

1. **Skup podataka**

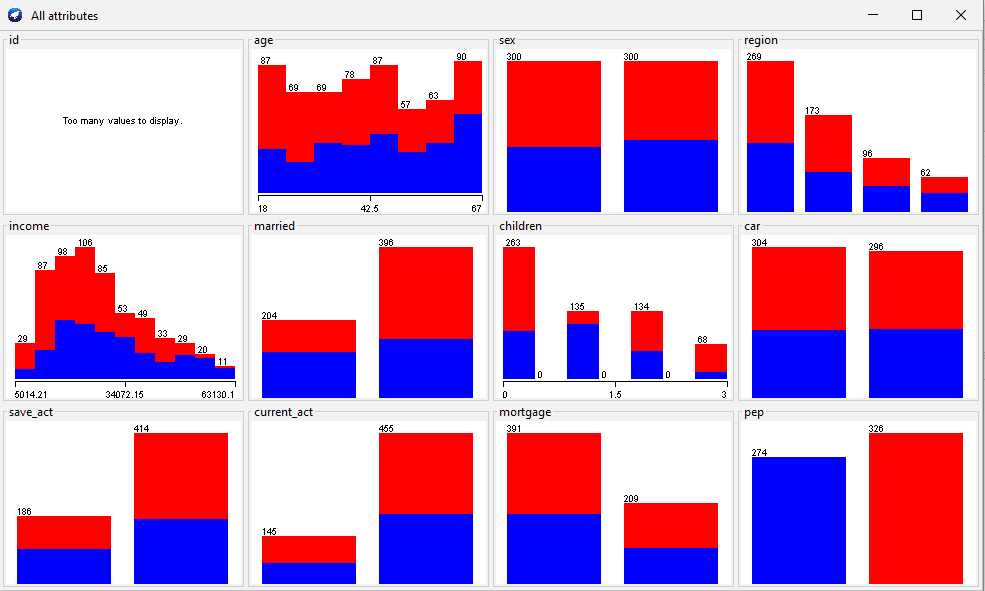
Postoje mnogi datasetovi gde je vec izvrsen data preprocessing i data cleaning, zato je uzet ovaj dataset kako bi izvrsili detaljnu analizu nad podacima:

<http://facweb.cs.depaul.edu/mobasher/classes/ect584/WEKA/data/bank-data.csv>

Sadrzaj dataseta:

|  |  |
| --- | --- |
| **id** | jedinstveni identifikacioni broj |
| **age** | starost klijenta u godinama (numerička vrednost) |
| **sex** | pol: male / female |
| **region** | region: inner\_city/rural/suburban/town |
| **income** | prihod klijenta: (numerička vrednost) |
| **married** | da li je klijent oženjen/udata: (yes/no) |
| **children** | broj dece (numeric) |
| **car** | da li klijent poseduje automobil (yes/no) |
| **save\_acct** | da li klijent ima štedni račun (yes/no) |
| **current\_acct** | da li klijent ima tekući račun (yes/no) |
| **mortgage** | da li klijent ima hipoteku (yes/no) |
| **pep** | da li je klijent kupio PEP (Personal Equity Plan) nakon poslednje pošiljke (yes/no) |

Detaljan pogled na sve atribute:

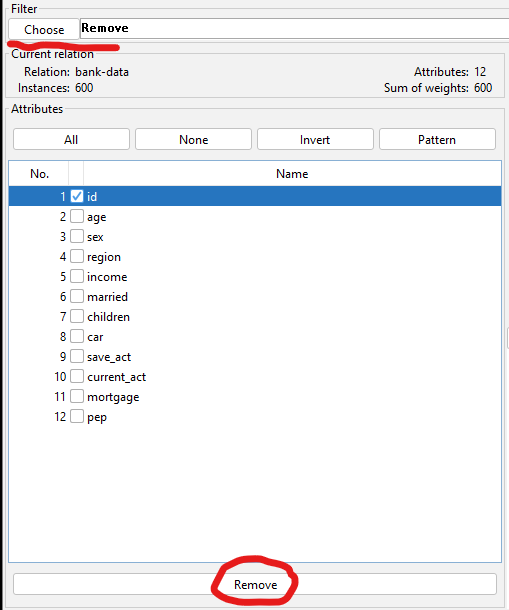


Cilj: Predvideti da li je klijent kupio PEP (yes/no) primenom klasifikacije

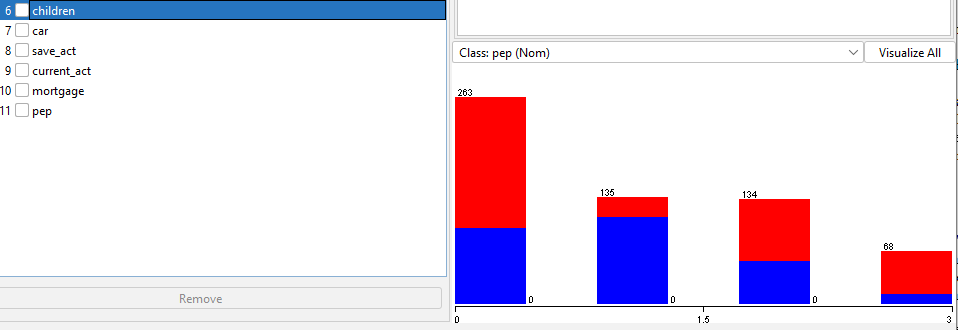
1. **Analiza podataka**

ID se koristi samo kao jedinstvena oznaka za identifikaciju svakog reda u skupu podataka, stoga nam nije potreban. Brisemo ga ovim postupkom: U “Filter” panelu, kliknemo Choose dugme. Onda nam se javaljaju sekcije sa svim filterima. Dalje weka -> filters -> unsupervised -> attribute -> Remove

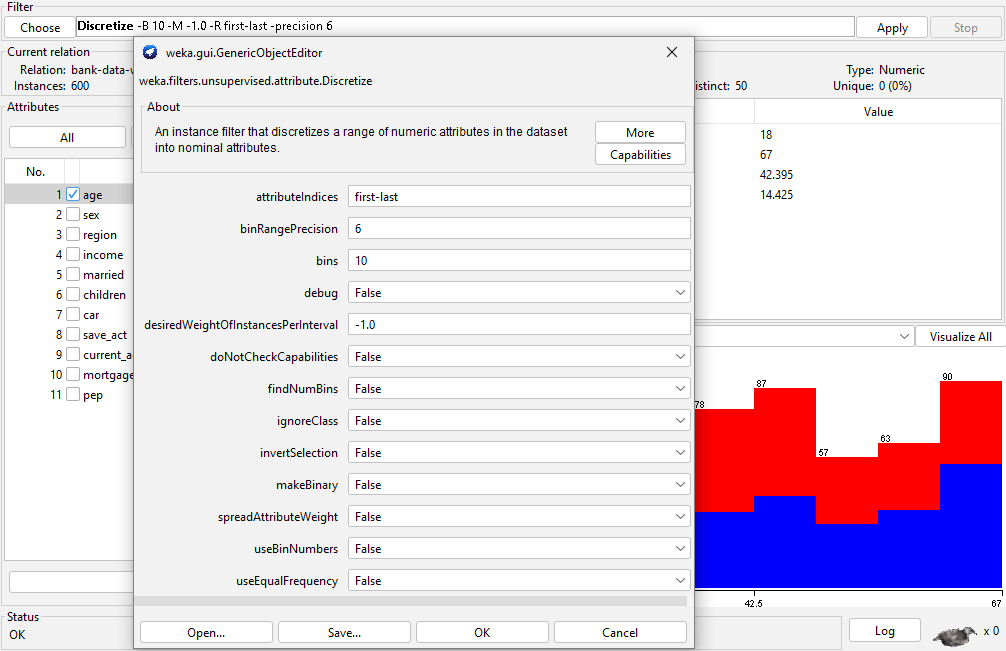
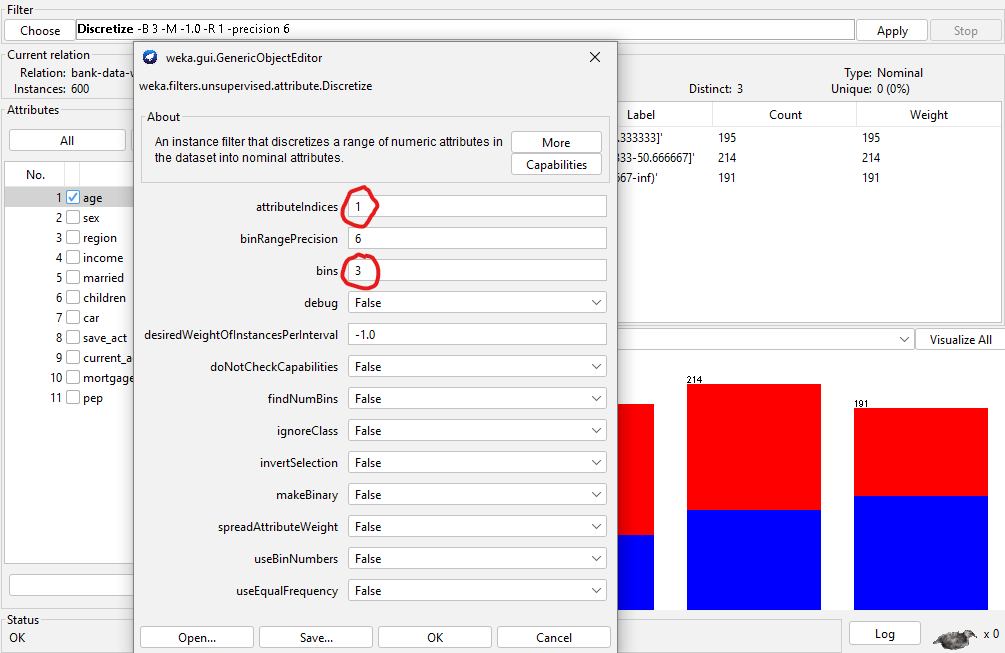
I biramo id kolonu za brisanje.

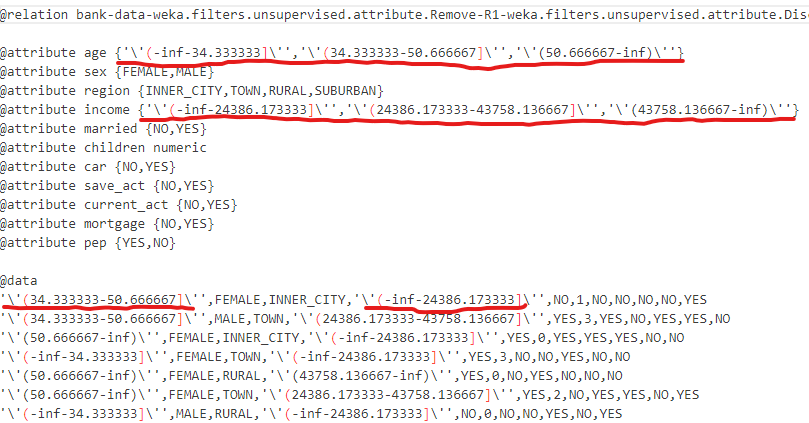


**Diskretizacija -** je proces pretvaranja numeričkih ili kontinuiranih atributa u kategoričke ili diskretne vrednosti - pogodno za klasifikaciju. Za “children” atribut primecujemo da uzima vrednosti: 0, 1, 2, 3.  
Na osnovu toga, sadrzacemo sve podatke, ali cemo promeniti tip **numeric** i zameniti ga diskretnim vrednostima: *@attribute children numeric 🡪 @attribute children {0, 1, 2, 3}*



Vidimo da su vrednosti podeljene “diskretno” u 4 kategorije.  
Dalje, ovo smo mogli “golim okom” da uvidimo, sada cemo se osloniti na weka alate za filtriranje “age” i “income” atributa. Filter, weka-> filters -> unsupervised -> attribute -> Discretize

Promenicemo “defaultne” vrednosti filtera  
**↓**  


Medjutim, desava se jedan problem, a to je u tome kako WEKA dodaje vrednosti atributima:  


Pomocu text editora (VS Code) uredicemo ove vrednosti:

**age :**

'\'(-inf-34.333333]\'' -> 0\_34

'\'(34.333333-50.666667]\' -> 35\_51

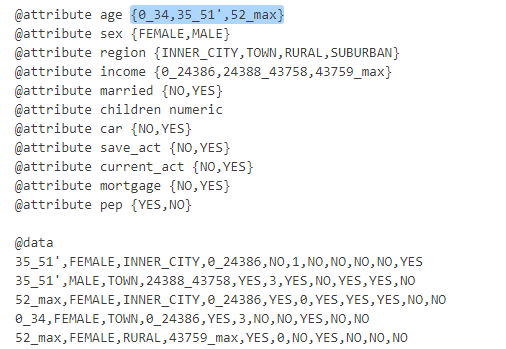
'\'(50.666667-inf)\'' -> 52\_max

**Income :**

'\'(-inf-24386.173333]\'' -> 0\_24386

'\'(24386.173333-43758.136667]\'' -> 24388\_43758

'\'(43758.136667-inf)\'' -> 43759\_max



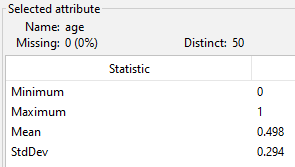
**Koriscenje standardnih metoda za analizu** (sa prez)

**Cišćenje podataka od pogrešnih vrednosti:** Analizom podataka iz weka alata, vidimo da nemamo nevalidne vrednosti (null,?, , …) ili vrednosti koje odskacu od opsega (outlier-a). Svaka instanca je popunjena, nema duplikata… Moglo je da se uvidi zbog postojanja ID-eva I vizualizacije atributa sa WEKA.

**Skaliranje i normalizacija podataka:** kolone koje su prikladne za to su:

* age (godine): Ovo je numerički atribut koji bi se mogao skalirati ili normalizirati kako bi se prilagodio određenom rasponu. Npr, možemo primeniti standardno skaliranje kako bismo dobili vrednosti u rasponu između 0 i 1.
* income (prihod): Ovo je još jedan numerički atribut koji bi se mogao skalirati ili normalizirati kako bi se postigao željeni raspon vrijednosti.
* children (kidare): Ovaj atribut je numerički, ali predstavlja broj djece. Verovatno nije potrebno skalirati ili normalizirati jer već ima diskretne vrijednosti.

Normalizujemo podatke iz razloga da ako neke kolone imaju razlicite vrednosti (age i income) da se ne da posebna vaznost visokom vrednostima, normalizujemo ih tako da imaju jednaku vaznost. Ovo mozemo uraditi jer za age i income nije vezana posebna logika.   
*weka -> filters -> unsuprivesed -> Normalize*   
Sveli smo kontinuirane vrednosti na opseg [0, 1]

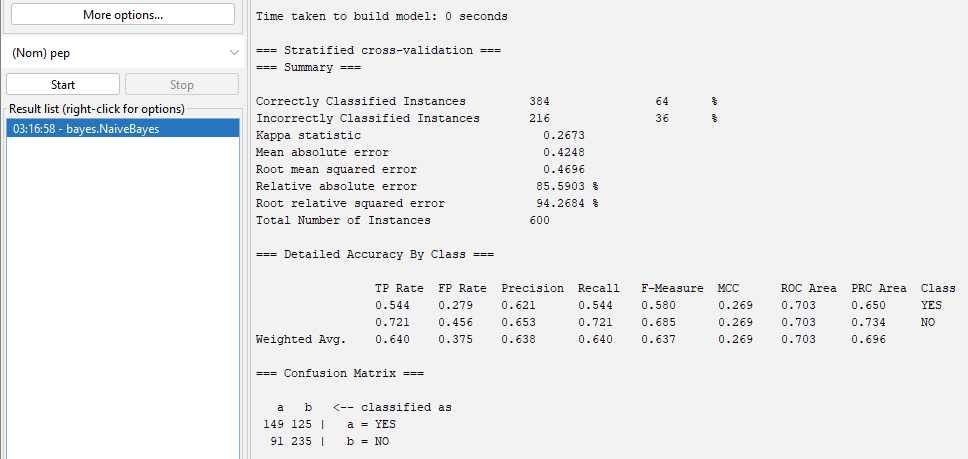


Ostale kategoricke atribute prosto nije neophodno menjati.

Atribut **pep** i treba da ostane nominalan jer u suprotnom Naïve Bayes algoritam ne bi radio.

**3. Primena algoritama** (bez ikakve analize podataka )

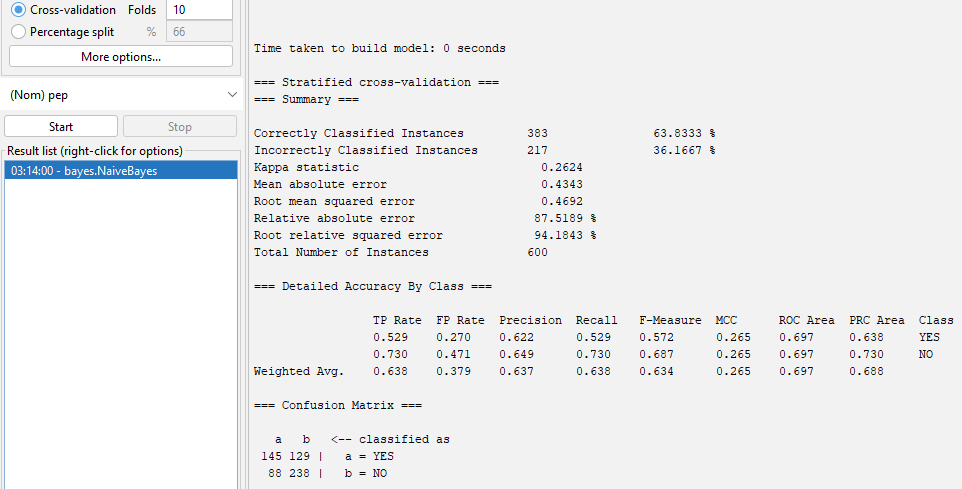
**NaiveBayes**: **64%**



Tacnost algoritma je 64%, ovo ce nam predstavljati dobar “baseline” o tome da li imamo poboljsanja u predstojecim analizama

**3. Primena algoritama** (nad diskretizovanim podacima)

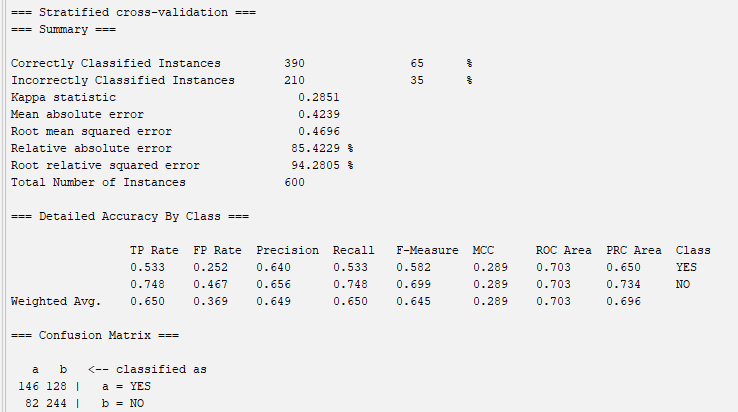
**NaiveBayes**: **63.83%**



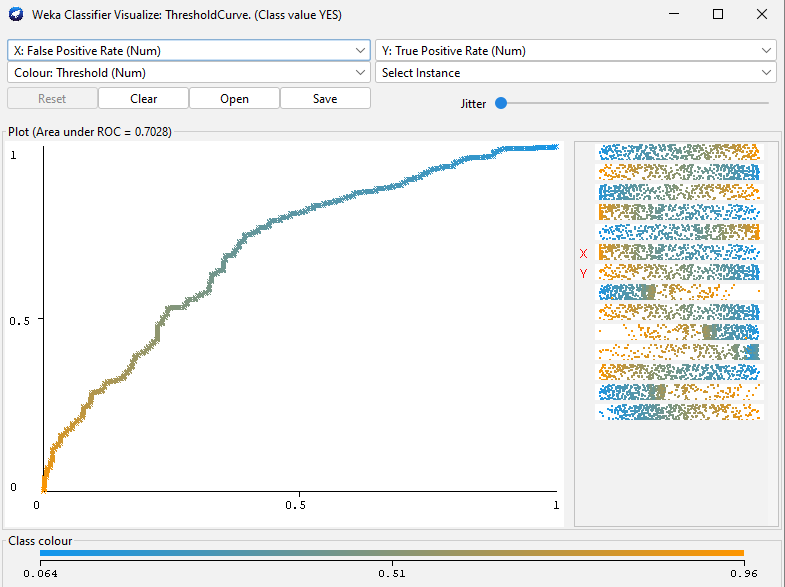
Odmah primecujemo losije rezultate (**63.83%**) u odnosu na nefitriranim dataset-om. (Znak da nije dobra analiza i da treba obustaviti dalji rad)

1. **Primena algoritama** (nad standardno analiziranim podacima)

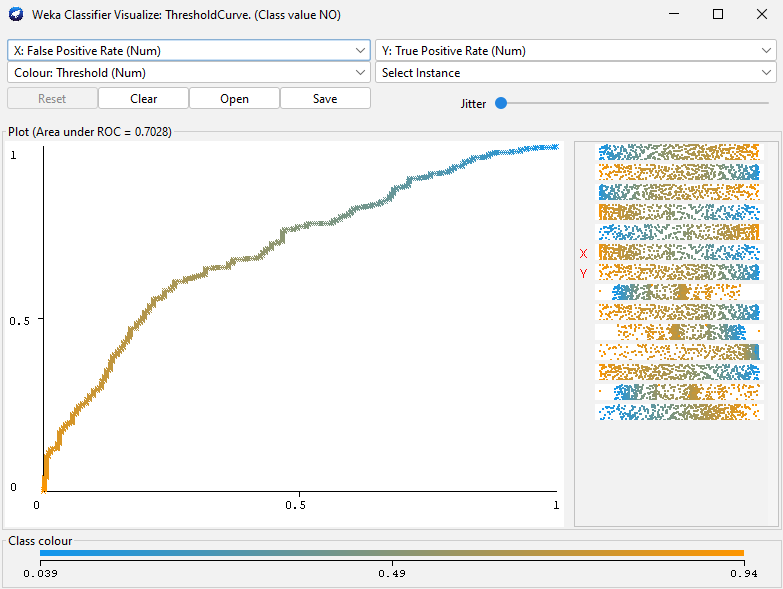
**NaïveBayes** sa cross validacijom**: 65%**



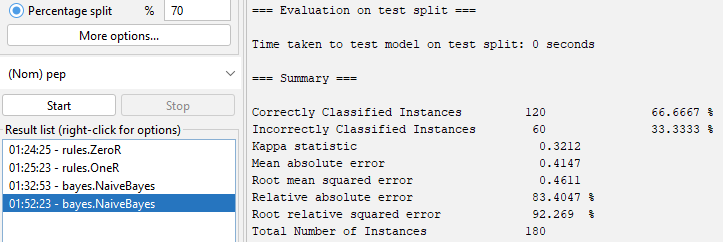
Roc kriva za klasnu vrednost YES:

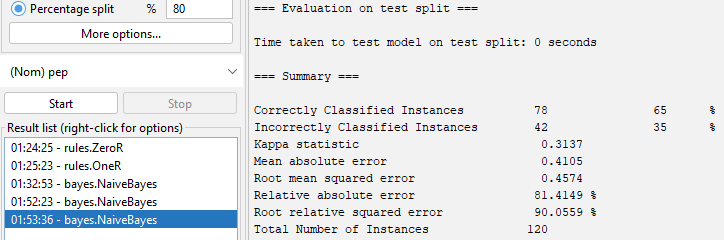


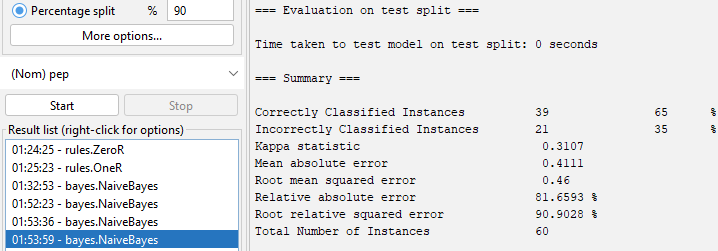
Roc kriva za klasnu vrednost NO:



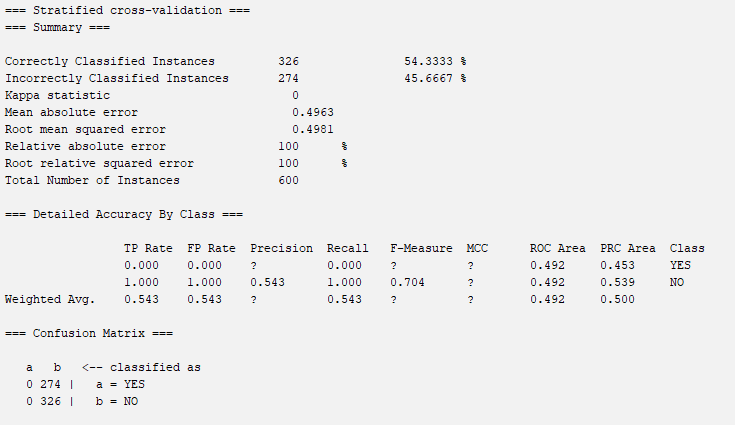
Primecujemo da su ROC vrednosti iste 0.7028. Tj. da se jednako dobro predvidjaju YES i NO vrednosti.

Koriscenjem percentage split opcije 70%, 80%, 90% redom:  


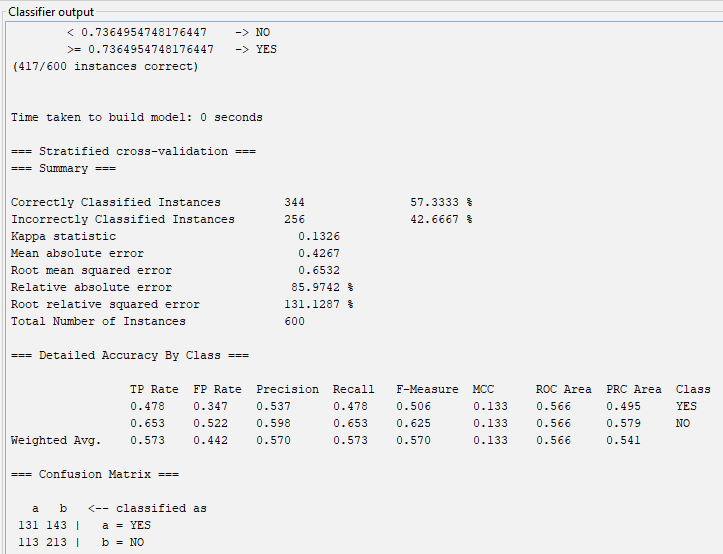




Ono sto primecujemo da cross validation na 70% daje cak bolje rezultate!

**ZeroR**:  


ZeroR je najjednostavniji klasifikator koji uvek bira onu klasu koja je bila u većini prilikom trenirranja podataka.

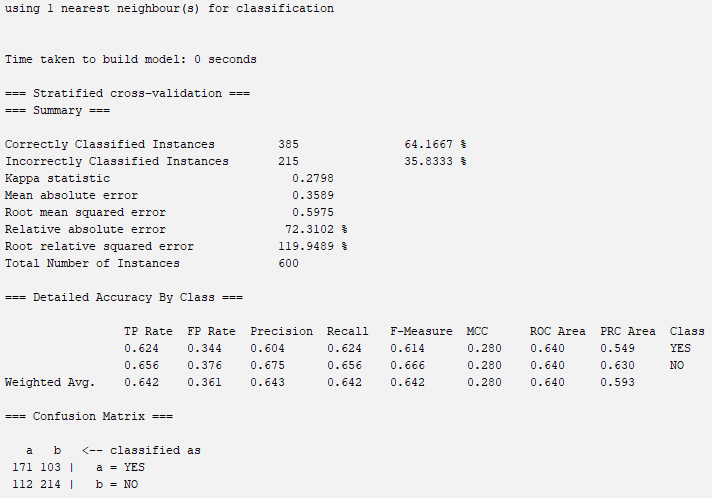
**OneR**: 

OneR je jednostavan klasifikator koji bira jedan atribut i za njega definiše pravila na osnovu kojih se radi klasifikacija. U nasem slučaju je to **income** atribut.

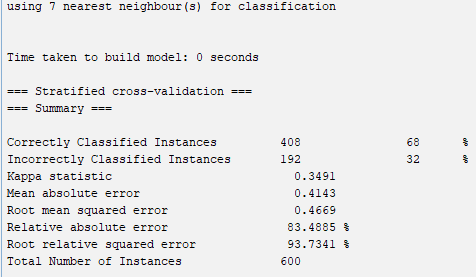
**K Nearest Neighbors (kNN)**

Za ovaj algoritam ne ocekujemo previse dobre rezultate jer nas dataset sadrzi dosta nominalnih vrednosti.

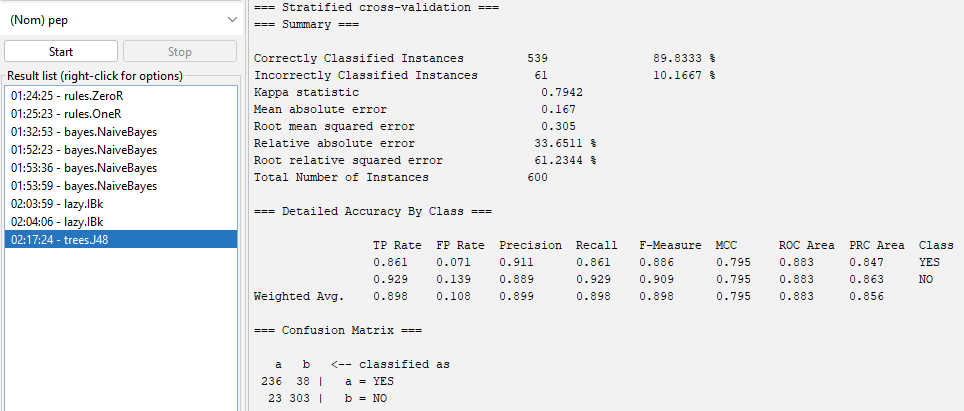
KNN sa vrednoscu 1:



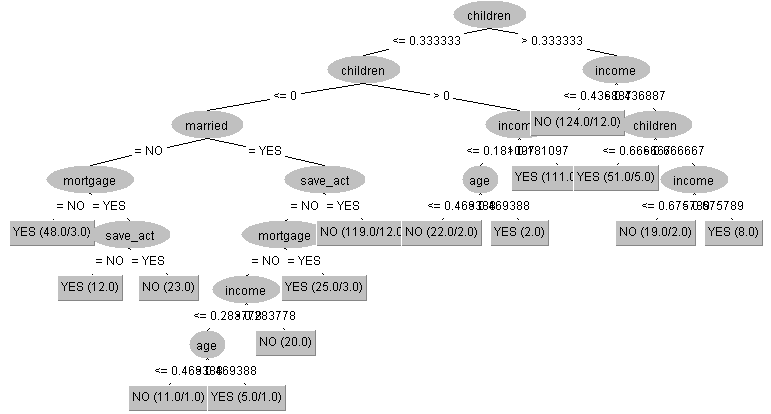
Vidimo da daje vrednosti slicne NaïveBayes algoritmu. Moguce je postici bolji rezultat modifikovanjem broja suseda. Npr. Na 7:

  
Vidimo nesto bolje rezultate, 68% instanci je tacno klasifikovano.

**J48** algoritam:

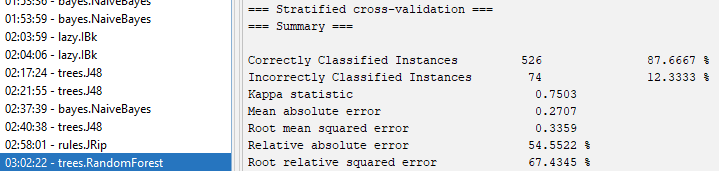


Znacajnije bolji rezultati.

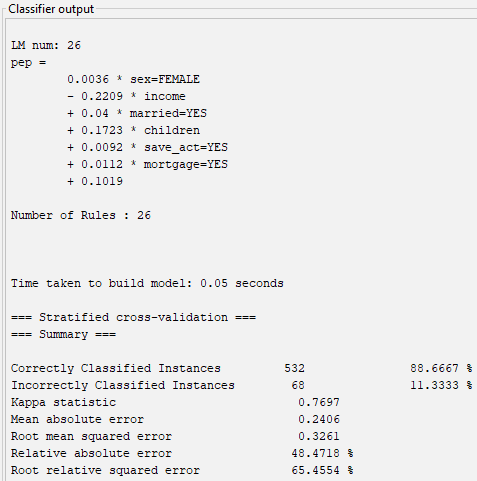
Izgled stabla:   


Ukoliko izabermo da postiji *odsecanje* stabla dobijaju se losiji rezultati (86.33%)

**Random forest**

****

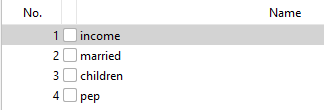
**Classification via Regression:**

****

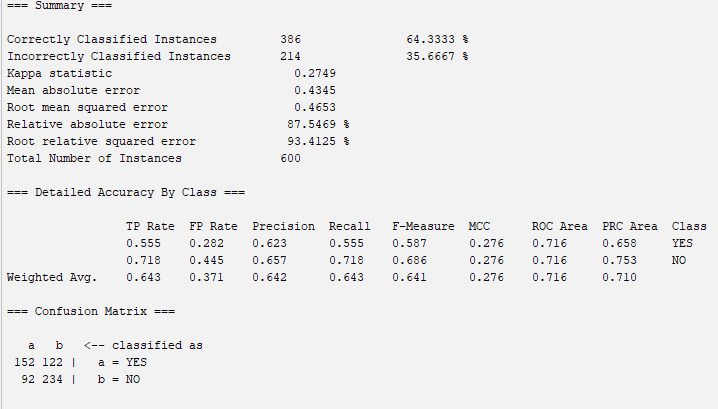
(objektivno bolji algoritmi daju bolje rezultate)

**Selekcija atributa i primena istih algoritama**

Filter -> Choose -> weka -> filters -> supervised -> attribute -> attribute selection. Weka je dosla da zakljucka da su samo ovi atributi bitni za odredjivanje klase:

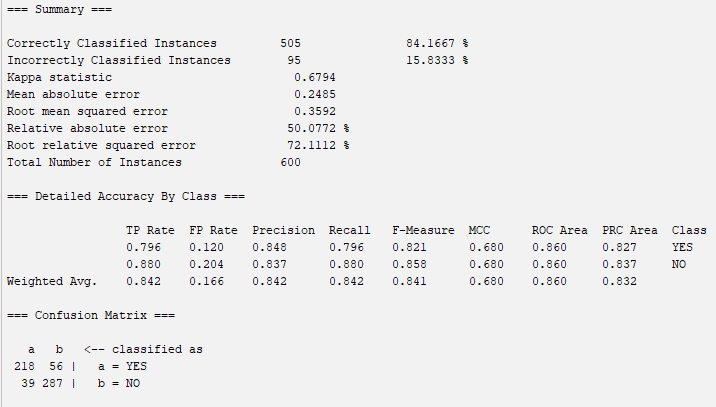


**NaïveBayes** sa cross validacijom:



Primena Bayesovog algoritma nad datasetom sa manje atributa daje nesto lose rezultate (- 0.7%)

**J48** algoritam:



J48 daje znacajnije lose rezultate (- 6%)

Ovo je zapravo sablon, svaki algoritam koji radi nad selektovanim atributima daje losije rezultate.

**Najbolje rezultate daje JRip algoritam**

JRip je algoritam koji kombinuje logiku pravila (rule-based) i stablo odlučivanja. Umesto izgradnje stabla, JRip izvlači pravila iz podataka kako bi se oblikovali klasifikacijski modeli. Ovaj algoritam je dobar izbor kada su podaci kompleksni i sadrže mnogo atributa (kao sto je kod nas slucaj) jer pravila mogu biti lakše razumljiva i interpretirana. 