CHA2555 – complementary material on Machine Learning

Data preprocessing and feature importance

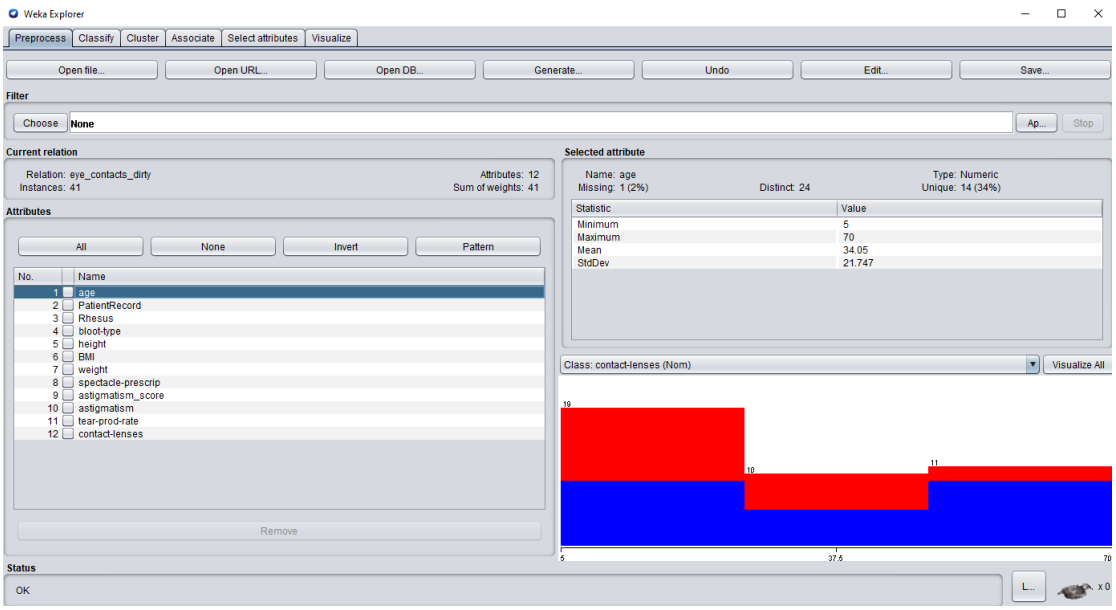
Dr. Emmanuel Papadakis

This mini-tutorial will guide you through common practices for preprocessing data before applying any ML algorithm. In particular, you will cover the following:

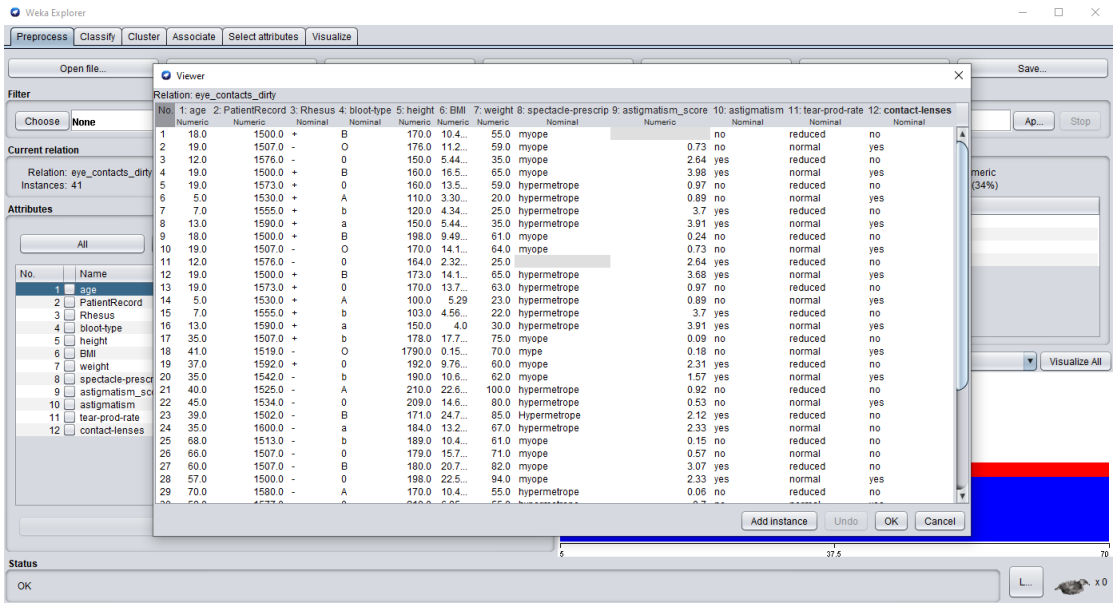
- Drop rows with missing values or replace with an estimate if applicable
- Ensure feature value consistency
- Spot and remove outliers
- Remove redundant or useless features
- Remove features with predictive power lower than 50%

Walkthrough

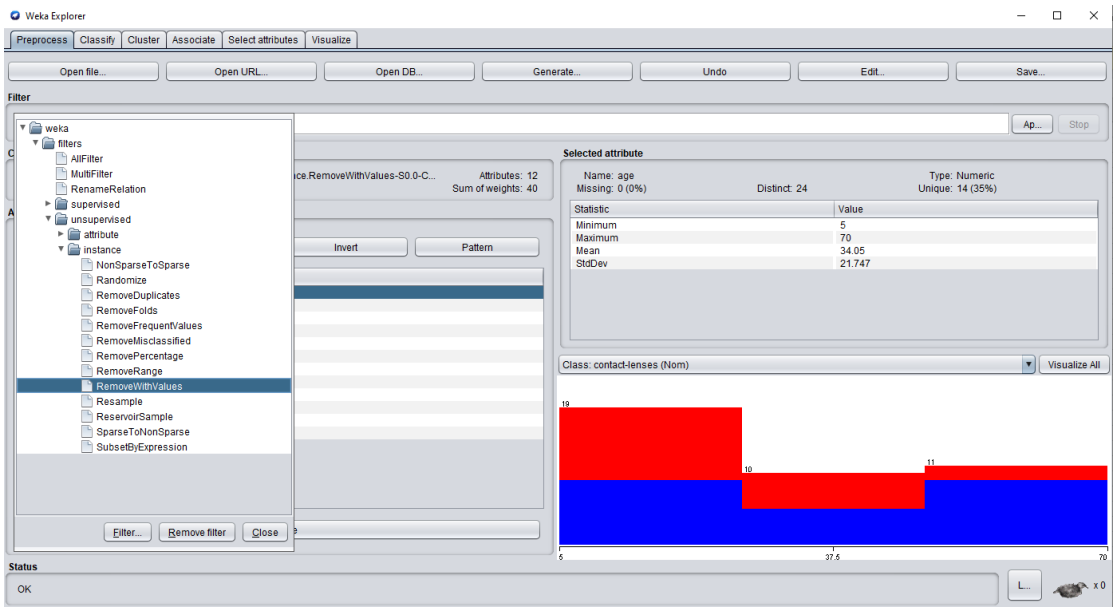
1. Download the sample data file eye_contancts_dirty.csv
2. Open Weka and select the application Explorer.
3. Click ***Open File…,*** Change the **Files of Type** to ***CSV data files (.csv)*** and navigate to the downloaded data file eye_contacts_dirty.csv. Select the appropriate file and click ***Open***.



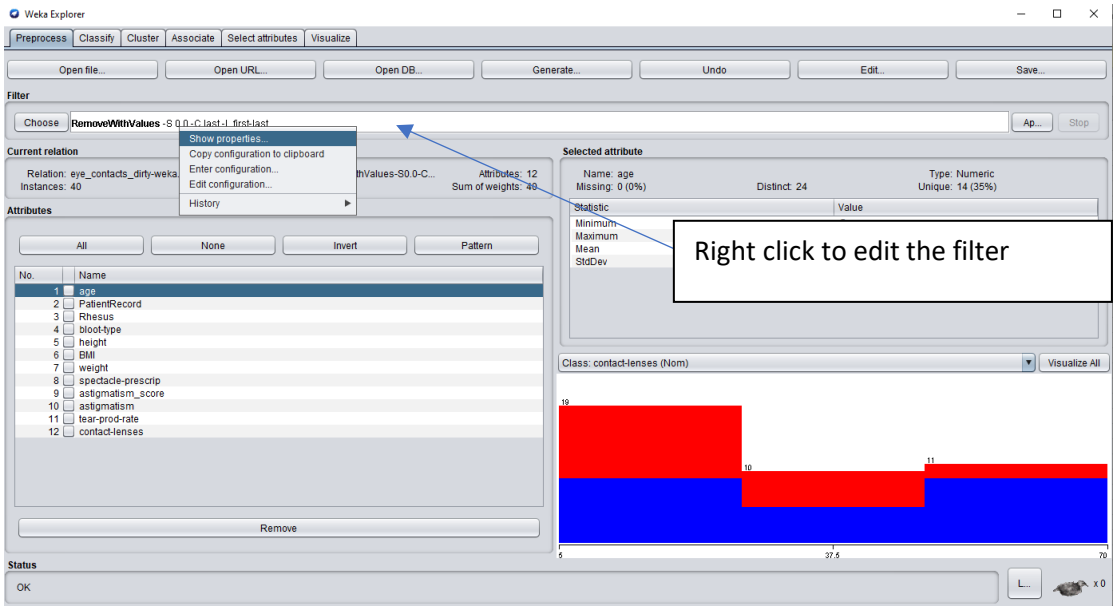
4. Inspect the data by clicking on ***Edit…***



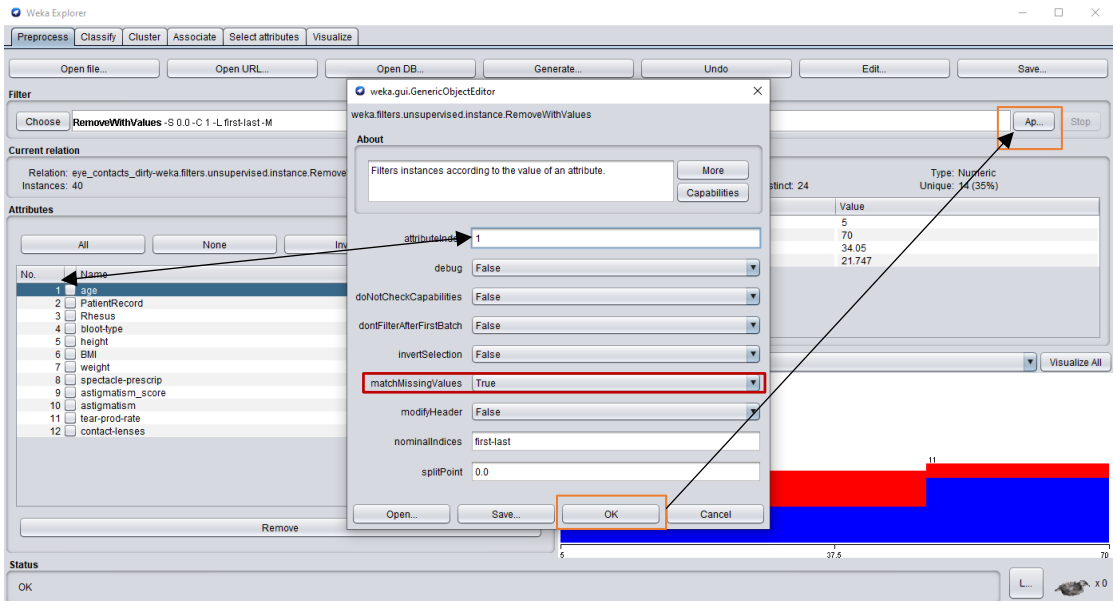
5. Spot features that contain at least one entry with missing values. The missing values are represented as gray cells, as shown above.
6. Remove data entries with missing values by applying filters
    a. Under the filter section click **Choose** and navigate to unsupervised -> instance -> **RemoveWithValues**



    b. Right click on the newly added filter (Text field next to the **Choose** button) and select **Show properties**

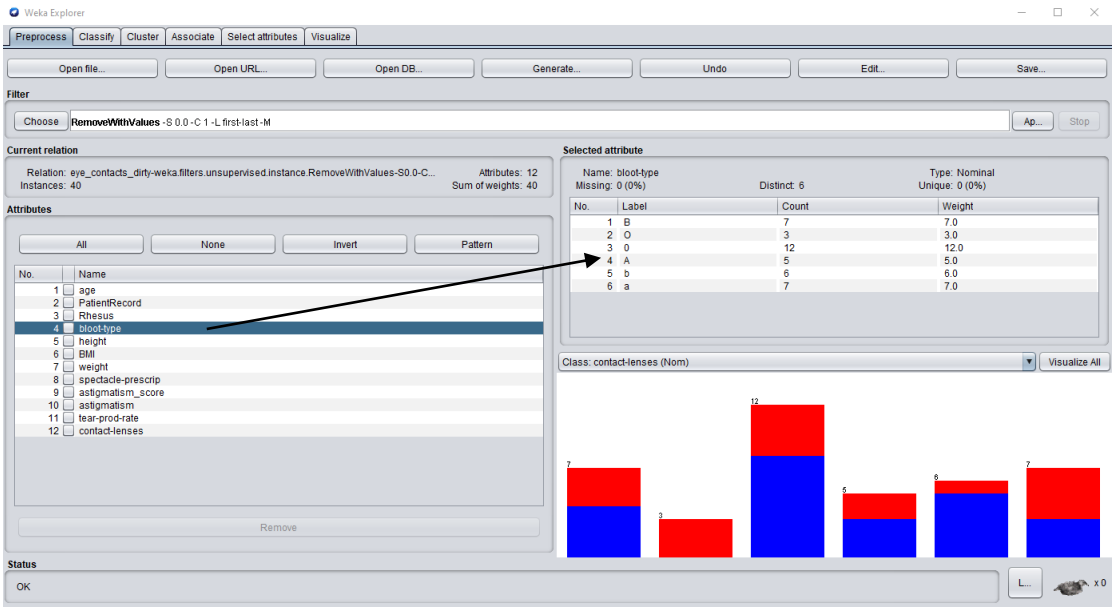


c. Change the field matchMissingValues to **True** and fill the attributeIndex with the index of the identified feature, which has missing values, for instance, age has index 1. Click OK and Apply the filter.

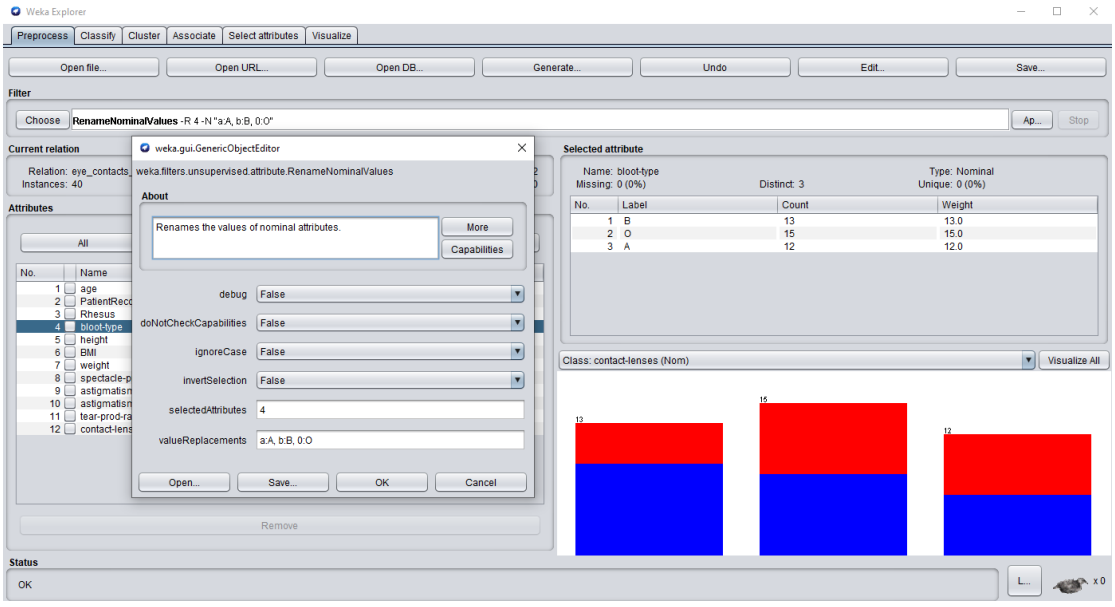


Check the raw data again (**Edit…**) to ensure that the filter deleted the entries, whose age was missing. Repeat the above steps for every other feature that you marked with missing values. The final view of the data **must not contain empty-gray cells**!

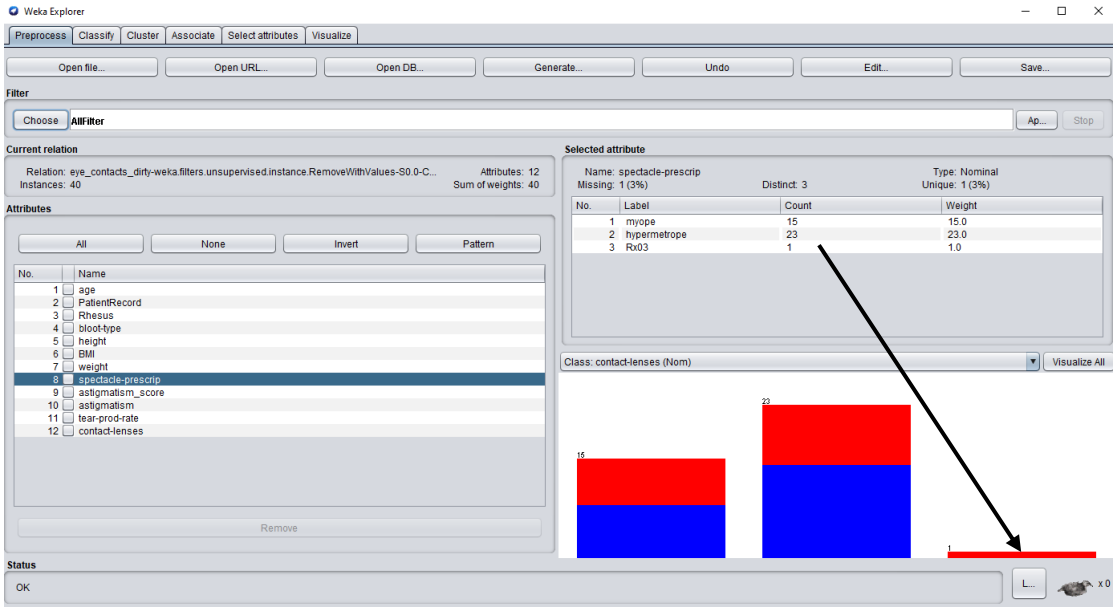
7. Inspect data for inconsistencies (For instance, blood_type includes classes in upper and lower case as well as 0 and O).



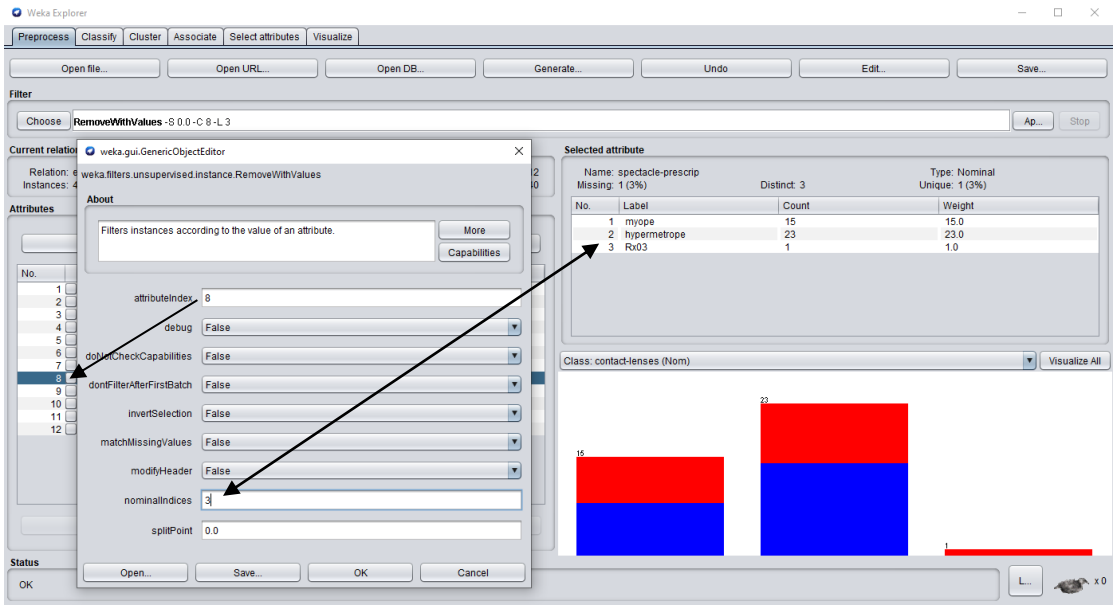
8. Fix inconsistencies by applying replacing filters.
    a. Navigate to the filter: unsupervised -> attribute-> RenameNominalValues and edit the filter as follows:
    b. **selectedAttributes** must be set on the index of the attribute you would like to edit (blood-type = 4)
    c. **valueReplacements** must have comma-separated tuples formed as: **findValue : replaceWith**. For instance, in the example below the filter will replace a with A, b with B and 0 with O.
    d. Apply the filter and observe the changes in the values distributions.
    e. Repeat this process to ensure that all non-numeric attributes are consistent.



9. Find outliers in the data
    a. Inspect the histograms of the attributes and spot outliers, for instance **spectacle-prescrip** has a nominal value Rx03, which not following the norm of myope/hypermetrope. It is best to remove it.



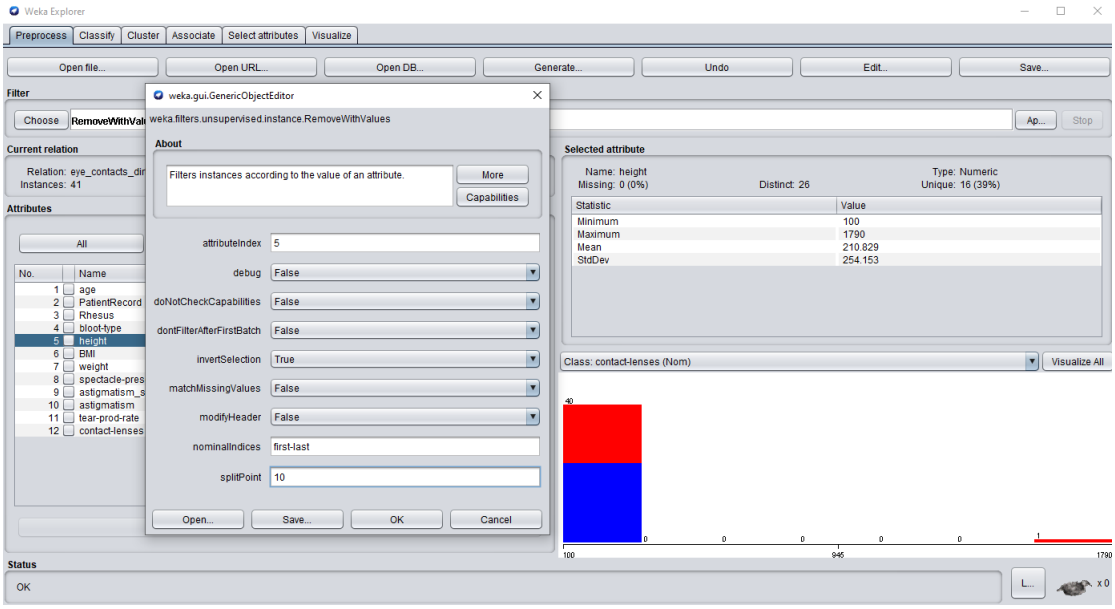
    b. Select and Edit the properties of the **RemoveWithValues** filter as shown below:
        - set the **attributeIndex** to the index of the attribute you are editing, (8 for **spectacle-prescrip)**
        - set the **nominalIndices** to the value index that contains the outlier (3 in the case of the outlier *Rx03*).



    c. Repeat this process with other attributes that contain outliers. In the case of numeric values, **RemoveWithValues** filter must be configured as follows:
        - **InvertSelection: False and splitPoint = 10**
            - Remove all entries with attribute value less than 10

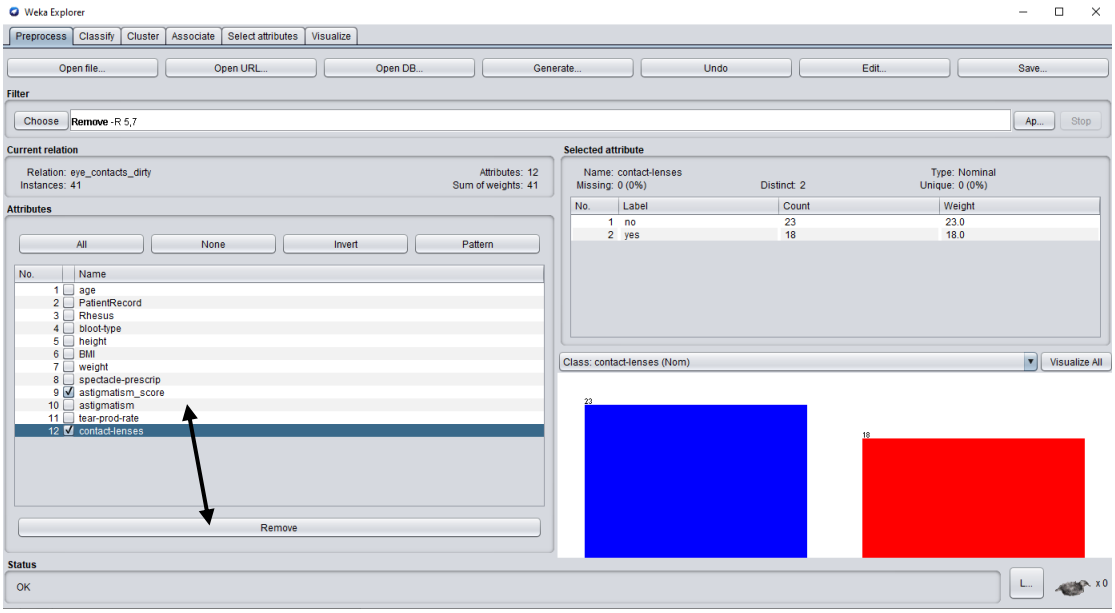
- **InvertSelection: True and splitPoint = 10**
    - Remove all entries with attribute value greater than 10

For instance, the example below will remove all entries whose height is greater than 10



10. Remove redundant or useless features

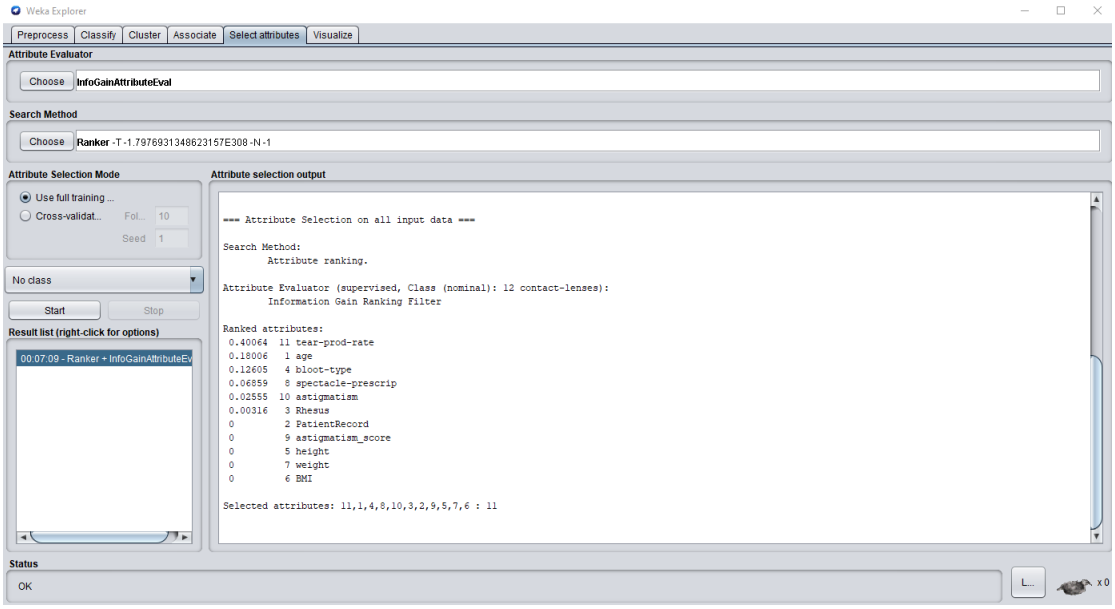
- **a.** Inspect the data and identify irrelevant features in respect to the goal of identifying whether a patient requires contact lenses (for instance ID related features).
- **b.** Do you notice something strange with the triple: height, weight, and BMI (body-mass index is the squared fraction of weight divided by height)?
- **c.** You may remove features by selecting their indices and clicking the **Remove** button.



With all the steps above, you can realize how elaborate but necessary is to preprocess and prepare the available data sets before trying to employ any machine learning algorithm to fit on the data. The final step is to drop features with low predictive power using the metrics: Information Gain.

11. Select the **Select attributes** tab and set
    a. Attribute Evaluator as **InfoGainAttributeEval**
    b. Search Method as **Ranker**
    c. Attribute Selection Mode: Use full training set
    d. Select the class feature **contact-lenses**

Click **Start** and observe the results…



You can now remove all the features that have predictive power lower than 50% and apply an ID3 or Bayesian classification.

How to convert the ranked features into a percentage of predictive power. Feature selection methods such as CfsSubSetEval + BestFirst detail features as a percentage of importance. However, in the case of information gain, the result is the actual information. If a feature has information gain equal to 0, its predictive power is negligible, hence it is safe to be removed as it can be a source of noise instead of a predictor. On the contrary, InfoGain values higher than 0 indicate the impact of the feature as a predictor against the target feature.

In order to convert this quantity into a percentage we use a simple normalization technique, which in essence transforms the given data a value between 0-100%

$$featureImport = \frac{actual\ value - \min value}{\max value - \min value} \cdot 100\%$$

- Actual value – the information gain of a feature
- Min/Max value – the minimum and maximum information gain encounted during feature selection, respectively.

MaxValue = 0.40064

MinValue = 0

tear_prod_rate	(0.40064 / 0.40064) = 1	100%
age	(0.18006/0.40064) = 0.44	44%
bloot_type	(0.12605/0.40064) = 0.31	31%
spectacle_prescrip	(0.06859/0.40064) = 0.17	17%
astigmatism	(0.02555/0.40064) = 0.06	6%
Rhesus	(0.00316/0.40064) = 0.007	0.7%
Rest features	0	0

The normalized values indicate that the only feature with adequate predictive power is tear_prod_rate. The threshold that can be used to drop insignificant features is usually defined by the user or set arbitrarily (at least 50%).