An Introduction to Machine Learning: Hands-On Tutorial

In this tutorial, you will build two relatively simple Machine Learning models and test them, all without ever leaving Galaxy. One Linear Regression (LR) and one Random Forest (RF) model will be created, trained and evaluated on an RNAseq dataset (already mapped to genes). The target value of our prediction is the patient's age.

The dataset was curated by Fleischer et al for their 2018 research article titled "Predicting age from the transcriptome of human dermal fibroblasts". It contains around 27.000 genes and their expression levels for 133 patients, as well as the patient's age. The authors trained an ensemble regressor model to predict the age of a patient from the gene expression data. Their model reached an R^2-Score of 0.81.

All of this tutorial is carried out on the Machine Learning instance of Galaxy (ml.usegalaxy.eu) Whenever you should select an option or click a button, the text will be in *italics*. If at one point you are presented with more options for a tool than those which are mentioned in this tutorial, let them empty or at their default value.

So, let's start by loading the data into Galaxy.

Loading the data

1. Click on *Upload Data*
2. Select *Paste/Fetch data*
3. Enter the following URL

   > https://zenodo.org/record/2545213/files/training_data_normal.tsv

4. Click *Start*
5. Close the window
6. Rename the data set to "training_data"
   1. Click on the *Edit attributes* icon (small pencil) next to the data set in your history
   2. Enter "training_data" in the *Name* field
   3. Click *Save*

For machine learning, we need a training set to, well train our models, and a test set on which we will later evaluate the models. As our data set is just a singular table, we have to do the split ourselves.

Creating training and test sets

1. In the *Tools* section, choose *Machine Learning → Split dataset*
2. The parameters should be set as follows:
    - Select the dataset containing array to split: *training_data*
    - Does the dataset contain header?: *Yes*
    - Select the splitting mode: *Train Test Split*
        - Test size: *0.1*
        - Random seed number: -
        - Shuffle strategy: *Shuffle*
3. Click *Execute*
4. Rename the data sets to "train_set" and "test_set"

At this point, we have to cut the target value from the test_set, because otherwise, Galaxy will not be able to predict on the test set later on.

1. In the tools section, choose *Text Manipulation → Cut columns from a table (cut)*
2. Choose the following options:
    - File to cut: *test_set*
    - Operation: *Discard*
    - Delimited by: *Tab*
    - Cut by: *fields*
        - List of fields: *Column: 27143*
3. Click *Execute*
4. Rename the new file to "test_set_no_label"

Now that we have our data sets, we can begin to build a model and train it. We will start with the Linear Regressor.

Build a Linear Regression model and train it

1. In the Tools section, choose *Machine Learning → Generalized linear models*
2. Set the parameters to the following:
    - Select a Classification Task: *Train a model*
        - Select an linear model: *Linear Regression model*
            - Select input type: *tabular data*
            - Training samples data set: *train_set*
            - Does the dataset contain header: y*es*
            - Choose how to select data by column: *All columns EXCLUDING some by column header name(s)*
                - Type header name(s): *age*
            - Dataset containing class labels or target values: *train_set*
            - Does the dataset contain header?: *yes*
            - Choose how to select data by column: *Select columns by column header name(s)*
                - Type header name(s): *age*
3. Click *Execute*
4. Rename the model to "LinearRegressor"

We trained the Linear Regression model on the training set, now off to the evaluation. First, we will predict data in the test set, so that we can later compare these predictions the true labels.

Test the Linear Regression model

1. In the Tools section, choose *Machine Learning → Generalized linear models*
2. Set the following parameters:
    - Select a Classification Task: *Load a model and predict*

- Models: *LinearRegressor*
- Data: *test_set_no_label*
- Does the dataset contain header?: *Yes*
- Select the type of prediction: *Predict class labels*

3. *Execute*
4. Rename the produced file to "LinearRegressor_pred"

And now, the same procedure for the Random Forest model.

Build a Random Forest model and train it

1. In the Tools section, choose *Machine Learning → Ensemble Methods*
2. Set the parameters to the following:
    - Select a Classification Task: *Train a model*
        - Select an ensemble method: *Random forest regressor*
            - Select input type: *tabular data*
            - Training samples data set: *train_set*
            - Does the dataset contain header: y*es*
            - Choose how to select data by column: *All columns EXCLUDING some by column header name(s)*
                - Type header name(s): *age*
            - Dataset containing class labels or target values: *train_set*
            - Does the dataset contain header?: *yes*
            - Choose how to select data by column: *Select columns by column header name(s)*
                - Type header name(s): *age*
3. Click *Execute*
4. Rename the model to "RandomForestRegressor"

Test the Random Forest model

1. In the Tools section, choose *Machine Learning → Ensemble Methods*
2. Set the following parameters:
    - Select a Classification Task: *Load a model and predict*
        - Models: *RandomForestRegressor*
        - Data: *test_set_no_label*
        - Does the dataset contain header?: *Yes*
        - Select the type of prediction: *Predict class labels*
3. *Execute*
4. Rename the produced file to "RF_Regressor_pred"

For evaluation, we will use the R^2-score, a commonly used metric for regression, which was also used in the original paper of Fleischer et al.

Evaluate and compare the two models

The following steps have to be done twice, once for each model.

1. Under the Tools section, choose *Machine Learning → Calculate metrics for regression performance*
2. Set the parameter as follows:
    - Metrics: *r2_score*
        - Dataset containing the true labels (tabular): *test_set*
        - Does the dataset contain header?: *yes*
        - Choose how to select data by columns: *Select columns by column header name(s)* (RF)
            - Type header name(s): *age*
        - Dataset containing predicted values (tabular): *LinearRegressor_pred (*LR) / *RF_Regressor_pred* (RF)
        - Does the dataset contain header?: *no* (LR) / *yes (RF)*
        - Choose how to select data by columns: *Select columns by column index number(s)* (LR) / *Select columns by column header name(s)* (RF)

    - Select target column(s) (LR): *c27143*
    - Type header name(s) (RF): *predicted*
3. Click *Execute*
4. Rename the scores to "r2_LR" and "r2_RF"

Let's have a look at the R^2-scores of our models. They are lower than this in the original paper (0.77) and the Linear Regressor performs better than the Random Forest. But they are not that far off. Pretty good for the first try. With a little bit of hyperparameter (the parameters of our model) tuning, one should be able to improve the performance. So, go on and play around with the parameters of our models. Does the R^2-score improve with more trees in the Random Forest? What about the number of features in a tree? What about different models in general (logistic instead of linear regression)? Don't be afraid of new terms, a simple explanation is often just one google search away.

Hopefully this inspires you to create your own ML models on Galaxy. Just start, it's not that difficult :)

Thanks to Ekaterina Polkh and Anup Kumar, who inspired this tutorial with their Galaxy Training called "Age prediction using machine learning". Have a look there, if you want to learn more about the R^2-score, dig deeper into this specific data set and learn about model building via pipelines. Also check out all the other great ML tutorials available on the galaxy training platform (https://ml.usegalaxy.eu/training-material/topics/statistics/).