**Machine Learning and Data Mining   
Εργασία 01 Titanic**

**Γεώργιος Βώβος**

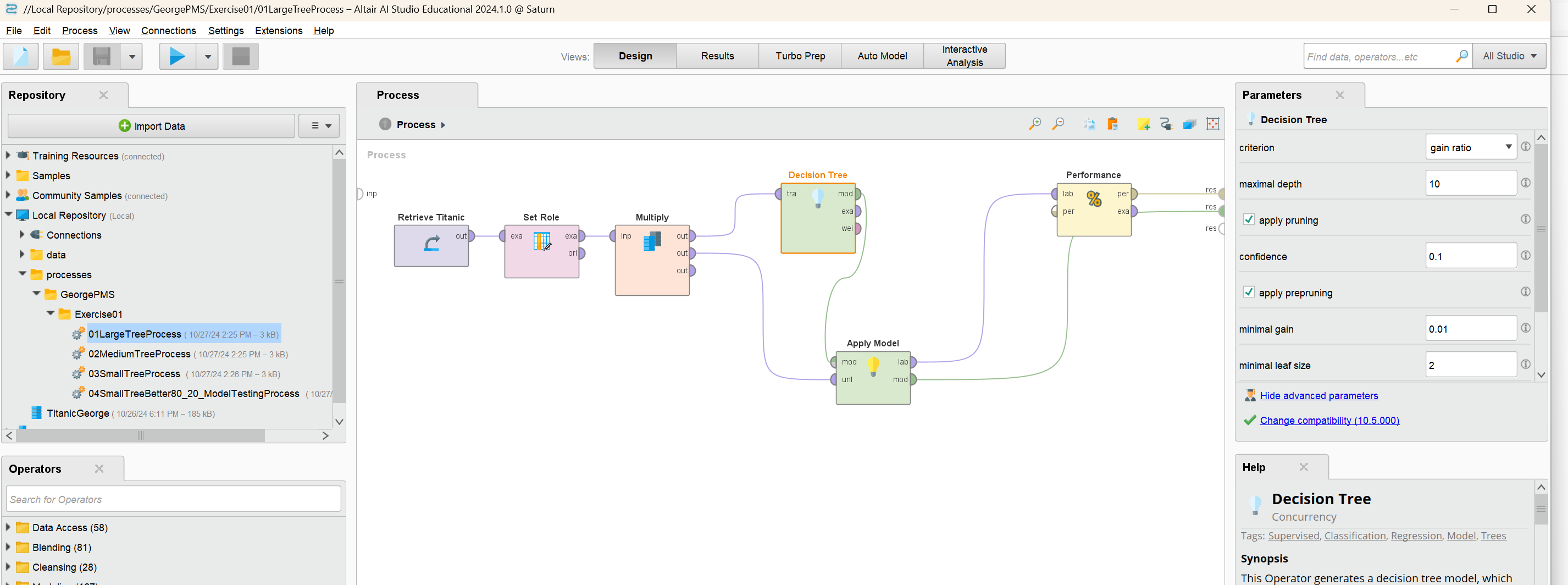
***Να κάνετε δένδρα ταξινόμησης με το Titanic dataset στο AI Studio (RapidMiner) και να σχολιάστε σύντομα τα βήματα.***  
Τα παραδείγματα 1 και 2 έγιναν με βάση τα [tutorial 13,14,15 του Get Started with Altair AI Studio](https://www.youtube.com/watch?v=u2RQcev_SGQ&list=PLssWC2d9JhOZLbQNZ80uOxLypglgWqbJA&index=13)  
 *1)Δοκιμάστε να δείτε την απόδοση με μικρό, μεσαίο και μεγάλο δένδρο. Λογικά όσο μεγαλώνει το δένδρο ανεβαίνει και η απόδοση. Ισχύει? Γιατί συμβαίνει αυτό?*  
 Στα screenshots [1],[2],[3] βλέπουμε τα 3 δένδρα και τα σχετικά confusion matrices.  
 Η μόνη ρύθμιση που αλλάξαμε για ήταν το “minimal gain” σε 0.01/0.05/0.09 για μεγάλο/μεσαίο/μικρό δένδρο   
 αντίστοιχα.  
   
 Όντως, βλέπουμε το accuracy να είναι μεγαλύτερο όσο πιο μεγάλο είναι το δέντρο : 84.42% / 80.44% / 79.14%   
 για για μεγάλο/μεσαίο/μικρό.  
 Αυτό είναι φυσιολογικό (αν και θέλουμε να αποσύφουμε υπερμοντελοποίηση) μιας και τα μεγάλα περιέχουν   
 περισσότερη πληροφορία/κόμβους.  
  
 Στις περισσότερες, όπως και μάλλον στην δική μας, το μικρό και απλό δέντρο έχει σχεδόν το ίδιο accuracy με τα   
 πολύ μεγαλύτερα/πολύπλοκα, ειδικά με το μεσαίο, οπότε θα το προτιμούσαμε.

*2. Χωρίστε το dataset (με split operator) σε training (80%) - testing 0%) και εφαρμόστε το apply model στο testing.*

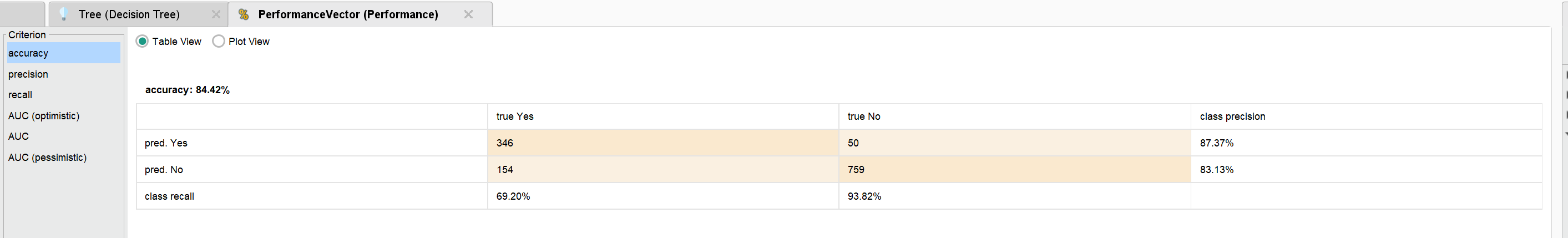
Στα παραπάνω 3 παραδείγματα,”κλέψαμε” λίγο και το accuracy μάλλον δεν θα είναι πραγματικό αν   
 προσπαθήσουμε να το τεστάρουμε με νέα δεδομένα μιας και το testing έγινε πάνω σε δεδομένα που ήταν μέσα   
 στο training dataset.  
 Στα screenshot [4.\*] βλέπουμε μια μικρή παραλαγή του “μικρού” process όπου χρησημοποιήσαμε το 80% των   
 δεδομένων για training και τεστάραμε το μοντέλο με το υπόλοιπο 20%.  
  
 Όπως θα περιμέναμε, αν κρατήσουμε το minimal gain σταθερό στο 0.09 το δένδρο δεν θα ήταν τότε μικρό όσο   
 στο προηγούμενο «μικρό» παράδειγμα [4.1] , για να φτάσουμε σε ίδιο περίπου μέγεθος, αυξήσαμε το minimal   
 gain σε 0.12 [4.2]  
   
 *3. Επειδή το dataset είναι μικρό, μια πιο καλή λύση είναι να αξιολογήσετε με 10-fold-cross-validation*

Στο screenshot [5] φαίνεται το process για το Cross Validation, τα operators είναι τα ίδια με τα προηγούμενα   
 παραδείγματα απλώς βάζουμε τα Decision Tree/ Apply Model /Performance μέσα στο Cross Validation όπως   
 δείχνει το tutorial [Validating a Model | RapidMiner](https://www.youtube.com/watch?v=Hw3tVDZ4Pmo&list=PLssWC2d9JhOZLbQNZ80uOxLypglgWqbJA&index=16)

* Σε όλα τα processes κρατήσαμε το αρχικό dataset, θα μπορούσαμε να το «πειράξουμε» λίγο ,π.χ να αγνοήσουμε στήλες που θα μπορούσαν να επηρεάσουν το αποτέλεσμα όπως το “Life Boat” στην περίπτωσή μας.  
  Τα αποτελέσματα θα ήταν λίγο καλύτερα αλλά η λογική του υπόλοιπου process δεν θα άλλαζε.
* Τα Confusion matrices είναι «ανάποδα» από τα slides του μαθήματος, αριστερά τα predictions,δεξιά τα actuals. Δεν μας επηρεάζει βέβαια μιας δεν κάναμε manual υπολογισμούς για metrics, απλώς δείχνουμε τα υπολογισμένα screenshots.

[1] Μεγάλο δένδρο, minimal gain =0.01 , Accuracy 84.42%  
  
A screenshot of a computer

Description automatically generated



[2] Μεσαίο δένδρο, minimal gain =0.05 , Accuracy 80.44%  
A screenshot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

[3] Μικρό δένδρο, minimal gain =0.09 , Accuracy 79.14%  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

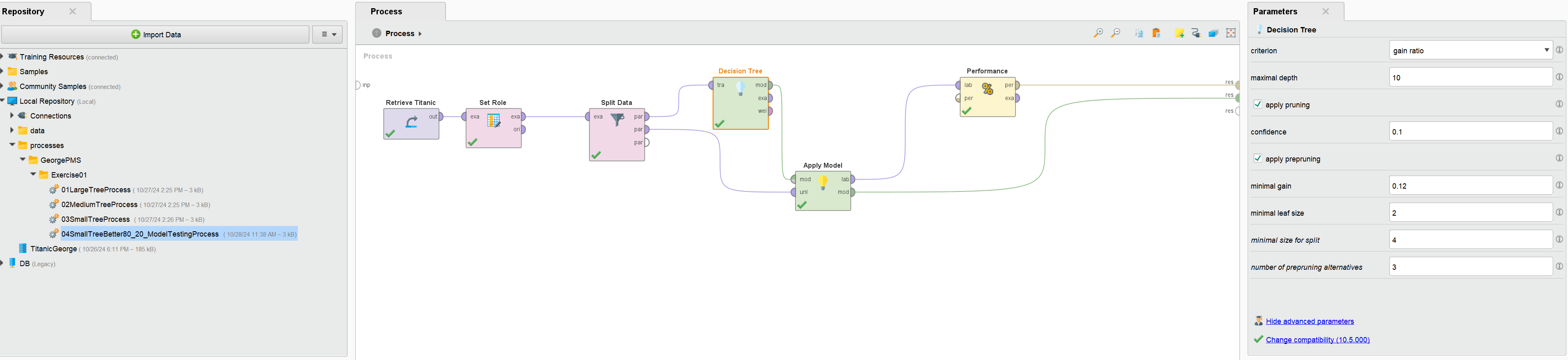
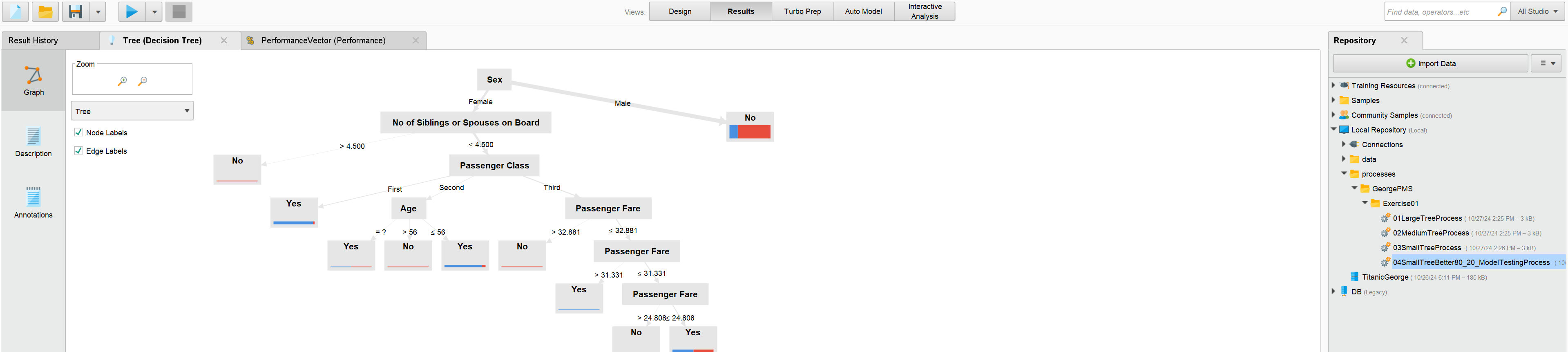
Description automatically generated  
A screenshot of a computer

Description automatically generated

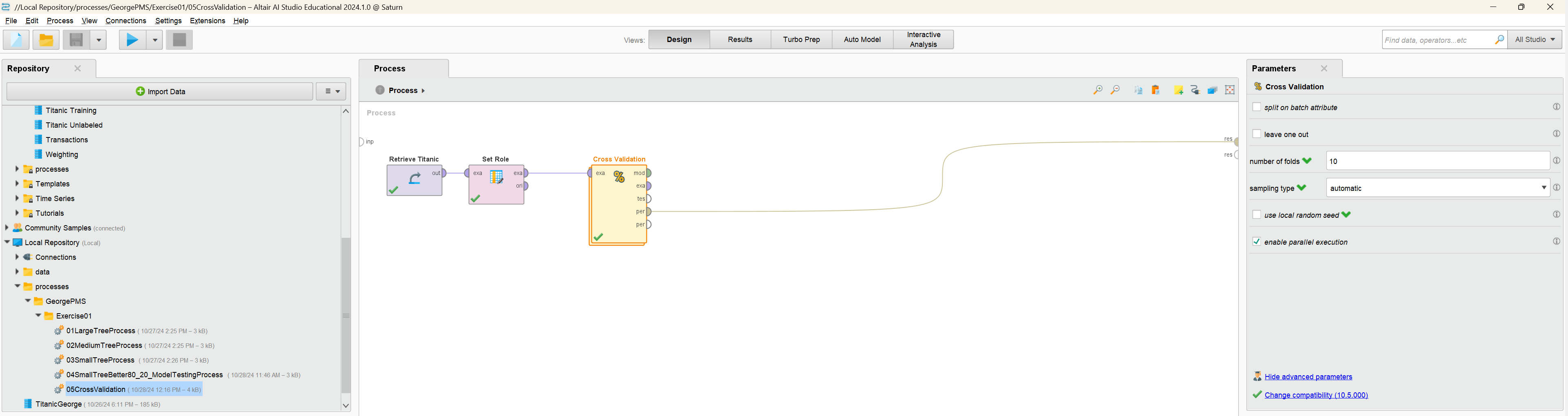
[4.1] «Μικρό δένδρο», minimal gain =0.09, , Accuracy 85.5%  
A screenshot of a computer

Description automatically generated  
A white background with many squares

Description automatically generated

[4.2] «Μικρό δένδρο», minimal gain =0.12, Accuracy 85.5%  
  
  
A screenshot of a computer

Description automatically generated

[5]10 fold Cross validation (minimal gain στο δένδρο μέσα στο cross validation operator=0.01), Accuracy 85.5%  
  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated