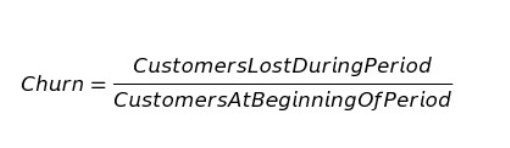
Churn Modeling

Ian Liu & Ian Shyue

# Introduction

## What is churn modeling?

* Churn: Customer did not complete his/her contract
* Churn modeling predicts what customers are likely to churn



Churn is defined as the ratio of customers lost during a period of time to the customers at the beginning. For example, if we had 1000 customers at the beginning of February but only 900 customers at the end of February, our churn rate is 10%.

## Why do we need churn modeling?

Subscription-based services, banking services, and many businesses who value long-term customer relationships could benefit from churn modeling. It tells us quantitatively how likely a customer will be lost and sometimes it also tells us how we might fix it. We can then focus our efforts on churn prevention with a targeted group of customers. Lastly, churn modeling allows us to determine how much resources spent on preventing churn is worthwhile. Since we have an anticipation of how many people might churn, we can plan our resources ahead of time for churn prevention.

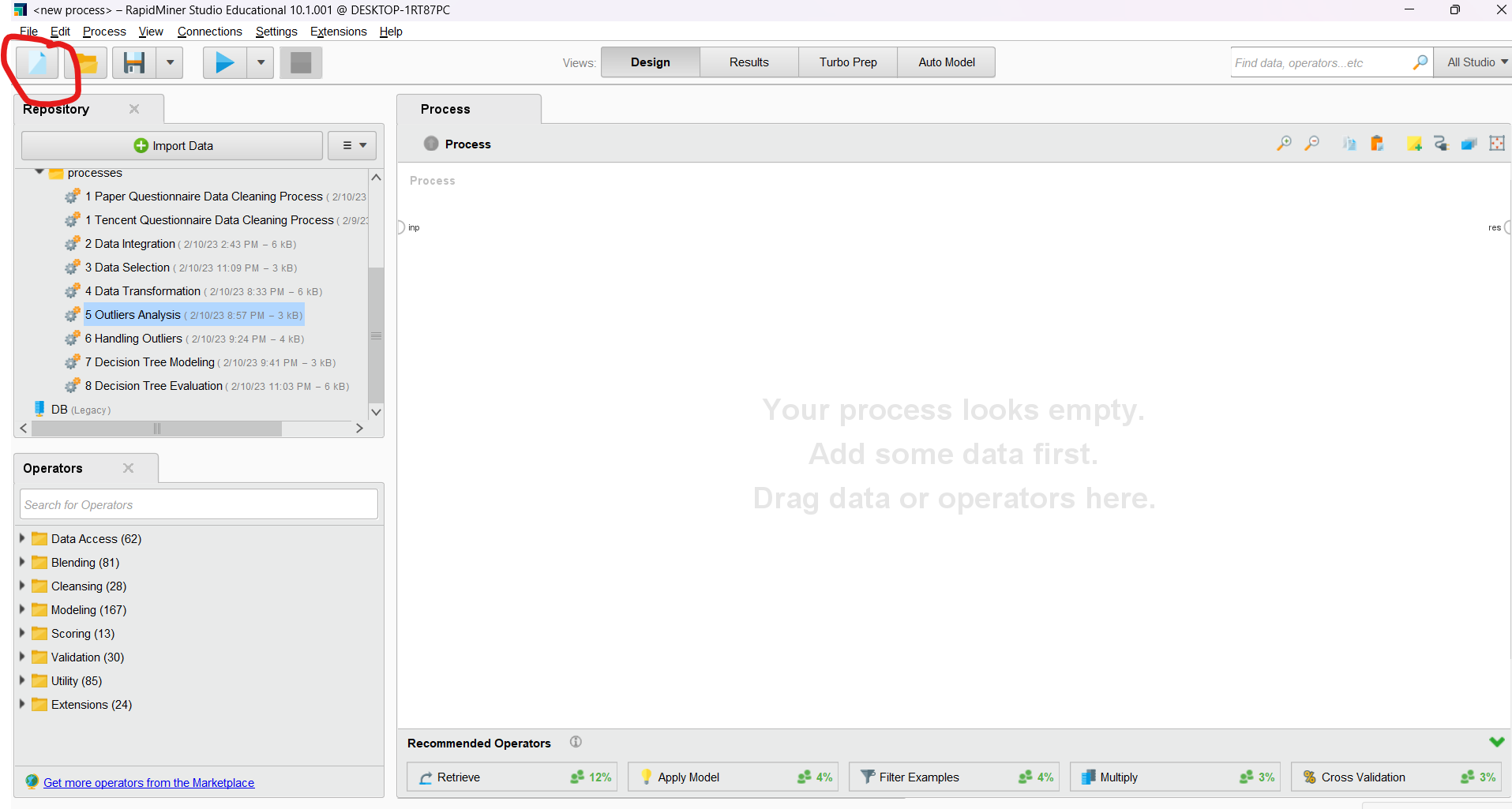
## Types of Churn

There are two types of churn. Involuntary and voluntary. Examples of voluntary churn include: switching from Netflix to Disney Plus because you like Disney movies more or using Apple Music instead of Spotify because you want lossless audio compression. Examples of involuntary churn include: expired credit card or the death of customers.

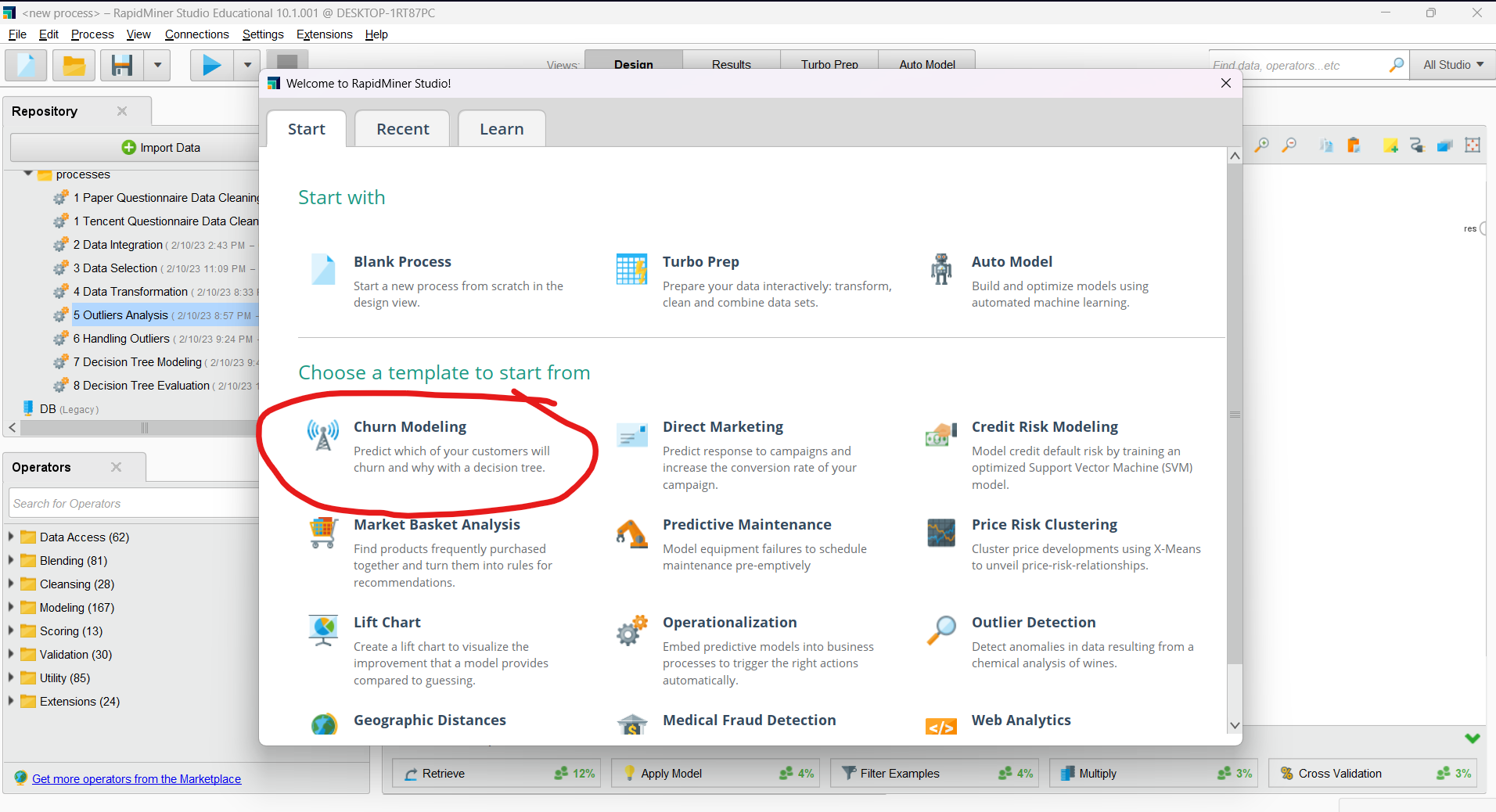
# Churn Modeling Tutorial

Retrieve Data

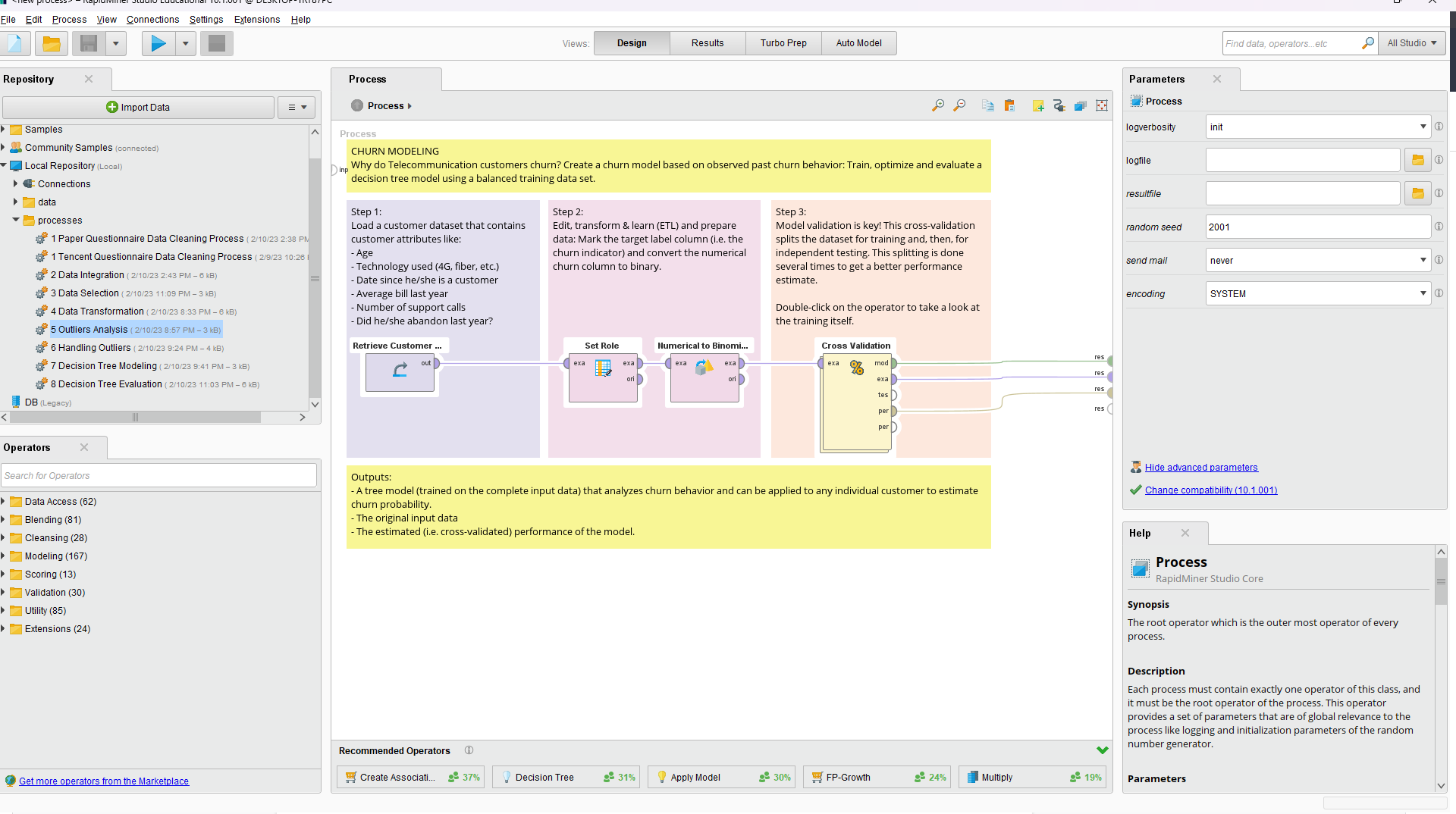
* Create a new process by clicking in the top left hand corner of the Rapid Miner interface.



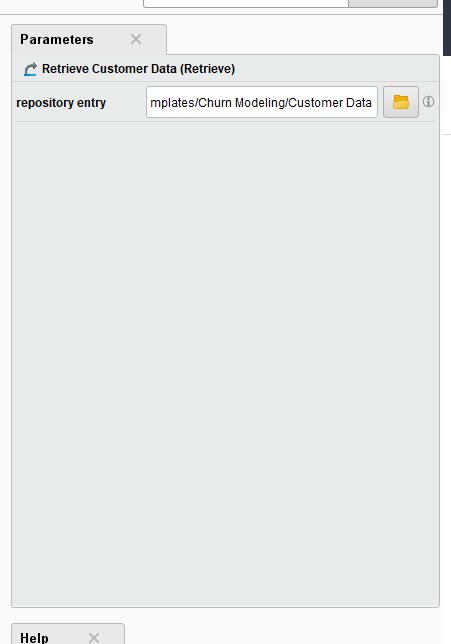
* Click on Churn Modeling



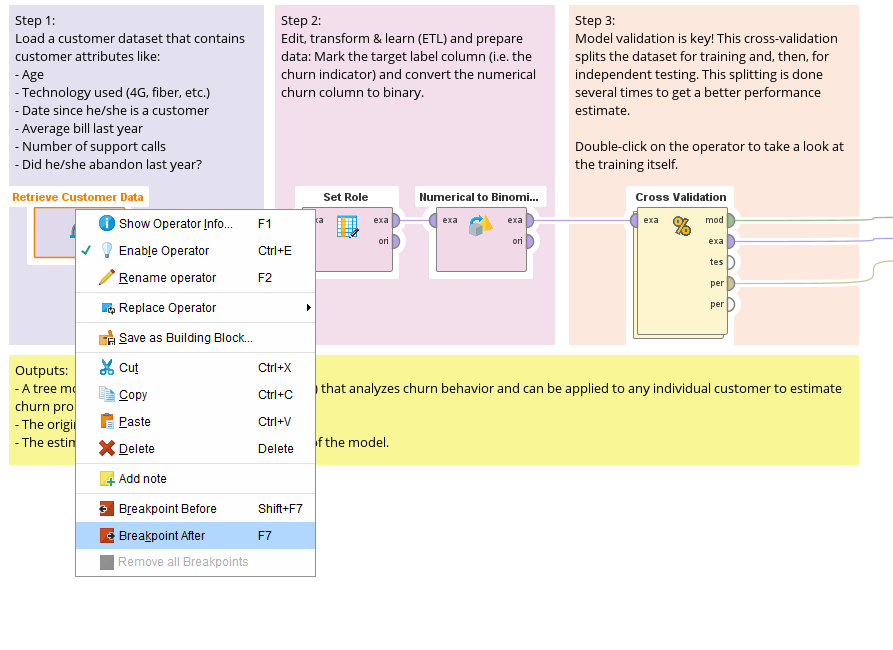
* An interface should show up like below

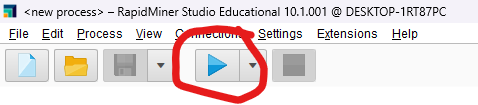


* Click on the Retrieve Customer Data component. Click on the folder icon in the Parameters window if you want to load your own data.

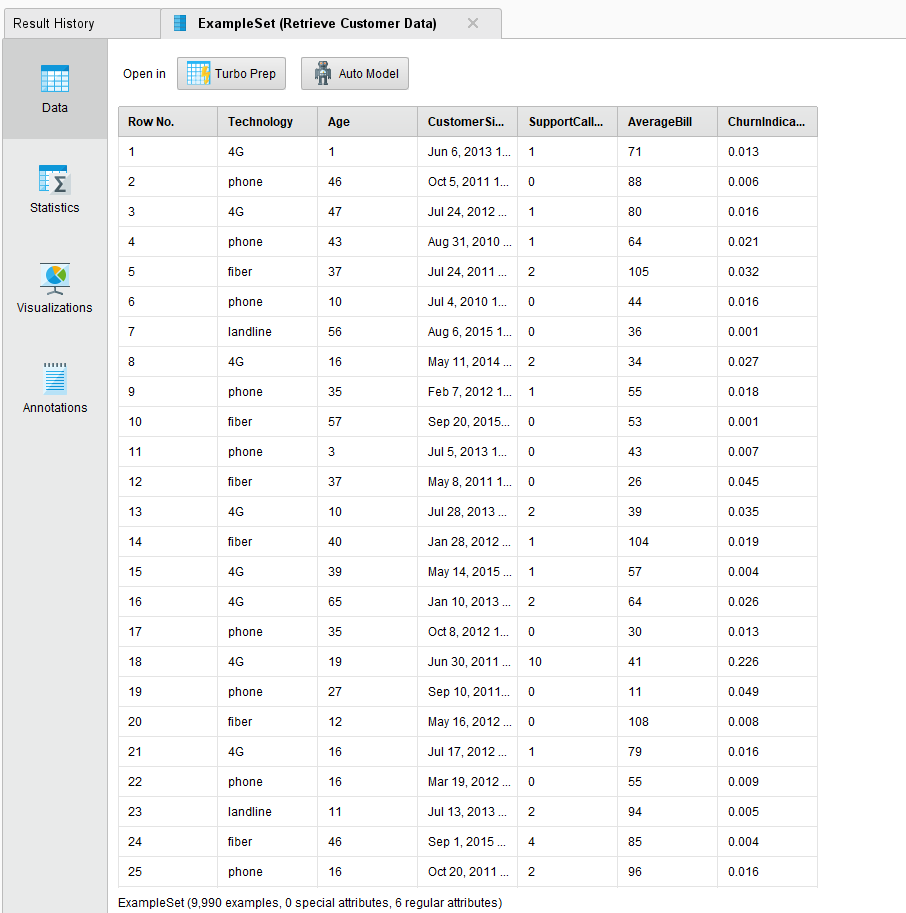


* To view what the data looks like, right click on the operator and click Breakpoint After. Then run the process by clicking the Play button in the upper left hand corner.



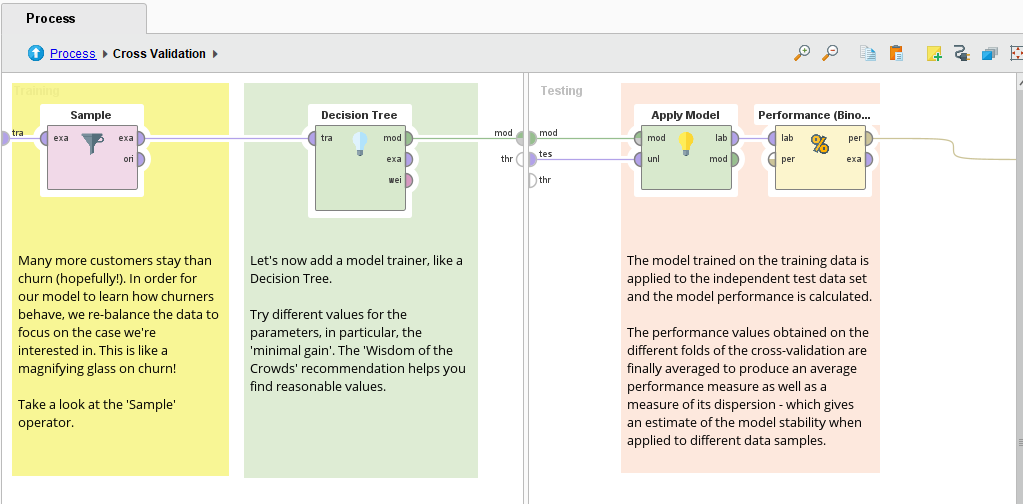


* You should see the data after running the process.

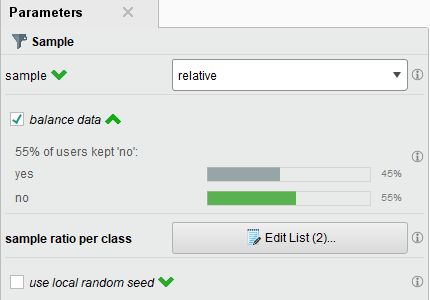


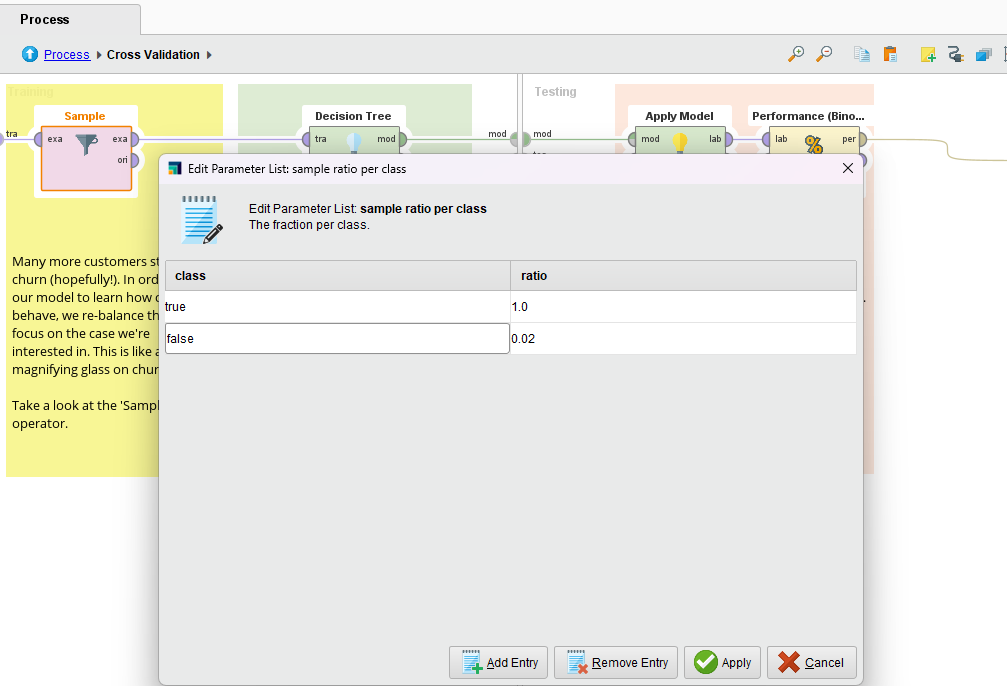
* Let’s understand the data. There are 6 regular attributes (attributes that have a specific role) and 9990 examples. You can click on the Statistics or Visualization panel on the left to better understand the data.
* In the next Set Role operator, we set ChurnIndicator to have the role of label. This means that our target variable (also known as our dependent variable) is ChurnIndicator as this is the variable we are trying to predict.
* In the next operator (Numerical to Binomial), we convert the numerical column (ChurnIndicator) to binary.
* **Cross Validation**. This operator splits the dataset into training and testing so we can better measure the performance of the model. If we look at the parameters of this operator we see that we have a split on batch attribute, which allows us to partition the dataset using a specific attribute. The leave one out option is an extreme version of k-fold cross-validation that has the most expensive computation. The model is evaluated for every example in the dataset. Since our dataset is not too large, some may consider using this. The “enable parallel execution” option speeds up the process by using multiprocessing (the idea is to use more than one core of the CPU to do computations).

Upon further discovery, we realize that this operator has multiple processes enclosed, so let’s double click on the operator and see what’s inside.

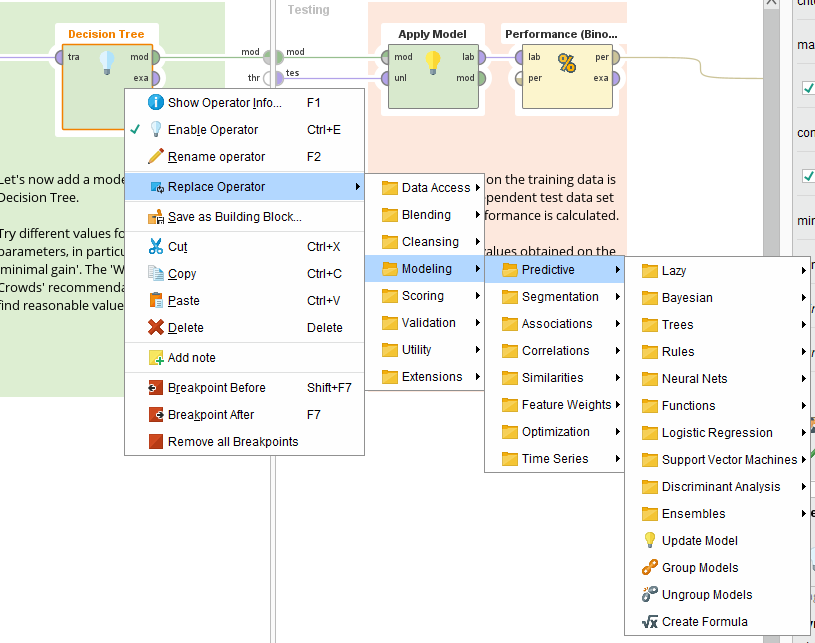


* **Sample**. Let’s go into each operator of the Cross Validation process in closer detail. The balance data selection is used for sampling data that may be imbalanced (ex. All are true and only a few are false). The sample size per class lets us specify a ratio or amount for each class so we don’t end up with an imbalanced dataset (stratified sampling). Lastly, he “use local random seed” option lets us get consistent results from our model by stabilizing the randomness.

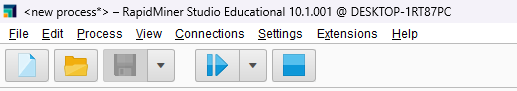




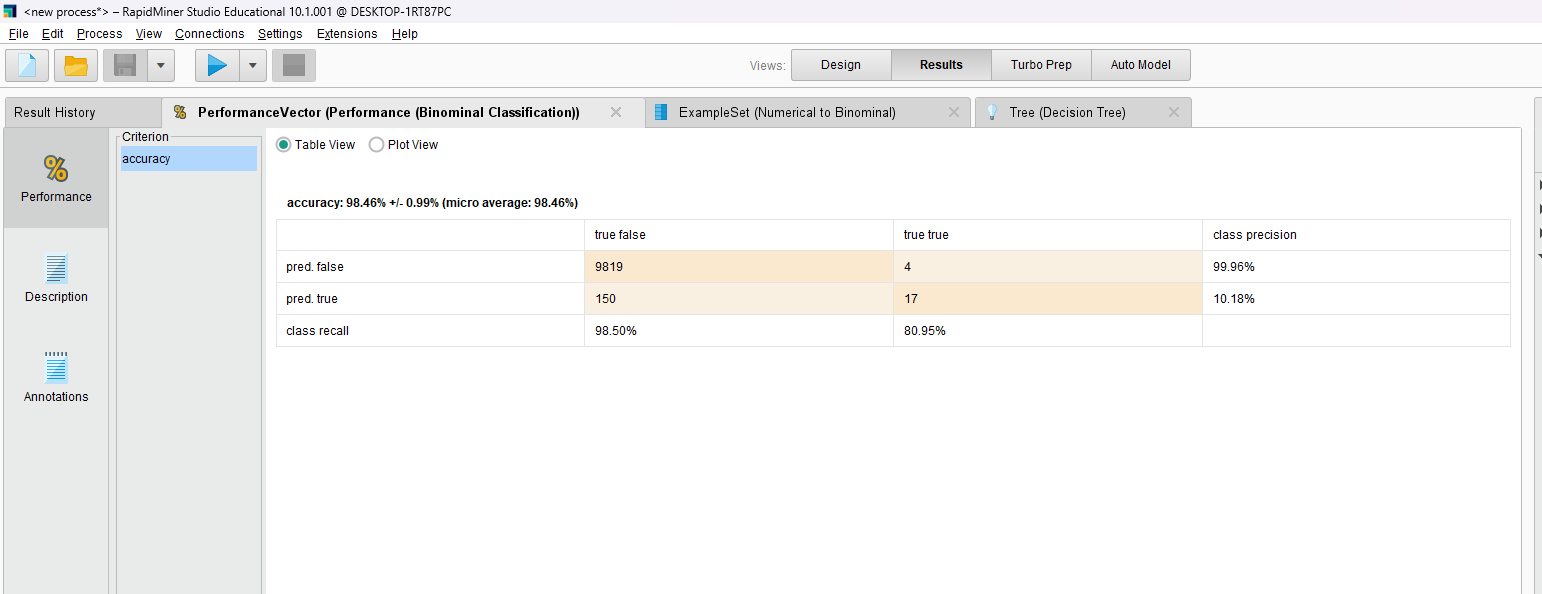
* **Decision Tree**. This is just one of the multitude of models RapidMiner offers. The Decision Tree we use is optimized for minimal gain. To replace this operator with another model, simply right click => Replace Operator > Modeling > Predictive and choose the desired model.



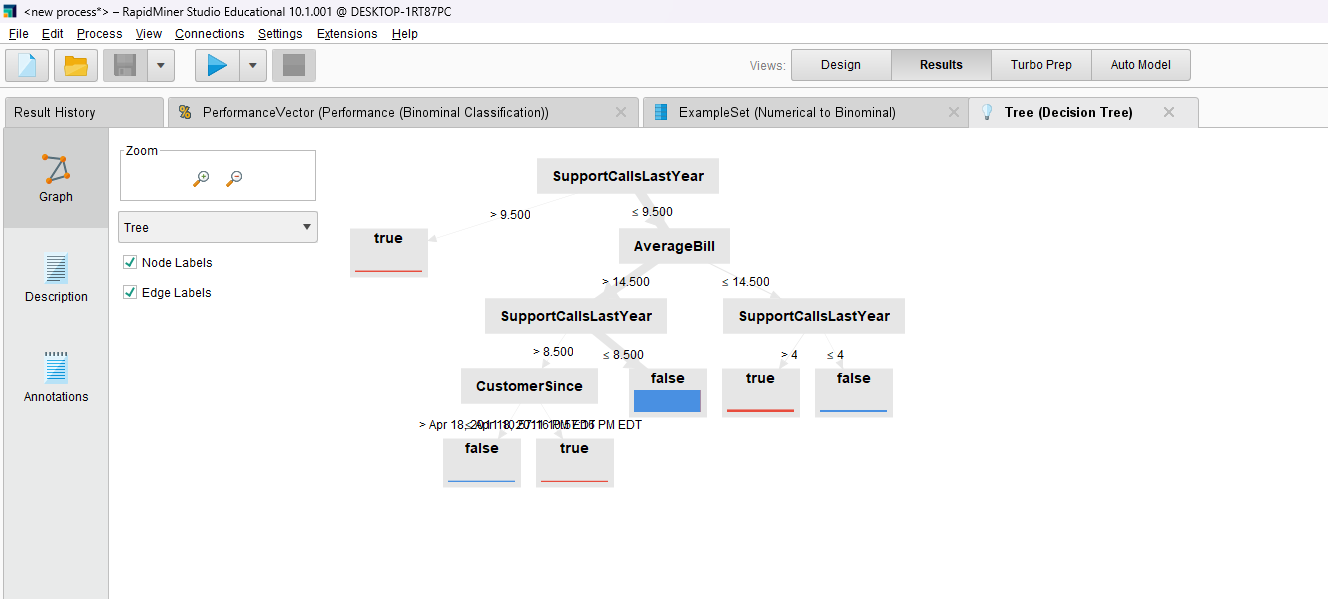
* **Apply Model**. The Apply Model operator receives the model and the testing dataset to evaluate the model.
* **Performance (Binomial Classification)**. This measures the performance of the model based on the criteria specified. Here we use accuracy as the criteria, but we can also use precision and recall if we wanted to.
* Let’s run the process from start to finish. If you ran the process with the breakpoint earlier, the run button will look like this. Click it again to finish running.



* As you can see below, our results tab presents us with a confusion matrix. It seems like we have better accuracy at predicting when customers will not churn, which makes sense because most of the times our customers did not churn.



* We can also click on the Tree (Decision Tree) tab to look at which features were the most important features for determining customer churn.



# Summary

We walked through a simple example of how to use a decision tree to conduct churn modeling. In reality, the data is much larger, far more messy, and the algorithmic choices much more complex. For example, the idea of seasonality patterns of customer churn or correlation with other features not present in the dataset could all present more difficulties in our model building process. However, this is nonetheless an educational lens for which beginners can use to step in to the world of churn modeling– and the fun that comes with it.