Università degli studi di Milano - Bicocca
Appello Febbraio 2020

Marketing segmentation with machine learning

Cesareo Giacomo[1], Banfi Davide[2]

Abstract

The annual income of every person is designed to reflect several factors such as the age, the education, the work class, his mansion and many others.
Knowing the total income of a range of people can be very useful especially for companies who plan to build their business upon specific products. For instance, it's quite useless developing a set of expensive items to be sold in an area in which the average income is barely enough to sustain the cost of living.
For this reason, we decide to discover the best predictive model through machine learning techniques in order to find a concrete solution to solve our problem.
We chose the database of the 1994 US Census and specifically the subset extraction made by Barry Becker. This dataset contains weighted census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau. B. Becker extracted from the whole database a set of reasonably clean records meeting the following conditions:
1) age had to be higher than sixteen
2) the average gross income being higher than 100
3) weekly working hours higher than 1
4) final weighting higher than 1.
The concerned dataset can be found on Kaggle under the name of "Adult Census Income". Our goal is to predict whether the yearly income exceeds 50K value.

[1] *Università degli Studi di Milano Bicocca, CdLM Data Science, matricola: 805716*
[2] *Università degli Studi di Milano Bicocca, CdLM Data Science, matricola: 806539*

Contents

1. Introduction

Inspecting our dataset before analyses

The dataset[1] consists of 32.561 units and 15 variables:

- age: it's a numeric variable representing the age of every person.
- workclass: it represents the public or private sector which the occupation of the worker belongs to.
- fnlwgt: the weight for a responding unit in a survey data set is an estimate of the number of units in the target population that the responding unit represents. In general, since population units may be sampled with different selection probabilities and since response rates and coverage rates may vary across subpopulations, different responding units represent different numbers of units in the population.
- education: categorical variable with the reached degree.
- education.num: numeric variable describing the years of study for every individual.
- marital.status: contains the civil status.
- occupation: categorical variable describing the mansion of the person.
- relationship: indicates the family status.
- race: categorical variable with the race.

[1] Cfr. with the official site https://www.census.gov/

- sex: gender of the person.
- capital.gain and capital.loss: these two columns collect the total amount of money gained or lost from financial activities such as investments.
- hours.per.week: numeric attribute to describe the number of working hours per week. The dataset was created extracting from the Census having this parameter higher than 1.
- native.country: individual's birth country.
- income: this one is a binary variable and it is our target attribute. It represents the total yearly income and takes values equals or below 50K and above 50K.

It has to be pointed out that our data is characterized by a strong class imbalance between the modalities of the attribute income and this may cause several problems if not handled correctly. As explained in the abstract, the aim is to give an accurate forecast of the rarer case occurrences, so we chose the class >50K as the positive class to study.

What we are going to do is to build several classification models[2] and confront the obtained performance measures in order to find out which one provides a trustworthy predictive pattern.

The report is structured in a way that in the first paragraph we illustrate a general overview of the preprocessing operations carried out and how missing values were handled. Later on, we will paste and comment the results obtained by the classifiers and compare the output measures on both the validation and test sets. The body of this analysis is such that a great attention is given to both cross validation[3] and feature selection[4] and the variations they accomplish on performance. It is quite important to underline how things change when the records that fall in the validation set can be somehow "controlled" and also when the number of useful attributes can be reduced.

In the last paragraph the cost analysis approach is presented, thus we'll see how every classifier behaves when costs of misclassification are present.

It is strongly recommended to confront with the KNIME workflow while reading as notes were added to ease the lecture.

2. Preprocessing

Enhancing the dataset

We noticed that a good percentage of people aged above 65 or below 23 had both occupation and workclass attributes unknown. Their occupation was assigned respectively to "retired" and "student" so as to not omit a consistent percentage of the database. At this point we applied a complete record removal technique which cut out around 1500 rows.

In two additional columns we grouped the values belonging to native.country in connection with the country's continent, as well as the difference between capital.gain and capital.loss was computed. It seemed reasonable to consider the difference rather than these two variables separately, being this last more significant. We rather took into account a binarization column of marital.status having values 1 as "yes" and 0 as "no". Some classes within workclass and occupation were aggregated thus reducing the number of modalities[5] . Irrelevant attributes in the determination of income such as race, native.country, relationship were cut out reducing the dimensionality of the dataset.

In the last part of preprocessing metanode we went through a normalization of both fnlwgt and capital.difference attributes. This was made with the purpose to transform features on a similar scale and improve the performance and stability of the models. Capital.difference has been transformed on a logarithmic scale to reduce the distribution skewness. Eventually only the normalized variables were used due to the presence of many negative values in logarithm's argument.

After all these preprocessing activities we have the final dataset on which the work relies, hence below the distribution of our target variable is pasted.

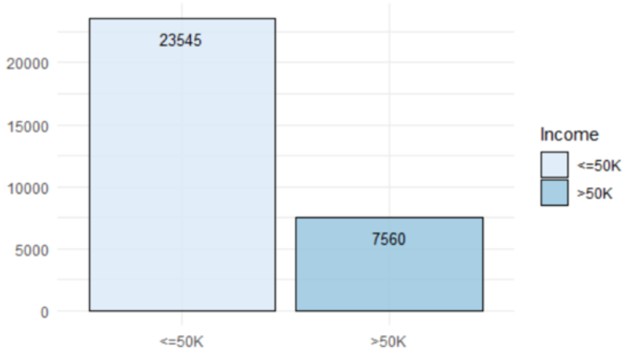

Figure 1. the histogram represents the strong class imbalance as the 75.69% falls in the negative class of our target attribute.

[2] A classification model scans the input data known as training set and tries to draw a pattern for future input values.

[3] Cross validation is a resampling procedure used to evaluate machine learning models. It splits a limited data sample into k groups and for each group it computes a partial accuracy using groups different than the kth as input data. Total accuracy is given by the mean of kth partial accuracies.

[4] A feature is an individual measurable property of a phenomenon being observed.

[5] White collar workers are suit and tie workers who work at the desk, blue collar workers carry out mainly manual jobs.

How did preprocessing affect variables statistics?

To this purpose three numeric variables of choice age, fnlwgt and hours.per.week were confronted.

Comparing the results on the workflow for the three variables we observe that all the statistic indexes do not change substantially.

We conclude that the record removal did not statistically affect the distribution.

3. Classification

Results of the inducer on the test set

The dataset was stratified partitioned into a train set and a test set, the former being the 80% of the whole. A second partitioning node further divided our first train set into a 67% train subset and a 33% validation set.

A SMOTE node that artificially enriched our train set was applied to mitigate the effects of class imbalance. It was our focus to study the different behavior of the main classifiers[6] seen during the course, having their inducers built upon the second partitioned train set. Every classifier achieved on the validation set the following performance measures.

Table 1. performance achieved by the classifiers on the validation set for income value >50K.

Models	Accuracy	Precision	Recall	F-Measure
RandomForest	0.837	0.659	0.683	0.671
J48	0.825	0.608	0.784	0.685
Logistic	0.807	0.570	0.823	0.674
SimpleLogistic	0.807	0.569	0.823	0.673
NaiveBayesLearner	0.803	0.779	0.257	0.387
NaiveBayes	0.839	0.736	0.519	0.608
BayesNet	0.843	0.664	0.706	0.685
NBTree	0.820	0.600	0.791	0.679
SPegasos	0.440	0.302	0.993	0.463
SMO	0.780	0.529	0.860	0.655
MultiLayerPerceptron	0.835	0.636	0.750	0.688

Considering mere accuracy may cause problems of interpretation because a classifier may accurately predict many negative records but fail to predict the few positive records of interest.

For example, the SPegasos retains a very high recall but a very low precision. This is definitely not favorable because the model may correctly classify many records whose income is ≤50K but fail to classify our priority target value >50K, whose misclassification is certainly heavier and more cost sensitive.

It has to be reminded that recall is defined as the ratio of positive records correctly classified by our algorithm while precision is defined as the percentage of entries that have been classified correctly amongst all the predicted positive records.

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

Another valuable performance index is F-Measure, a combined metric computed as the harmonic mean between recall and precision that indicates the tradeoff between these two measures.

$$F-Measure = \frac{2*Recall*Precision}{Recall+Precision}$$

Precision, recall, F-measures and ROC curves have to be considered at the same time. Looking at the measures in table 1, we decided to focus our attention on the five most common classifiers: the J48, Logistic, BayesNet, NBTree and the MultiLayer Perceptron[7] . From this point, we will consider just the above models and show how they performed on the test set.

Table 2. performance achieved by the classifiers on the validation set for income value >50K.

Models	Accuracy	Precision	Recall	F-Measure	AUC
J48	0.818	0.595	0.789	0.678	0.865
Logistic	0.801	0.560	0.841	0.672	0.889
NBTree	0.841	0.661	0.712	0.685	0.903
BayesNet	0.818	0.593	0.806	0.683	0.900
MultiLayerPerceptron	0.835	0.636	0.755	0.690	0.902

The corresponding ROC curves are pictured here.

[6] A classifier develops an algorithm that maps the input data to a specific category, known as inducer.

[7] For those who want to delve deeper into the topic, we recommend reading this article: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623?gi=f253a2a01a80

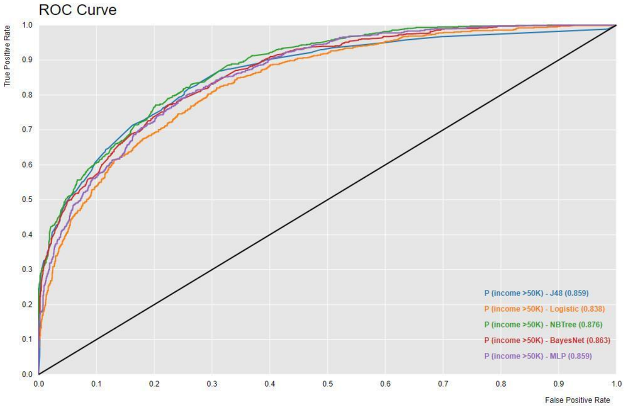

Figure 2. ROC curves described by the classifiers when run on the test set. The curves represent the ratio between False Positives and True Positives and the area under the curve appoints the performance of the model.

A ROC curve tells us how well a model is capable of distinguishing between classes – the closer the curve is to the left border, better the model is at predicting correctly the elements of the main diagonal of confusion matrix.

The area under the curve instead represents the ability of the model to distinguish between positive classes and negative classes and takes values closer to 1 when the model is really well performing.

As we can see every classifier carries out an overall good performance with respect to holdout[8] method, although the Multilayer Perceptron acts a little bit better whilst the J48 is the least performant.

By a confrontation of the two tables above we notice that accuracy and precision slightly decreased but the recall underwent an increasement when the inducers were applied on the test set. We must not forget that holdout is highly subject to distortion and overfitting as we can not establish prior which records would fall in the test set.

This is the reason behind our choice of applying a k-fold stratified cross validation, that is a process through which the total accuracy was determined as the mean of accuracies calculated on 10 varying subsets of the train set, letting us partially reduce the faults coming from sampling.

Accuracy and other measures can be found on the workflow while here we focus on this chart to illustrate the error difference rate when using a classifier rather than another one.

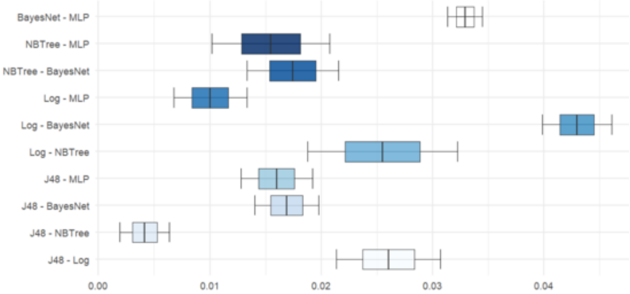

Figure 3. the chart pictures a 90% confidence interval of all the combinations of errors differences amongst classifiers, having applied cross validation. It's irrelevant using a classifier than another in cases when the box plot area is smaller.

The chart above depicts how the confidence intervals of errors differences are minimal between BayesNet and MLP classifiers, and maximum (wider box area) between Logistic and NBTree. Since accuracy is defined also as 1-error we conclude that classifiers having a very small box area will likely have a very similar accuracy value because the difference between errors is minimal.

In the next chapter we'll see how feature selection changes these confidence intervals when applied in concurrence with cross validation.

4. Feature Selection

Evaluating measures filtering out worthless columns

At this point of the analysis it is likely that we ask ourselves whether the computed measures depend on specific columns, and whether removing those may improve the interpretability but affect significantly the output on the other side.

The methods we focused on were the univariate and multivariate filter as well as backward and forward wrapper[9] , run on a second partition of the initial train set made with the 67% of the records. We refer you to take a closer look at the workflow for what concerns the very next steps.

Every classifier has been developed on the second partition's train set and measured on its validation set. In particular, the maximum number of attributes to be chosen for the univariate filter technique was set to six.

The Weka predictor outputs returned us a list of ranked attributes which are considered to be crucial for the measures estimates. Those attributes that weren't on the list were excluded from both the first partition's test and train sets (the

[8] Holdout is when we split a database into a train set and a test set. The train set is what the model is trained on, while the test set is used to see how well that model performs on unseen data. Cross-validation is usually preferred because it gives the model the opportunity to train on multiple train-test splits, while holdout depends on a single train split.

[9] Filter based methods use mathematical evaluation functions that are based on innate properties of the features. Wrapper methods consider the performance of a classifier to build the ranked attributes.

80-20 partition), following a computation of performance.

Once again, we believe that pasting comparison results than every single result would be more constructive. It is noteworthy to look at all the accuracies obtained on the test set and see which feature selection technique performed with higher accuracy.

Table 3. this table represents the best feature selection technique which renders the best accuracy for every classifier. We'll use these values in chapter four.

Models	Accuracy	Method
J48	0.826	Univariate
Logistic	0.799	Univariate
NBTree	0.832	Univariate
BayesNet	0.853	Forward
MultiLayerPerceptron	0.829	Forward

It's easy to notice that BayesNet achieves the best accuracy when we apply a forward wrapper. For heuristic and regression classifiers instead the preferred feature selection is a univariate filter, which returned us the list of key attributes: capital.difference, married, education.num, age, hours.per.week and occupation.

For all these methods a 90% confidence interval for accuracy was computed and by consulting the workflow you can check that all the values fall in the estimated interval.

We certified in the fourth section how events change when cross validation is applied. For this reason, we decide to see the difference between our results whether we apply a feature selection and a cross validation.

Table 4. CV performance without feature selection.

Models	Accuracy	Precision	Recall	F-Measure
J48	0.827	0.617	0.759	0.681
Logistic	0.801	0.561	0.824	0.668
NBTree	0.826	0.609	0.797	0.690
BayesNet	0.844	0.668	0.709	0.688
MultiLayerPerceptron	0.811	0.580	0.806	0.674

The Logistic classifier is the only one that improves when a feature selection is applied along the cross validation. Its accuracy increases with an adequate tradeoff between recall and precision.

On the counterpart, all the other classifiers become reliably worse as their accuracies tend to decrease by 2.5% average.

Table 5. CV performance with feature selection.

Models	Accuracy	Precision	Recall	F-Measure
J48	0.807	0.572	0.814	0.672
Logistic	0.829	0.611	0.811	0.697
NBTree	0.820	0.595	0.814	0.688
BayesNet	0.819	0.594	0.807	0.684
MultiLayerPerceptron	0.799	0.558	0.829	0.667

We conclude the feature selection chapter by saying that when we cross validate subsets of the train set, the classifiers generally perform better without feature selection apart from the Logistic. However, this does not have considerable statistical variations.

5. Cost analysis

Introducing misclassification costs

Errors of misclassification may be expensive. For instance, let's assume we are a company that needs an overview about the customers' budgets before delivering our costly products in that area. Let's suppose also that a non-optimal classification model scanned the area as non-profitable as most customers have been classified having an income below 50K. Following this result, the company decides to move elsewhere his products distribution.

If the customers have been classified wrongly and a good percentage of them actually gains over 50K, moving the distribution would end up in a big revenue loss for the company. This situation is indeed more expensive than a situation where the company decides to settle but a wide range of poor customers have been classified as wealthy.

Introducing costs is a useful way to weigh misclassification errors. The following cost matrix has to be considered.

		Predicted class	
		≤50K	>50K
Actual class	≤50K	0.0	3.0
	>50K	7.0	-1.0

False negatives were assigned a value of 7 because it's the worst expensive scenario we plan not to happen. False positives were assigned 3 as value because it has a lower impact on costs.

How does a classifier behave when we consider costs of misclassification?

Typically, a bad classifier takes into account that a mistake may be really expensive thus it behaves in a way such that it prefers identifying records as False Positives or False Negatives rather than attempting to risk the prediction output. The result is that only a small percentage of records are correctly classified as TP or TN and therefore the classifier's accuracy and precision decrease while recall increases. Up to this point we can say with a certain trust that our five classifiers are quite good, so we don't expect drastic changes when costs are introduced.

All of their inducers were run on the first partition's test set. Their new performance measures are now:

Table 6. accuracy of classifiers when costs are set without feature selection

Models	Accuracy	Precision	Recall	F-Measure
J48	0.801	0.561	0.829	0.669
Logistic	0.710	0.454	0.941	0.612
NBTree	0.780	0.528	0.902	0.666
BayesNet	0.816	0.585	0.839	0.689
MultiLayerPerceptron	0.759	0.503	0.927	0.652

By confronting table six with table two, we can clearly see that recalls increased by an average 10% while accuracy decreased by an average 4%. We conclude that our classifiers responded quite well. An utter comparison was made when we filter out useless attributes determined by the feature selection.

This time we did not apply again all the feature selection variants to avoid a chaotic workflow, instead we chose the preferred method which previously rendered us the best accuracy seen in table 3. For the classifiers J48, Logistic and NBTree we took in consideration only ranked attributes the univariate filtering returned us. The same applies for the forward wrapper with the remaining two classifiers.

Results are as follows:

Table 7. performance measures achieved when costs are set with feature selection.

Models	Accuracy	Precision	Recall	F-Measure
J48	0.783	0.532	0.886	0.665
Logistic	0.709	0.453	0.944	0.612
NBTree	0.783	0.532	0.911	0.672
BayesNet	0.807	0.569	0.853	0.683
MultiLayerPerceptron	0.765	0.509	0.917	0.655

In particular, if we confront table seven with table six we verify that when costs of misclassification are present the classification model is more "cautious" in predicting, thus accuracy is lower and recall is higher.

Using the same cost matrix we computed the total cost sustained by every classifier with and without feature selection.

In short we are asking ourselves how do costs and performance change when we reduce attributes?

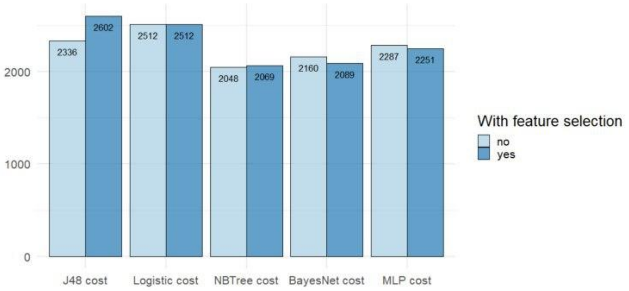

Figure 4. comparison of total costs sustained by classifiers whether feature selection is applied.

We can see how the J48 classifier achieves the biggest difference with an increasement of costs by 266, while Logistic's cost does not vary.

Being the models quite similar we must select the model which gives the best tradeoff between costs and performance. If we prefer a cost oriented approach we would definitely not choose J48 or Logistic because they achieve lower accuracy and higher costs rate.

On the counterpart, if we prefer a performance oriented approach, we would choose BayesNet or NBTree.

6. Conclusions

Final observations

The analysis that brought us to the prediction of our target attribute showed some important results.

Given our class imbalance we verified that accuracy is not enough to determine when a model is well performing. Basing our scores on all the metrics, the model that seems to perform better is Logistic when we apply cross validation and feature selection. It has a precision of 61% and a recall equal to 81% which turn into a 70% F-measure (the highest value amongst all models).

As expected, cross validation balanced the overall performance: by this means, some classifiers who had high precision and low recall met a growth of recall and a reduction of precision, and vice versa.

During the project, we tried to implement different approaches to improve our models, speaking about feature selection and cross validation. All the respective comparisons are available on the report.

The workflow provides several frames, or sections, in which we provide the opportunity to confront between all models and gives access to many more tables and charts.

Feature selection does not have a consistent impact on scores: some models perform better when all attributes are present.

When we introduce costs, things change. We have a relatively high recall because classifier prefers not to include records in confusion matrix cells which implicate high costs. In fact, if we check confusion matrixes on the workflow we always have a very small number of FN because it has been verified the classifier prefers cataloguing records as FP. This was due the very high cost we set for a misclassification of FN.

Back to the context, we were the company whose problem was the segmentation of the product distribution and the identification of a profitable area. By observing high recalls – thus the capability to correctly identify potential customers – in conjunction with the relatively low costs, we are provided with enough parameters to weigh our decision.

We conclude by saying the estimates we have found are not excellent but are quite good for prediction.

7. References

- https://www.census.gov/
- https://www.kaggle.com/uciml/adult-census-income
- https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623
- https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f

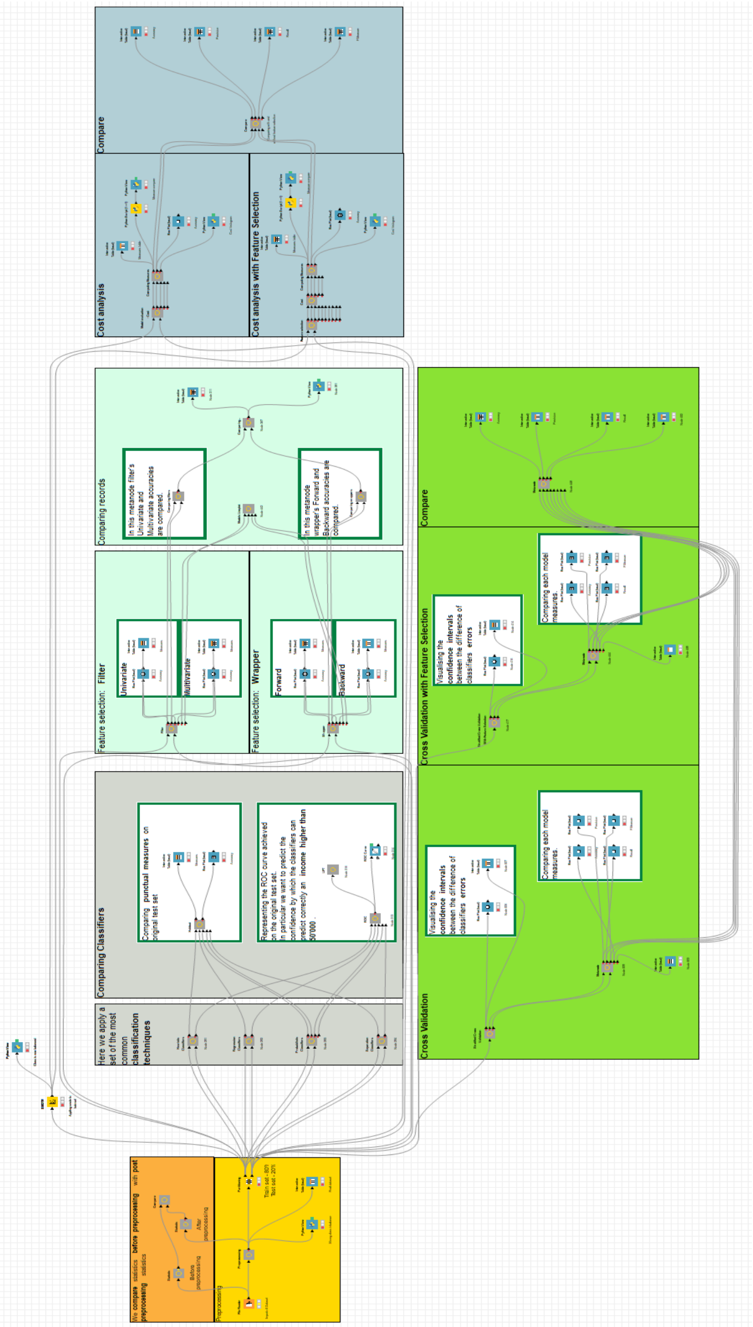
Compare
Cost analysis
Cost analysis with Feature Selection
Comparing records
In this metanode filter's Univariate and Multivariate accuracies are compared.
In this metanode wrapper's Forward and Backward accuracies are compared.
Feature selection: Filter
Univariate
Multivariate
Feature selection: Wrapper
Forward
Backward
Comparing Classifiers
Comparing punctual measures on original test set
Representing the ROC curve achieved on the original test set in particular we want to predict the confidence by which the classifiers can predict correctly an income higher than 50000.
Here we apply a set of the most common classification techniques
Compare
Cross Validation with Feature Selection
Visualising the confidence intervals between the difference of classifiers errors
Comparing each model measures
Cross Validation
Visualising the confidence intervals between the difference of classifiers errors
Comparing each model measures
We compare statistics before preprocessing with post preprocessing statistics
Before preprocessing
After preprocessing
Preprocessing
Train set - 80%
Test set - 20%

Figure 6. Knime WorkFlow