

1.1)

Gini (Transporation\_mode | Income=Low) =1-(2/2) ^2 – (0/2) ^2 – (0/2) ^2=0

Gini (Transporation\_mode | Income=Medium) =1-(2/6) ^2-(3/6) ^2-(1/6) ^2=1-0,1-0,25-0,028=0,622

Gini (Transporation\_mode | Income=High) =1-0-0-(2/2) ^2=0

Gini|income=2/10 x 0 + 6/10 x 0,622 + 2/10 x 0=0,3732

Κέρδος=0,66-0,3732=0,2868

Gini (Transporation\_mode | Travel\_Cost=expensive) =1 -0 -0 – (3/3) ^2=0

Gini (Transporation\_mode | Travel\_Cost=standard) = 1-0-1-0=0

Gini (Transporation\_mode | Travel\_Cost= cheap) =1-(4/6) ^2-(2/6) ^2-0=1-0.45-0,1=0.45

Gini|Travel\_Cost = 3/10 x 0 + 1/10 x 0 + 6/10 x 0,45=0.27

Κέρδος=0,66-0.27=0,39

Gini (Transporation\_mode | Gender=M) =1-(4/6) ^2-(1/6) ^2-(1/6) ^2=1-0,45-0,028-0,028=0.494

Gini (Transporation\_mode | Gender=F) =1-0-0,5^2-0,5^2=0.5

Gini| Gender = 6/10 x 0.494 + 4/10 x 0.5=0.3+0.2=0.5

Κέρδος=0,66-0.5=0.11

Gini (Transporation\_mode | Car=Y) =1-(2/7) ^2-(2/7) ^2-(3/7) ^2=1-0.08-0.08-0.18=0.66

Gini (Transporation\_mode | Car=N) =1-(2/3) ^2-(1/3) ^2=1-0.45-0.1=0.45

Gini|Car =7/10 x 0.66 + 3/10 x 0.45=0.462+0.135=0.597

Κέρδος=0,66-0.597=0.063

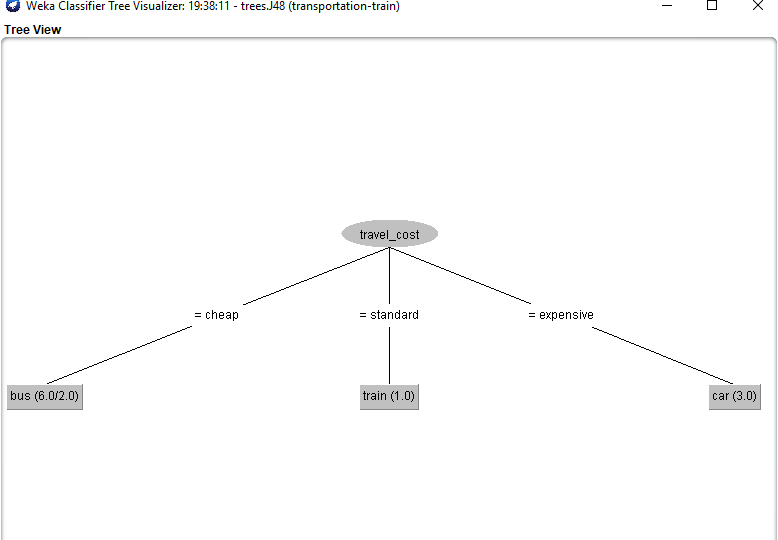
Gini (Transportation\_mode) =1-(3/10) ^2-(3/10) ^2-(4/10) ^2=0,66

Επειδή το μεγαλύτερο κέρδος μας το δίνει το travel\_cost για αυτό το επιλέγουμε για την ριζά του decision tree.

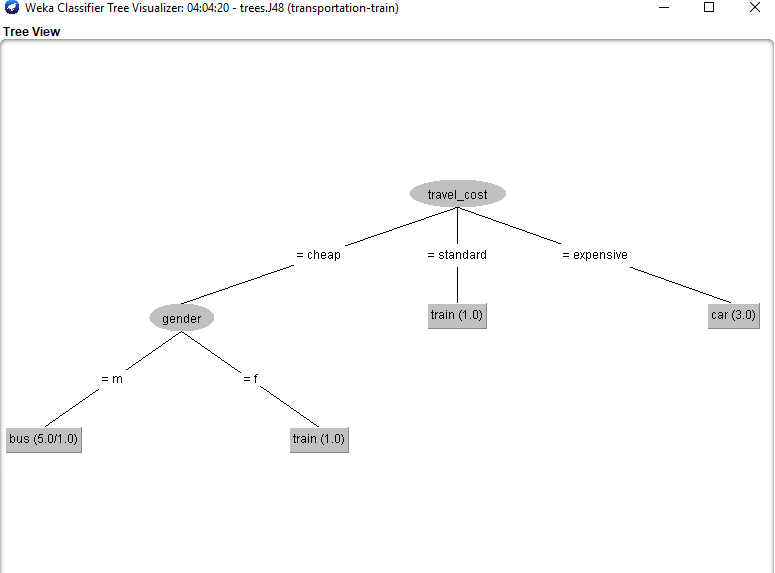
A Decision Tree recursively splits training data into subsets based on the value of a single attribute. Splitting stops when every subset is pure (all elements belong to a single class)

2.1) Το travel cost

2.2)



2.3)

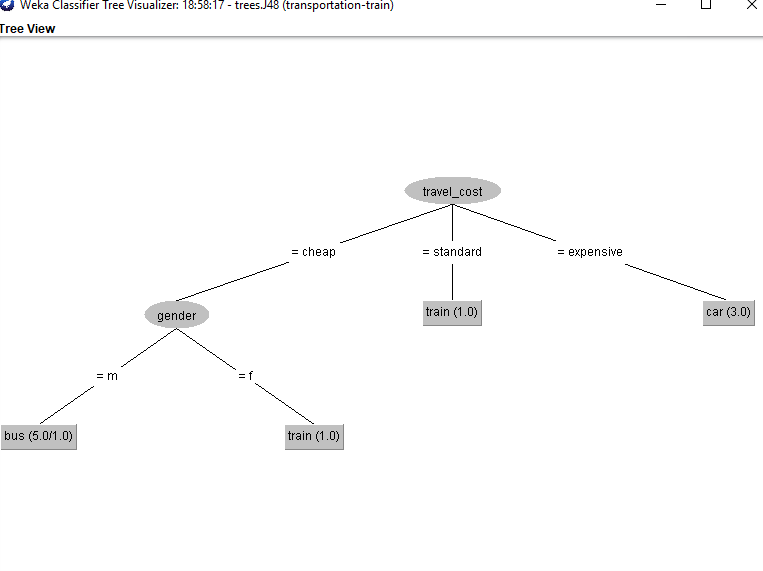
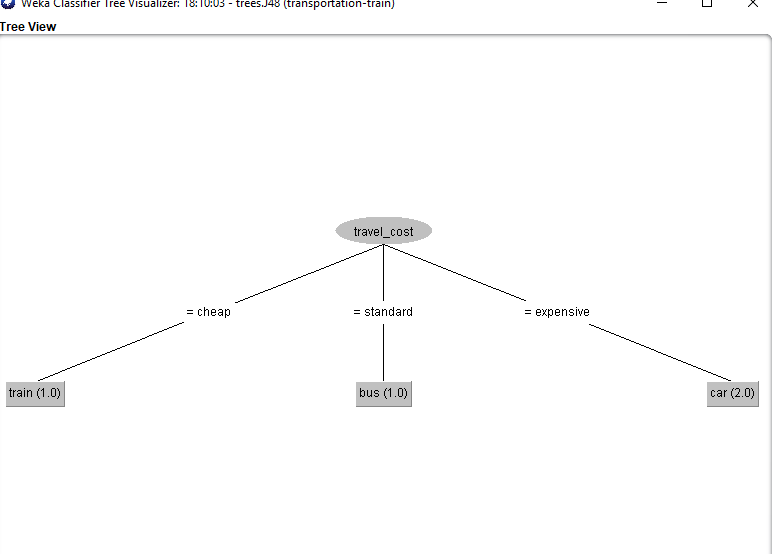


1.2) ουσιαστικά αυτό που φαίνεται και από την ακριβώς πάνω φωτογραφία είναι ότι άμα φτιαχτεί μόνο η ριζά τότε θα έχουμε 1 εγγραφή που θα κατηγοριοποιηθεί λάθος ( το κλαδί cheap)

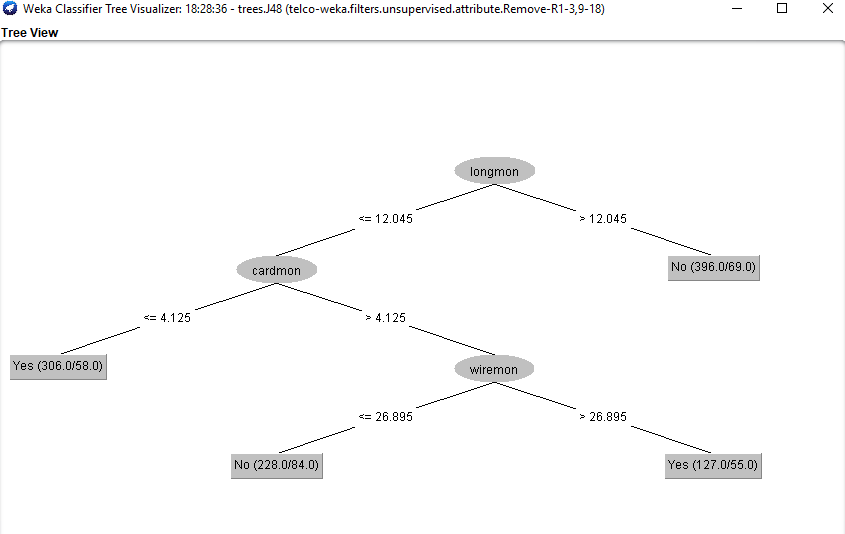
3) και με το χέρι και με το weka έχει 50-50 αποτελέσματα (Correctly Classified Instances 2 50%

Incorrectly Classified Instances 2 50%) με minNumObj=2 και με minNumObj=1 έχουμε 75-25 % αποτέλεσμα (Correctly Classified Instances 3 75 %

Incorrectly Classified Instances 1 25 %)



**ΠΡΟΒΛΗΜΑ 2**

****

Ουσιαστικά το δέντρο μας δείχνει ότι δεν έχουν πρόθεση να φύγουν από την εταιρία όσοι είναι συχνοί χρήστες υπεραστικών κλήσεων, ενώ δεν ισχύει το ίδιο για αυτούς που χρησιμοποιούν κάρτες κλήσης με χαμηλή συχνότητα και αυτοί που έχουν ασύρματο δίκτυο και το χρησιμοποιούν με αραιή συχνότητα έχουν την τάση να παραμείνουν στην εταιρία ( J48 αλγόριθμος ,προεπιλεγμένες τιμές)

2)για κ=1 (Correctly Classified Instances 718 67.9281 %)

Incorrectly Classified Instances 339 32.0719 %

Για κ=5 (Correctly Classified Instances 750 70.9555 %

Incorrectly Classified Instances 307) 29.0445 %

Για κ=15 (Correctly Classified Instances 749 70.8609 %

Incorrectly Classified Instances 308 29.1391 %)

Για κ=25 (Correctly Classified Instances 765 72.3746 %

Incorrectly Classified Instances 292 27.6254 %)

Για κ=35 (Correctly Classified Instances 766 72.4693 %

Incorrectly Classified Instances 291 27.5307 %)

Για κ=45 (Correctly Classified Instances 748 70.7663 %

Incorrectly Classified Instances 309 29.2337 %)

Άρα το καλύτερο είναι το κ=35!

3) Ο J48 είναι πιο αξιόπιστος γιατί μας δίνει την ακριβέστερη συσχέτιση των δεδομένων και είναι ίσως και πιο αποδοτικός από τον ΙΒΚ, ο IBK από την άλλη άμα καταφέρουμε και βρούμε την κατάλληλη κΝΝ τιμή τότε θα μας δώσει το μεγαλύτερο «Correctly Classified Instances» το οποίο και θέλουμε.

4) Σε σχέση με τους δυο αυτούς αλγορίθμους θα ήταν προτιμότερο να χρησιμοποιηθεί ο ΙΒΚ με κ=35 καθώς έχει μεγαλύτερο ποσοστό Correctly Classified Instances άρα και πιο σίγουρο.