Algorithms for Data Science
Practical Report

Andrei Karpau
MSc in Applied Data Science and Analytics
Institute of Technology Blanchardstown
11.12.2016

Part 1: Classification

Dataset 1:

Introduction to the dataset and its meta data.

The dataset adult_mixed contains the following information about different people:

- Age attribute holds a person's age. It is of numeric type. It has values between 17 and 90.
- Workclass holds a work class information. It is a polynominal attribute. It has 7 possible values. One of them is a null value.
- Fnlwgt is a continuous demographic characteristic. It is a numeric value and holds the values in range between 19302 and 750972.
- Education is a polynominal attribute that holds information about person's education. It has 15 different nominal values.
- Education_num relates to the "education" attribute. It has only numeric values, but actually it is a polynominal attribute that has 15 different nominal values.
- Marital_status defines the marital status of a person. It is a polynominal attribute and has 7 distinct values.
- Occupation holds the information about profession of a person. It is also polynominal and has 14 distinct values. One of the is null.
- Relationship relates to marital status of a person. It is a polynominal attribute with 6 distinct values.
- Race is a race of a person. It is also a polynominal attribute with 5 distinct values.
- Sex defines a sex of a person. It is binominal attribute with 2 distinct values.
- Capital_gain relates to the increase of value of person's capital. It is a numeric attribute with values in range of 0-99999.
- Capital_loss relates to the losses of capital. It is a numeric attribute with values in range 0-2559.
- Hours_per_week is number of hours a person works per week. It is a numeric attribute that holds the values between 2 and 99.
- Native_country is a person's native country. It is a polynominal attribute that holds 29 distinct values including nulls.
- Label is a binominal attribute that has information about the income of a person. It has only 2 distinct values and is the label role.

Classification algorithms used

This data set has a lot of nominal values. The number of numeric attributes is much smaller and most of them do not have a very high range of values. The label is binomial. So, decision tree can be a good fit for dealing with this type of dataset and for understanding its structure. However, Decision Trees is rather simple model and can often miss some more complicated rules. Another potential problem is overfitting the model. So, I decided to try also an ensemble algorithm on this dataset. There are some good variants for it such as AdaBoost, Bagging or RandomForest. I have chosen the last one. For calculating the accuracy, the cross-validation is used with the number of validations set to 10.

Results of training using default parameters.

Building the decision tree with the default parameters (criterion – gain_ratio, maximal depth – 20, pruning with confidence level of 0.25 and repruning turned on) gives the 75.4% accuracy. It is not a bad result. Nevertheless, it makes sense to try to reach better results.

Random forest with default parameters, which is 10 trees with the same settings as default decision tree, shows a little bit better accuracy = 76.6%.

Results when training using modified parameter settings.

After looking on final decision tree model it is observed that the model is not very deep (much less than 20 splits) and the most of errors occur on numeric parameters. So, I decided that the pruning of the tree is too strong. The try to run it without pruning has shown better results on depth = 10 the accuracy is 78.9% with lower deviation values. But further increasing the depth of the tree decreases the accuracy, because of overfitting problem. Since, the model with depth = 10 can still overfit the dataset, I decided to turn on the pruning again. The pruning with confidence level of 0.5 has removed some noisy splits and increased the accuracy to 79.1%. Actually, it is the maximum accuracy observed.

For Random Forest at first the same setting as for decision trees can be used. So, I have turned off the repruning, decreased the depth of the tree and set the confidence level of pruning to 0.5. Here, it makes even more sense, because the overfitting of trees in the forest is negated by using the ensemble. The accuracy shown by this setting is 80.7% with deviation +-3.52%. Increasing the number of trees in forest to 30 has increased the accuracy to 81.4%. Then I tried to make the final model more complicated by increasing the maximal depth to 20 and the number of trees to 40. The final accuracy observed is 83.5% +/-2.8%. The further growing of the model seems not bring the increase of final accuracy because of the overfitting problem. So, 83.5% is the best accuracy observed on this dataset.

Patterns in the data

The patterns and dependencies can be easily observed on the decision tree model. The most important splits are made on capital_gain, age and capital_loss attributes. If capital_gain is higher than 6897, then the salary is >50K. If age is less than 24.5 or over 72, then the income is <=50K. If capital_loss is >2322, then income is >50K. These splits filter out near 22% of samples with almost 100% confidence. Also, very important role play relationship and education attributes.

Dataset 2:

Introduction to the dataset and its meta data.

The dataset adult_numeric contains the information about different people. All the values except label are of numeric type and are normalized. So, they have the averages equal to 0 and deviations to 1:

- Age attribute holds a person's age.
- Fnlwgt is a continuous demographic characteristic.
- Education_num relates to person's education. It is more a polynomial value, however since the numbers in the original dataset are ranged from low education to doctorate (from 2 to 16), the normalized numeric values make also sense and can be used in prediction.
- Capital_gain relates to the increase of value of person's capital.
- Capital_loss relates to the losses of capital.
- Hours_per_week is number of hours a person works per week.
- Label is a binominal attribute that has information about the income of a person. It has only 2 distinct values and is the label role. The number of rows in each category is almost the same.

Classification algorithms used

All the attributes in the training dataset are the numeric values. They are already normalized, so there is no need in further preparation of the dataset. The label is binomial. The first algorithms that is used is K-Nearest Neighbours. The second algorithm that can fit the data well is Support Vector Machines (SVM). Both algorithms work well with numeric data and predict the categorical (in case of SVM binary) labels. So, they can deal with the provided dataset.

Results of training using default parameters.

KNN its basic configuration, has k equals to 1, unweighted votes and mixed Euclidean distance used as a measure. It provides the accuracy equal to 65.4% with deviation +/-4.69%.

The default version of SVM uses dot kernel type, complexity constant set to 0 and convergence epsilon set to 0.001. SVM in its default configuration shows better results. The accuracy of the model is 74.00% +/-5.23%, which is a good result.

Results when training using modified parameter settings.

In KNN the most important setting is the number of nearest neighbours "k". The are several suggestions in the literature, which k can fit better. For example, k=log(n) that is a value near 3 or k=sqrt(n) that is a value near 30. So, in order to find out which value fits better the OptimizeParameters element is used that tries k values from 1 to 41 by 20 steps. The highest accuracy provided by KNN is for k=37 72.10% with variance +/-5.78%. This is a high improvement comparing to basic configuration but still is less than the value shown by SVM. The use of different measures seems not to have much sense in this dataset, so the default EuclideanDistance is used. Another one option "weighted vote" tends not provide much influence on the final results, especially when the value of k is high. Also, it is worth to note that the final accuracy is measured on a random seed that is used in X-Validation. Using the same seed that is used in other examples decreases the final accuracy on 1 percent, which can be produced by noise in the data.

The SVM has provided the high accuracy results even with default parameters. The data seems to be well divided by Dot kernel. From another kernel variants, the radial has provided the slightly worse result equal to 72.2% and anova 73.5%. All the other kernels do not fit the data well. Also, it is worth to adjust the C value (complexity constant) to get better results. The initial value of C is 0. Increasing it to 1 increases the accuracy to 74.7% on Dot kernel. The further increase of this value does not lead to higher results. Increase of C value for radial and anova kernels tend not to make any improvements or even reduces the accuracy. So, the best result for his dataset is shown by SVM with dot kernel and C values equals 1.

Patterns in the data

Unfortunately, it is very difficult to interpret the result of both SVM and KNN algorithms, so it is worth to take a look into the previous dataset example and the patterns found by decision tree.

Dataset 3:

Introduction to the dataset and its meta data.

The dataset SkeletalMeasurements contains different skeletal measures and 4 possible class labels: age, gender, weight, and height. All the values in data set are numeric and not normalized. The only value that can be considered as categorical (binominal) is Gender. So, when using gender as a label, it can be converted to binominal type and logistic regression can be used.

Classification algorithms used

The dataset SkeletalMeasurements has several possibilities of choosing the label. Gender is the only attribute that can be considered as binominal. So, when predicting the gender, it is set to binominal type and logistic regression is used. For all the other labels (age, weight and height) the linear regression algorithm is used.

Results of training using default parameters.

When predicting gender, the logistic regression is used. For validating the model and calculating the accuracy, cross validation is applied. The default logistic regression has AUTO solver, regularization being turned off and standardization being turned on. These parameters provide very high accuracy equal to 94.86% with deviation +/-2.98%.

The labels age, weight and height are of numeric type, so for predicting them the linear regression is used. Since the accuracy is not used for evaluating the linear labels, the root mean squared error (RMSE) and squared correlation (R squared) are considered. The linear regression with default parameters provides the following values:

- For height: RMSE is 5.612 +/-0.595, R squared is 0.644 +/-0.073. These values show that the model fits pretty good, but not perfect.
- For weight: RMSE is 4.639 +/-0.668, R squared is 0.887 +/-0.033. It is a very good fit.
- For age: RMSE is 8.878 +/-1.041, R squared is 0.145 +/-0.068. The R squared value is very low, the average error of more than 8.8 years is high. So, the model does not fit the data well.

Results when training using modified parameter settings.

For gender prediction, the default logistic regression parameters have shown very high results. All the tries of the other attributes including the usage of regularization with different lambda values and applying different solvers has made no improvements in accuracy.

For height and weight classification, the adjusting of parameters (feature selection and eliminate collinear features) has made only the slight improvement of result model, however it seems to be related to R squared and RMSE deviation, then to real improvements. For age classification turning off the feature selection slightly improves the R squared value to 0.155, but increases its deviation, so this cannot be considered as a real improvement either.

Patterns in the data

Looking in the coefficients returned by regression models the following conclusions can be made:

- In gender prediction, the highest role play elbowDiam, wristDiam and biacromial attributes. From the other side, the coefficient of chestDiam is almost 0, which means almost no influence of this attribute in gender prediction.
- For height, the important attributes are biacromial, pelvicBreath, elbowDiam and ankleDiam. The smallest influence has chestDiam.
- For weight the most important attributes are chestDepth, chestDiam, pelvicBreath, bitrochanteric and kneeDiam. The ankleDiam makes almost no influence.
- For age model, the most important are biacromical and chestDepth. Nevertheless, the overall model prediction is very low, so it is more reasonable to say that the age make almost no influence to the skeletal measurements and vice versa.

Part 2: Clustering

Introduction to the dataset and its meta data.

ClusterDataset2016 contains four numeric attribute values:

- Att1 has the values in range -7.129 – 4.767 with the average -1.504.
- Att2 has the values between -5.912 and 5.772. Its average is 0.135.
- Att3 has the values in range -7.250 – 7.087. The average is -0.290.
- Att4 has the values between -5.400 and 3.968 with the average -0.157.

The potential problem in the dataset are the distances of the attributes. Since, it is a generated dataset and it has no business objective behind it, it is impossible to decide if the units of all attributes are the same. So, it is worth to apply Z-normalization, to be sure that the distances of all attribute are of the same weight. The normalized dataset is visualized on 3d scatterplot on figure 1. On the x-axis is att1, att2 is on the y-axis. Z-axis is att3. The colours are related to att4.

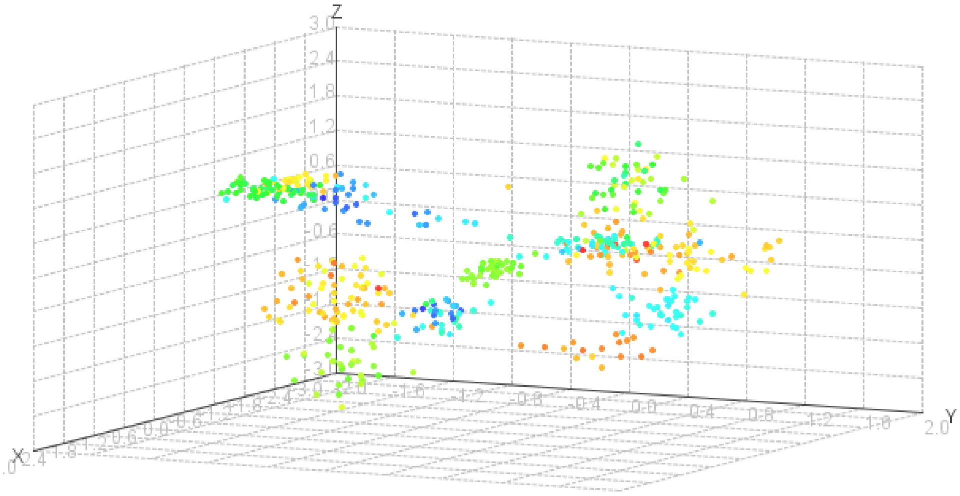

Figure 1 Visualization of normalized dataset

On the scatterplot, it can be observed that the data has some more or less defined clusters, although it is still difficult to find out the right number of them. Also, it is worth to note that the Euclidean Distance seems to be reasonable for this dataset, so it is used for both clustering algorithms.

Clustering algorithms used

The first algorithm used is a hierarchical clustering (agglomerative clustering) algorithm. It is worth to use this technique on the first step in order to build the overview of the clustering of the dataset. For analysing the number of clusters, the flattering is made in a loop after the hierarchical model is built.

On the scatterplot, there are some noise points. Excluding them, the inner cluster distances do not vary very much. So, DBScan can be good fit for this dataset.

Objective measures of clusters found

After building the hierarchical model it can be visualized as a dendrogram. Depending on the mode, different observations can be made. For single linkage, the observed number of clusters is 10-15, for complete linkage it is seems to be between 15 and 20, for average one the number of clusters is near 15. The next step is to add flattering and to run it in the loop with min equals to 5 and max equals to 25. It can be seen that after first 6 clusters, two one-item groups are added. It is very probably, that it happens due to noise. After it, the clustering successfully continues till 12 clusters. Here is the first breakdown. The second breakdown seems to be near 16 clusters. In case of average linkage and complete linkage, the breakdown seems to be somewhere near 20 clusters.

For evaluating the results, the cluster density performance and distribution performance are measured. For single linkage, the clear final break down in both density and distribution is observed on 20 clusters. Nevertheless, single linkage creates some noise groups that contain only one item. For average linkage, the density and distribution break down seems to be between 16 and 20 clusters. After looking on the scatterplot, it can be observed that 16 clusters provide almost no noise and look more reasonable. Complete linkage also has a breakdown between 16 and 20 clusters. However, after investigating the 3d scatter plots, 16 looks more reasonable, since the higher values seem to divide big clusters into smaller ones.

Before running the DBScan it is worth to investigate the set of similarities between all the items. On the histogram of similarities, two peaks can be observed. The first one, which is the low peak, starts from 0, has the top point near 0.4 and ends between 0.75 and 1. This peak is related to the clusters. The second peak starts afterwards and is much higher. It is related to distances between the clusters. So, after investigating the diagram epsilon value between 0.75 and 1 can be considered. In order to determine the value of K, k-distances plot can be explored. For epsilon equals 0.75 and k equals to 4 only 3 items have the distances higher than 0.75. Changing k to 5 makes much more items above 0.75, so k equals 4 seems to be the best variant. For epsilon equals to 1, the k value between 6 and 8 can be chosen.

After running the DBScan with epsilon = 0.75 and k = 4, 10 clusters are created. The smallest one has 5 items and the biggest one 158. One additional cluster (cluster 0) has 2 items considered as a noise. Changing epsilon to 1 and k to 6 reduces the number of clusters to 7, but has no noise items at all. In this case, there is one big and stretched group that contains 302 items, 5 middle size groups and one small group with 8 items.

Conclusion

The hierarchical clustering has provided the good overview of the possible clustering possibilities. Since, the dataset has different forms and sizes of clusters and some noise, it is impossible to define the precise number of clusters. The hierarchical algorithms have provided different reasonable results between 16 and 20 clusters. In this case, almost all the clusters are of similar size, there is no very big clusters. However, some small clusters can be considered as noise.

DBScan provides from 7 to 10 groups. In case of 7 clusters, one big cluster is not properly divided, however it can make sense under some business perspectives. The smaller epsilon values divide the big cluster into smaller ones, but in this case 2 rows are classified as noise.