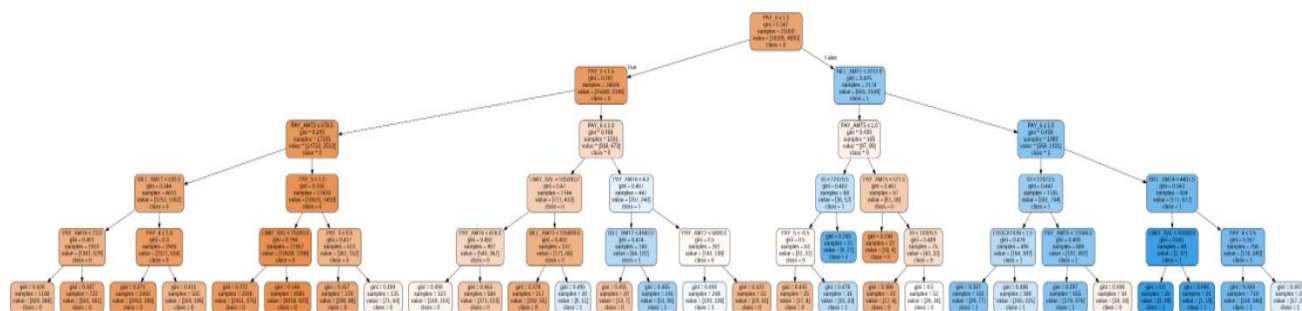
1. Opis problema

Predstavljen je problem klasifikacije klijenata banke prema tome da li hoće ili neće da isplate kredit. Skup podataka klijenata sadrži finansijske podatke kao i isplaćenja klijenata prethodnih meseci. Na osnovu ovih podataka se može obučiti neki od algoritama mašinskog učenja kako bi smo zamenili stanje eksperta.

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default.payment.next.month
0	1	20000.0	2	2	1	24	2	2	-1	-1	-2	-2	3913.0	3102.0	609.0	0.0	0.0	0.0	609.0	0.0	0.0	0.0	0.0	0.0	1
1	2	120000.0	2	2	2	26	-1	2	0	0	0	2	2682.0	1725.0	2682.0	3272.0	3455.0	3261.0	0.0	1000.0	1000.0	1000.0	0.0	2000.0	1
2	3	90000.0	2	2	2	34	0	0	0	0	0	0	29239.0	14027.0	13559.0	14331.0	14948.0	15549.0	1518.0	1500.0	1000.0	1000.0	1000.0	5000.0	0
3	4	50000.0	2	2	1	37	0	0	0	0	0	0	46990.0	48233.0	49291.0	28314.0	28959.0	29547.0	2000.0	2019.0	1200.0	1100.0	1069.0	1000.0	0
4	5	50000.0	1	2	1	57	-1	0	-1	0	0	0	8617.0	5670.0	35835.0	20940.0	19146.0	19131.0	2000.0	36681.0	10000.0	9000.0	689.0	679.0	0

Idealan metod je klasifikacija stablima odlučivanja, zato što iz njih možemo izvući pravila kojima ćemo podhraniti ekspertski sistem. Stabla odlučivanja su lako interpretabilna i laka za razumevanje. Jedno pravilo će biti formirano tako što se iz stabla odlučivanja izvuče put od korena do jednog lista. Ovim metodom bi se prikupila sva pravila ako izvučemo sve putanje od korena do lista jednog stabla odlučivanja.

Ispod je prikaz stabla odlučivanja maksimalne dubine 5 nivoa što će dovesti do formiranja stabla sa 30 listova, što znači da ćemo imati stablo od 30 pravila (U stablu je uvek jedinstven put od korena do lista)



Stablo je učeno **Gini** metodom koja je inače podrazumevana za učenje stabla i slična je entropiji, samo što se umesto entropije u **Gini** metodi koriste uslovne verovatnoće, ali su jako slične. S obzirom da je maksimalna dubina stabla 5 nivoa, stavili smo parameter da podela uzorka po listovima ne bude manja od 12 kako ne bi smo imali previše malo uzorka po listu i da ne bi došlo do neobučenosti (**underfitting**).

Postoje razni načini za prikupljanje naučenih pravila, u ovom slučaju putanja od korena do lista u stablu odlučivanja. Ovde se se iskoristila metoda `export_text` iz `sklearn` biblioteke za prikaz ugnježdenih pravila, a posebna funkcija je kreirana kako bi se izvukle putanje iz stabla i konvertovala u pravila kojima će biti napunjen ekspertski sistem.

2. Rezultati učenja

Za simulaciju ekspertskog znanja se koristio klasifikator stable odlučivanja **DecisionTreeClassifier** iz **sklearn.tree** paketa.

```
from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier(criterion='gini', max_depth=5, min_samples_leaf=12)
dtree.fit(X_train,y_train)

DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                      max_depth=5, max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=12, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=None, splitter='best')
```

Nakon obučavanja, korišćeni su test podaci za ocenu greške modela stable odlučivanja. Skor preciznosti (malo iznad 82%) modela kao i testiranja modela je prikazan ispod.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

score1=accuracy_score(y_val,predictions)
print("Accuracy score: " + str(score1))

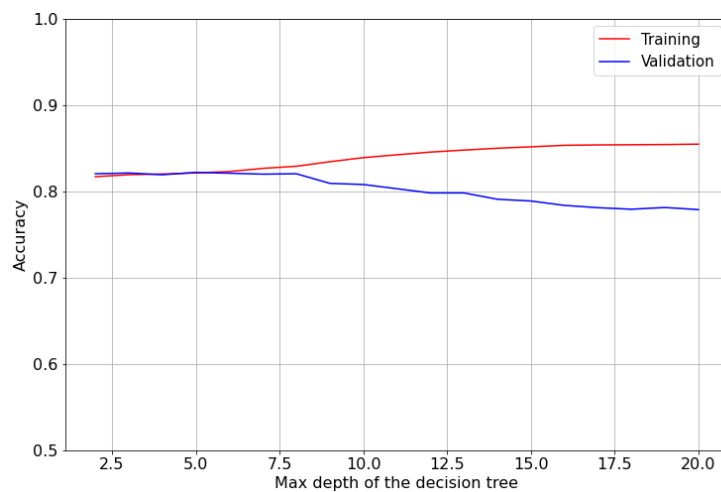
Accuracy score: 0.822
```

Prikaz naučenih pravila u eksportovanom obliku iz modela stable odlučivanja je dat ispod kao ugnježđena struktura.

```
from sklearn.tree.export import export_text
tree_rules = export_text(dtree, feature_names=list(X_train))
print(tree_rules)

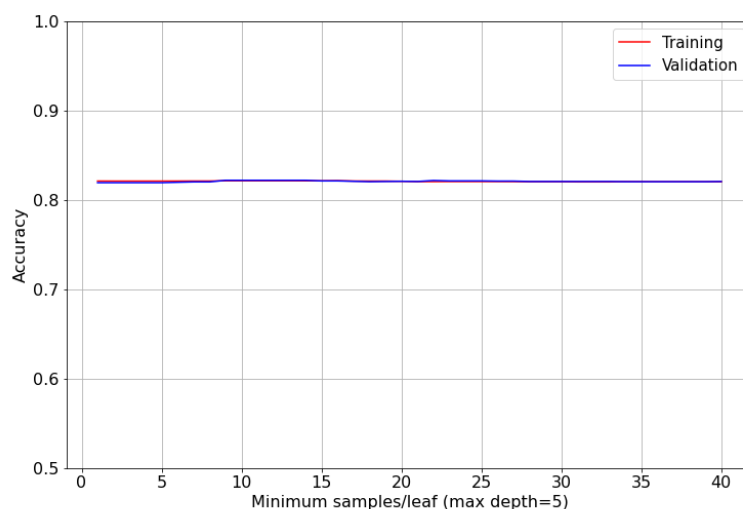
|--- PAY_0 <= 1.50
|   |--- PAY_2 <= 1.50
|   |   |--- PAY_AMT3 <= 678.50
|   |   |   |--- BILL_AMT1 <= 530.00
|   |   |   |   |--- PAY_AMT6 <= 73.00
|   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- PAY_AMT6 > 73.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- BILL_AMT1 > 530.00
|   |   |   |   |   |--- PAY_4 <= 1.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- PAY_4 > 1.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- PAY_AMT3 > 678.50
|   |   |   |   |--- PAY_5 <= 1.00
|   |   |   |   |   |--- LIMIT_BAL <= 75000.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- LIMIT_BAL > 75000.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- PAY_5 > 1.00
|   |   |   |   |   |--- PAY_3 <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- PAY_3 > 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |--- PAY_2 > 1.50
|   |   |   |--- PAY_6 <= 1.00
|   |   |   |   |--- LIMIT_BAL <= 185000.00
|   |   |   |   |   |--- PAY_AMT6 <= 418.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- PAY_AMT6 > 418.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- LIMIT_BAL > 185000.00
|   |   |   |   |   |--- BILL_AMT4 <= 272297.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- BILL_AMT4 > 272297.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- PAY_6 > 1.00
|   |   |   |   |--- PAY_AMT6 <= 4.00
|   |   |   |   |   |--- BILL_AMT1 <= 3989.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- BILL_AMT1 > 3989.00
|   |   |   |   |   |   |--- class: 1
```

Testiranjem raznih vrednosti paramtera `max_depth` i `min_samples_leaf` smo procenili koje su najbolje vrednosti za ova 2 parametra modela. Ovim grafikonima se može proceniti koje bi vrednosti bile najidealnije za obučavanje modela stable odlučivanja kako ne bi došlo do neobučenosti ili pak preobučenosti.



Može se primetiti kako je bitno da dubina stable ne bude prevelika, jer u tom slučaju će preciznost na test skupu da opada, iako raste na trening skupu i ova pojava je preobučenost (Overfitting). Znači da je model previse kompleksan i da je dubina prevelika. Može se odrediti parameter `max_depth` tako da će stablo i ako se previse razvije da se posle potkreše.

Na slici ispod je prikaz grafikona gde za dubinu 5 pratimo vrednosti parametra `min_samples_leaf` (minimum uzoraka po jednom listu) gde ne bi valjalo imati premalo uzoraka ako je u pitanju velika dubina stable, što ovde i nije slučaj sa dubinom od 5 nivoa u stablu.



Sa prethodnih grafika se može zaključiti da je idealna dubina stabla 5 nivoa, dok je sa dubinom 5 skoro svejedno koliko će biti minimalno uzoraka po listu, mada je najidealnije negde između 10 i 15. U našem modelu je podešena maksimalna dubina na 5 i minimalno 12 uzoraka po listu.

3. Prikaz rada Ekspertskog Sistema

Ispod je prikaz rada konzolnog korisničkog interfejsa gde je izvršena interakcija između korisnika i ekspertskog Sistema, gde na kraju dobijamo rezultat nakon odgovora na par pitanja gde korisnik daje odgovore sa 'yes' ili 'no' premda je u pitanju binarno stablo odlučivanja.

```
PS C:\Users\38163\Desktop\Završni projekat - Default of Credit Card Clients> cd  
' 'c:\Users\38163\.vscode\extensions\ms-python.python-2020.8.109390\pythonFiles\I  
Expert System to predict will client go to default with loan payment.  
Answer on few questions with 'yes' or 'no' and you will get an answer!  
PAY_0 <= 1.5? yes  
PAY_2 <= 1.5? yes  
PAY_AMT3 <= 678.5? no  
PAY_5 <= 1.0? yes  
LIMIT_BAL <= 75000.0? no  
Client will go to default: NO  
PS C:\Users\38163\Desktop\Završni projekat - Default of Credit Card Clients> |
```