University of Ljubljana – Faculty of Economics
Course: Business Intelligence and Analytics
Course director: Full Professor Jurij Jaklič, PhD

Study programme: Business Informatics
Study year: 2018/2019

Project work

in the course

Business Intelligence and Analytics

Churn modeling and prediction in RapidMiner

Students: Eva Murko, Giacomo Tomada, Qiao Zheng
Student numbers: 19197550, 19215176, 19212070

Ljubljana, May 31, 2019

Table of contents

Index of graphs

Index of tables

1 Introduction

Our project is dealing with a problem called customer churn which occurs when customers stop doing business with a company. The companies are interested in identifying segments of these customers because the price for acquiring a new customer is usually higher than retaining the old one (Orac, 2019).

In our case scenario, telecommunication company approached a consultancy company in the field of analytics with a problem of identifying customers most likely to churn. Project will present various solutions for churn prediction for this specific company, based on their data set, with the help of different data mining techniques, used in RapidMiner. Our goal is to test different algorithms in order to present the client with the best interpretable one to help them solve the churning problem.

Next chapter will present our business case, followed by requirement definition and description of data and metadata. Forth chapter will include explanation of our main development steps, data gathering, cleansing and preparation. Solutions, achieved with different algorithms, will be introduced in the next chapter, presenting the main part of this project. Towards the end, the comparison between all the models will show which one to choose, with the help of the ROC curve. Last but not least, we will describe how can this communication company act further, what business value will our solution bring them.

2 Business case assessment

A good churn prediction model can help companies to prevent customer from churning, by obviously taking the right countermeasures. In short, the model predicts whether a customer is likely to churn or not based on his attributes. Some attributes might be more determinant of his decision, others might be not useful at all. The added value of a good churn prediction model is straightforward to comprehend: companies are interested in identifying common characteristics of churning customers because the price for acquiring a new one is likely higher than retaining the old one. The main purpose is to find their common features and work for preventing customer churn (reducing the price, modifying the contract etc.) It is worth mentioning that the best model is not necessarily the easiest to interpret: the actions to take to prevent customer churn are based on the interpretation of the model, so it would be best to find a good compromise between “accuracy” and “interpretability”. We will examine this concept more thoroughly in the following chapters.

As mentioned above, in our business case scenario we work as data analysts in a consultancy company that operates in the field of analytics. A telecommunication company sent us their dataset, asking to build the best possible model for churn-prediction. Since we offer an end-to-end service, we also had to take care of the interpretation of the results, providing some suggestions on how to prevent customer churn. We therefore used different data mining techniques to fulfil our client’s demanding request.

3 Requirement definitions

To predict whether a customer will be a churner or non-churner, there are a number of data mining techniques applied for churn prediction, such as artificial neural networks, decision trees, support vector machines, … For data mining we will be using a tool called RapidMiner.

Data mining has emerged over recent years as an extremely powerful approach to extracting meaningful information from large databases and data warehouses. The methodology of data mining views the discovery of information from a database as a four-step process. First, the business problem must be identified. After the problem is defined and related data are collected, the next step is to process the collected data by data transformation, data cleaning, etc. for the later mining process. The third step is to apply some specific mining algorithm(s) over the processed data. In this project, prediction/classification algorithms will be used. Finally, the mining result is evaluated to examine whether the finding is useful for the business problem (Tsai & Lu, 2010).

Very important step is data preparation (cleaning, transformation, coding…). We used Microsoft Excel for this step. Some classification algorithms only accept categorical and numerical attributes, like logistic regression, so it is important to focus on this step, in order to prepare your data in the right way before applying a certain algorithm.

After data are pre-processed, knowledge discovery algorithms can be applied to the processed data. What type of algorithms to use depends on the nature of the problem. If the problem can be viewed as an issue of classification or prediction, and a complete set of training data is available, then the problem is well structured, as it is in our case as well. Supervised learning algorithms like multilayer neural networks, regression, or decision trees can be used to learn the relationship between variables and correct decisions.

4 Data and metadata analysis

In the raw dataset, there are information about 7043 customers with 21 features, structured in 7044 rows and 21 columns. Among the attributes in raw data, 6 attributes are binominal, 11 are polynominal, 4 are numerical. The attributes can be classified into demographics, services and subscriptions, and expenses records.

Demographics	Customer ID	Customer’s unique ID number.
	Gender	If the customer is “Male” or “Female”.
	Senior Citizen	“1” means the customer is a senior citizen, “0” means the customer is not a senior citizen.
	Partner	If the customer has a partner or not.
	Dependents	If the customer has any dependents or not.

Services and subscription	Phone services	Phone Service	If the customer has phone service or not.
		Multiple Lines	Has the customer subscribed to have the multiple lines service or not. If the customer doesn't have phone service, it's recorded as "No phone service" which means not applicable.
	Internet services	Internet Service	If the customer has internet service or not.
		Online Security	All the extra internet services the customer can subscribe to. "Yes" means the customer has the certain extra service, otherwise it's "No". If the customer doesn't have internet service, it's "No internet service" which means not applicable.
		Online Backup	
		Device Protection	
		Tech Support	
		Streaming TV	
		Streaming Movie	
	Payment services	Paperless Billing	If the customer has paperless billing or not.
		Payment Method	Customer can choose among 4 payment methods. The customer can pay by "electronic check", "mailed check", or two automatic payment method "bank transfer" and "credit card".
Expenses records		Contract	Contract term of the customer. There're three types of contract, which are "month-to-month", "one year", and "two year".
		Monthly Charges	The amount of money charged to the customer monthly.
		Tenure	Number of months the customer has stayed with the company.
		Total Charges	The total amount of money charged to the customer.
		Churn	Has the customer churned or not.

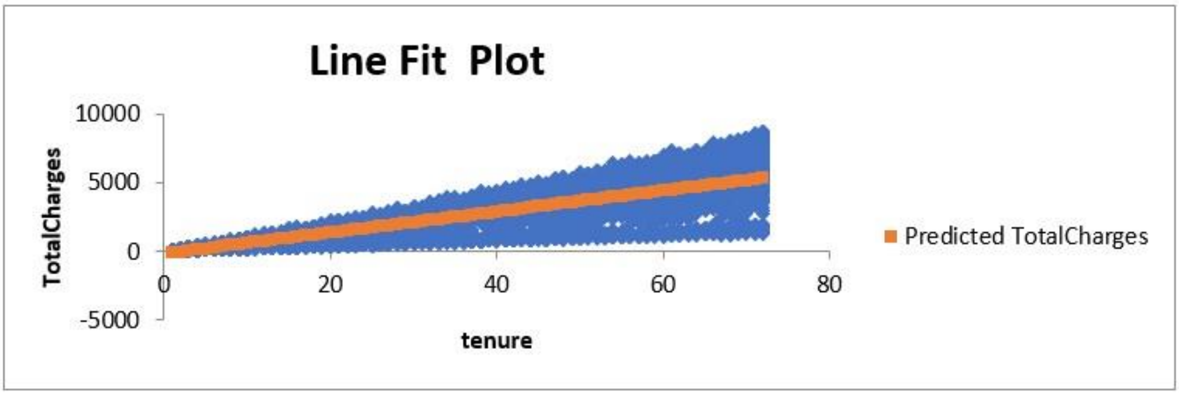
Table 1 - Metadata analysis

In general, the data is easy to understand. The ratio of female and male customers is almost 1:1. There are only 11 values missing in the dataset which will be explained in the next chapter. The only metadata which is not specified in the dataset is the currency. To match with the business case scenario, we assumed that the currency is euro.

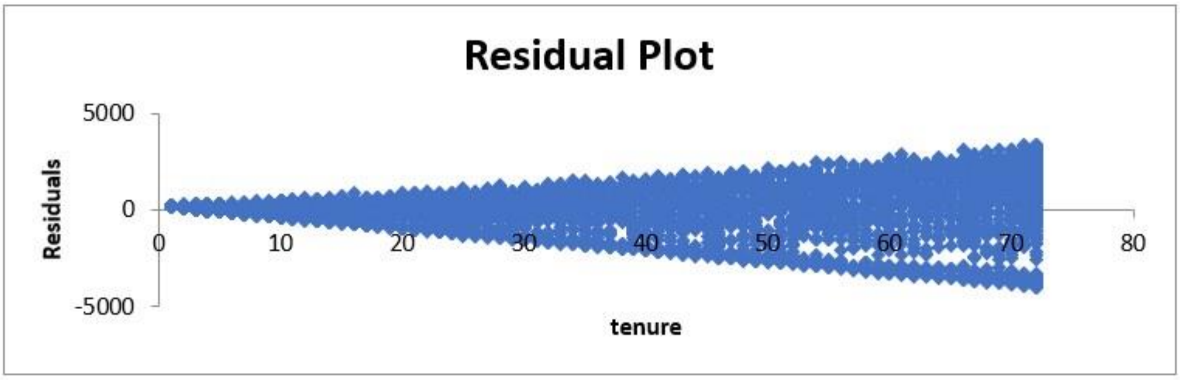
5 Solution development

5.1 Data cleansing and preparation

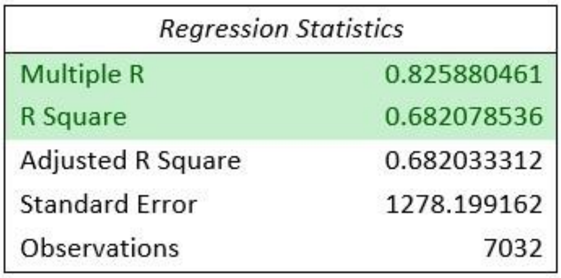
The data cleansing process was divided into two main parts. First, we checked if there was any missing value in our dataset. We immediately identified 11 missing values in the “Total Charges” column. Considering the size of the data set (more than 7000 rows) we thought we could simply remove them: the quality of our models would not have been affected in any way. Secondly, we checked if some variables were correlated with other ones: multicollinearity can be a serious problem, especially in Logistic Regression. By logic, we identified three variables that could have been correlated: “Tenure”, “Total charges” and “Monthly charges”. We decided to use the “data analysis” add-in of Microsoft Excel to analyse their likely correlation. First of all, we ran Linear Regression algorithm with “Tenure” as the independent variable and “Total charges” as the dependent one. The results confirmed our concerns:



Graph 1 - Linear regression



Graph 2 - Residual plot

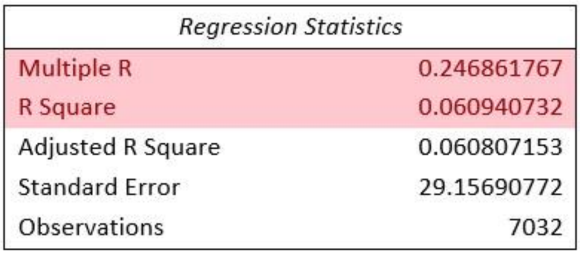


Regression Statistics	
Multiple R	0.825880461
R Square	0.682078536
Adjusted R Square	0.682033312
Standard Error	1278.199162
Observations	7032

Table 2- Regression statistics 3

Even though heteroskedasticity is evident, correlation (0.826) and R square (almost 0.7) are too high to be neglected. We decided to exclude "Total charges" from our analysis, since the future result would have been more easily interpretable.

After that, we ran Linear Regression algorithm with "Tenure" as the independent variable and "Monthly charges" as the dependent one. We decided to analyse these two variables because Telecommunication companies usually take advantage of customers who stick with the same contract for too long, hence they keep increasing their monthly charges. In this case, the result proved we were wrong.

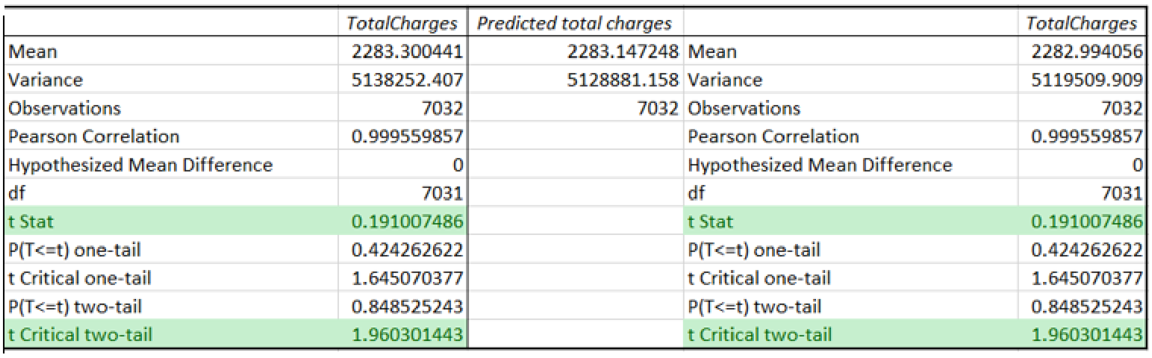


Regression Statistics	
Multiple R	0.246861767
R Square	0.060940732
Adjusted R Square	0.060807153
Standard Error	29.15690772
Observations	7032

Table 3 - Regression statistics 2

Correlation between these two variables is only 0.247 and R squared is incredibly low (only 0.06). For this reason, we decided to keep both variables in our dataset.

As a third step, we wanted to verify if there was a perfect correlation between "Total Charges" and the product between "Monthly Charges" and "Tenure". The purpose was to calculate the percentage of customers that modified their contract. We ran a Paired Sample T-test between the actual total charges and the hypothetical ones ("Monthly Charges" * "Tenure"). These are the results we obtained:



	TotalCharges	*Predicted total charges*		*TotalCharges*
Mean	2283.300441	2283.147248	Mean	2282.994056
Variance	5138252.407	5128881.158	Variance	5119509.909
Observations	7032	7032	Observations	7032
Pearson Correlation	0.999559857		Pearson Correlation	0.999559857
Hypothesized Mean Difference	0		Hypothesized Mean Difference	0
df	7031		df	7031
t Stat	0.191007486		t Stat	0.191007486
P(T<=t) one-tail	0.424262622		P(T<=t) one-tail	0.424262622
t Critical one-tail	1.645070377		t Critical one-tail	1.645070377
P(T<=t) two-tail	0.848525243		P(T<=t) two-tail	0.848525243
t Critical two-tail	1.960301443		t Critical two-tail	1.960301443

Table 4 - Paired samples t-test

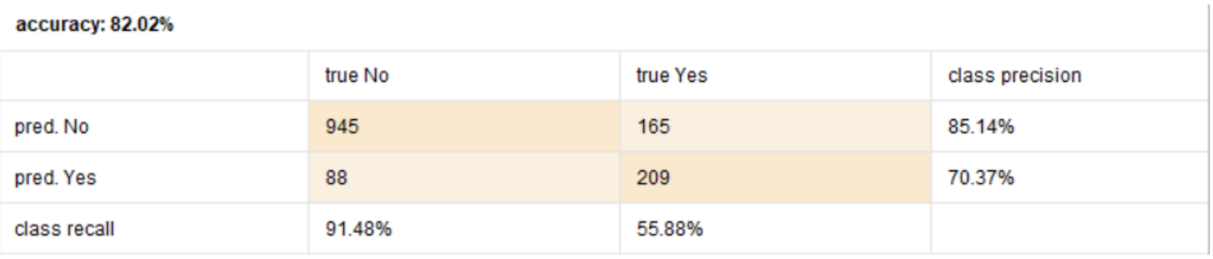
It is clear that these two values are basically the same (P-value is well above 0.05 and correlation is almost 1). We can claim only few customers decided to modify the contract.

5.2 Cross-validation

Once the dataset was cleaned, we could finally start our research of the best model for churn prediction. For each algorithm we executed cross-validation for better reliability of the confusion matrix. To do so, we used the “Split Data” operator, setting the following ratio: 0.8 for the training dataset, 0.2 for the testing one.

5.3 Logistic regression

Before running this algorithm, we had to convert all the binominal and polynominal attributes into numerical values. To do so, we used the “Nominal to Numerical” operator. This is the result we obtained:

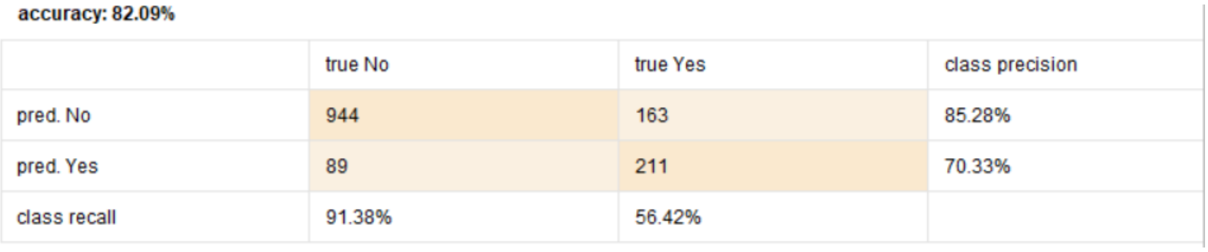


accuracy: 82.02%

	true No	true Yes	class precision
pred. No	945	165	85.14%
pred. Yes	88	209	70.37%
class recall	91.48%	55.88%	

Table 5 - Logistic regression; confusion matrix 1

Even though the general accuracy was definitely high, we were a bit disappointed of the ratio of “true Yes” that were actually predicted as actual Yes. Our purpose was in fact to create the best possible model to predict customers that could likely churn. In order to improve the model, we implemented two solutions. First, we removed from our dataset all the attributes that were not helpful in predicting customer churn: we examined all the standardized coefficients for each attribute and its p-value. In the end, we removed “Dependents”, “Device Protection”, “gender”, “Online Backup”, “Partner” and “streaming TV”, since their p-values were all above 0.05. This is the confusion matrix we obtained after having implemented these changes:

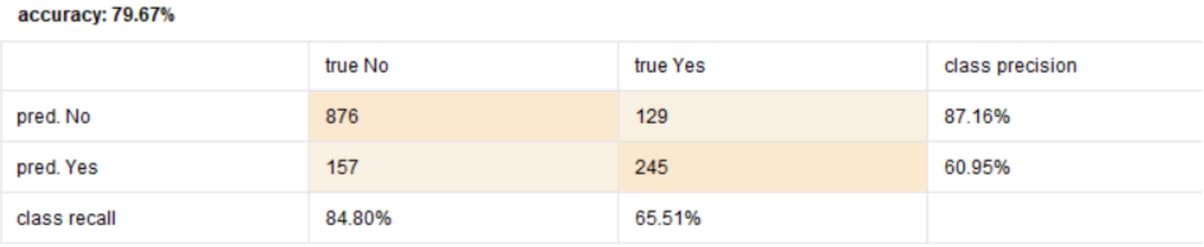


accuracy: 82.09%

	true No	true Yes	class precision
pred. No	944	163	85.28%
pred. Yes	89	211	70.33%
class recall	91.38%	56.42%	

Table 6 - Logistic regression; confusion matrix 2

Unfortunately, the accuracy was only slightly improved. As a second solution, we decided to sacrifice the overall accuracy in favour of the accuracy of predicting customer churn. To do so, we used the “Change Threshold” operator to change the classification threshold from 0.5 to 0.6:



accuracy: 79.67%

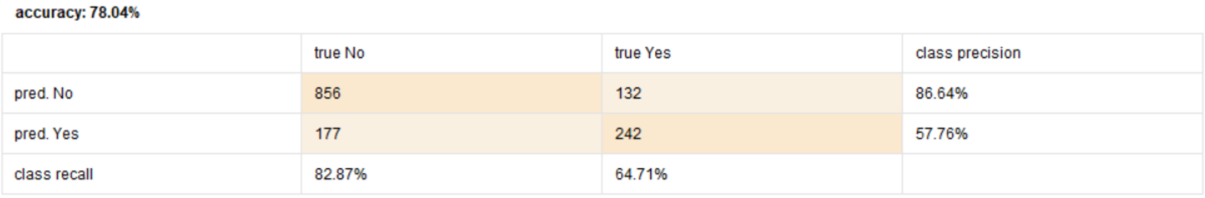
	true No	true Yes	class precision
pred. No	876	129	87.16%
pred. Yes	157	245	60.95%
class recall	84.80%	65.51%	

Table 7 - Logistic regression; confusion matrix 3

This model is evidently more precise in predicting actual customer churn (65.51%) and the overall accuracy was not affected that much, since it had decreased only by 2.42%.

Even though the accuracy of this model was very high, the regression results were not easily interpretable. The main reason was the presence of polynominal variables converted to numerical: it is impossible to interpret their standardized coefficient and therefore their impact on the model. This interpretation is easy when you deal with native numerical variables or binominal ones. Our new purpose was to modify the dataset to find the best interpretable model.

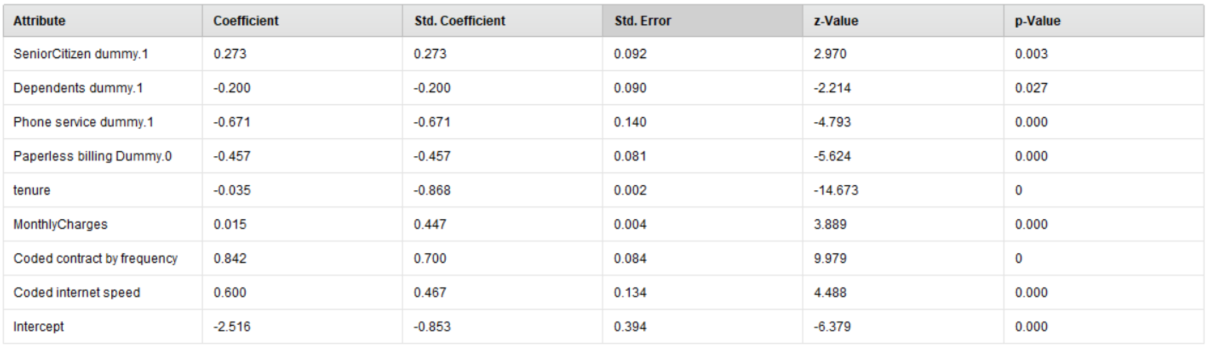
First of all, we coded all the binominal variables into dummies (0 and 1), then we had to identify all the polynominal variables that were also ordinal. Ordinal variables can in fact be interpreted after turning them into numerical. We immediately spotted two of those: “Contract” and “Internet speed”. The variable “Contract” could be coded in terms of the increasing frequency of payment. In short, “Two years”, “One year” and Month-by-month” were coded to “0”, “1” and “2” respectively. We followed the same logic with the “Internet speed” attribute, by coding it in terms of the speed of the internet connection. Therefore, we coded “No”, “DSL” and “Fiber optic” to “0”, “1” and “2” respectively. Unfortunately, we had to ignore all the polynominal variables that could not be meaningfully coded into numerical ones. We had to do it for the sake of model’s interpretability. Once we ran the model, we removed the variables with a p-value below 0.05 for the second time. After having run Logistic Regression, this is what we obtained:



accuracy: 78.04%

	true No	true Yes	class precision
pred. No	856	132	86.64%
pred. Yes	177	242	57.76%
class recall	82.87%	64.71%	

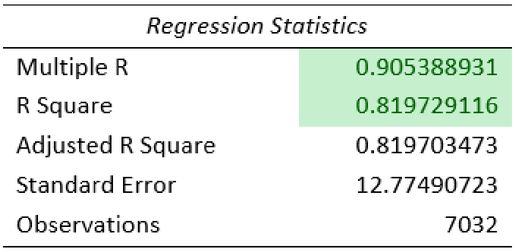
Table 8 - Logistic regression; confusion matrix 4



Attribute	Coefficient	Std. Coefficient	Std. Error	z-Value	p-Value
SeniorCitizen dummy.1	0.273	0.273	0.092	2.970	0.003
Dependents dummy.1	-0.200	-0.200	0.090	-2.214	0.027
Phone service dummy.1	-0.671	-0.671	0.140	-4.793	0.000
Paperless billing Dummy.0	-0.457	-0.457	0.081	-5.624	0.000
tenure	-0.035	-0.868	0.002	-14.673	0
MonthlyCharges	0.015	0.447	0.004	3.889	0.000
Coded contract by frequency	0.842	0.700	0.084	9.979	0
Coded internet speed	0.600	0.467	0.134	4.488	0.000
Intercept	-2.516	-0.853	0.394	-6.379	0.000

Table 9 - Logistic regression; attributes 1

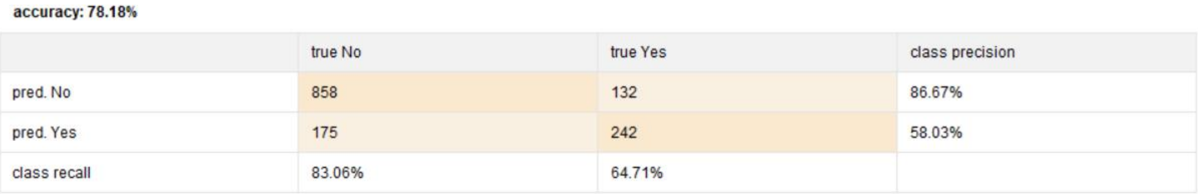
The quality of the model was satisfying, and all the attributes were easy to interpret. However, there was a problem that immediately caught our attention: the coefficient related to internet speed was positive. This did not make any sense, since it would have meant that the faster the internet connection was, the more likely the customer would churn. For this reason, we wandered if there was any other problem of multicollinearity. We thought that the attribute that was more likely to be correlated with internet speed was "Monthly charges", so we ran another linear regression with them. The results confirmed we were right:



Regression Statistics	
Multiple R	0.905388931
R Square	0.819729116
Adjusted R Square	0.819703473
Standard Error	12.77490723
Observations	7032

Table 10 - Logistic regression; statistics

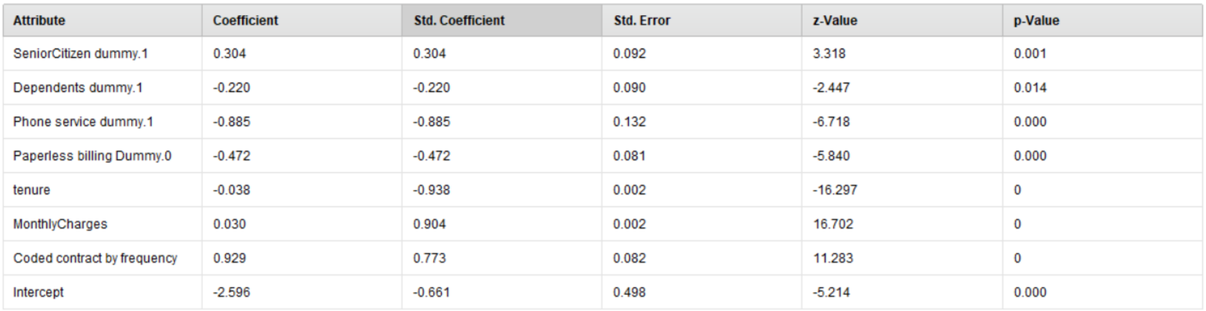
Correlation was incredibly high, so we decided to remove "Internet speed" from our model. Finally, we ran Logistic Regression algorithm for the very last time:



accuracy: 78.18%

	true No	true Yes	class precision
pred. No	858	132	86.67%
pred. Yes	175	242	58.03%
class recall	83.06%	64.71%	

Table 11 - Logistic regression; confusion matrix 5



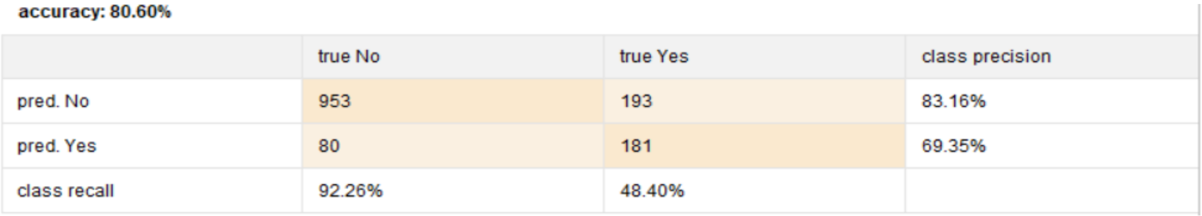
Attribute	Coefficient	Std. Coefficient	Std. Error	z-Value	p-Value
SeniorCitizen dummy.1	0.304	0.304	0.092	3.318	0.001
Dependents dummy.1	-0.220	-0.220	0.090	-2.447	0.014
Phone service dummy.1	-0.885	-0.885	0.132	-6.718	0.000
Paperless billing Dummy.0	-0.472	-0.472	0.081	-5.840	0.000
tenure	-0.038	-0.938	0.002	-16.297	0
MonthlyCharges	0.030	0.904	0.002	16.702	0
Coded contract by frequency	0.929	0.773	0.082	11.283	0
Intercept	-2.596	-0.661	0.498	-5.214	0.000

Table 12 - Logistic regression; attributes 2

We eventually obtained a model that was both quite accurate and easy to interpret. The standardized coefficient of “Monthly charges” had boosted from 0.447 to 0.904, and that proves that multicollinearity negatively affected our previous model.

5.4 k-NN

We followed the same steps also for k-nearest neighbour algorithm, including the comparison with an increased threshold (from 0.5 to 0.6). We chose 25 as the number of k since it is not neither too big nor too small, and it avoids ties (that might occur with an even number of k). This is the result we obtained without changing the threshold and weighted vote:

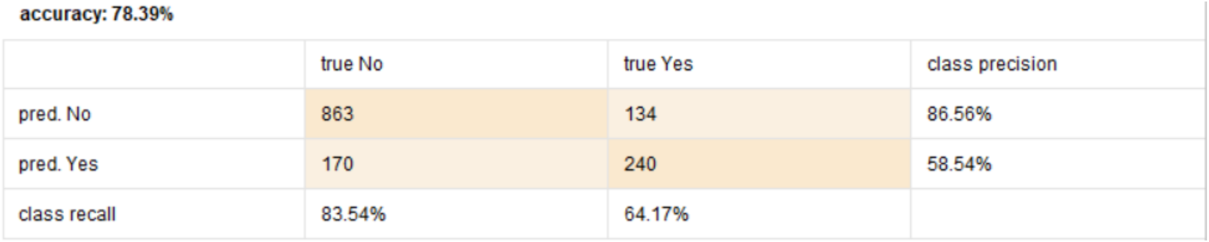


accuracy: 80.60%

	true No	true Yes	class precision
pred. No	953	193	83.16%
pred. Yes	80	181	69.35%
class recall	92.26%	48.40%	

Table 13 - k-NN; confusion matrix 1

Even in this case we had to re-balance the confusion matrix by increasing the threshold to 0.6:



accuracy: 78.39%

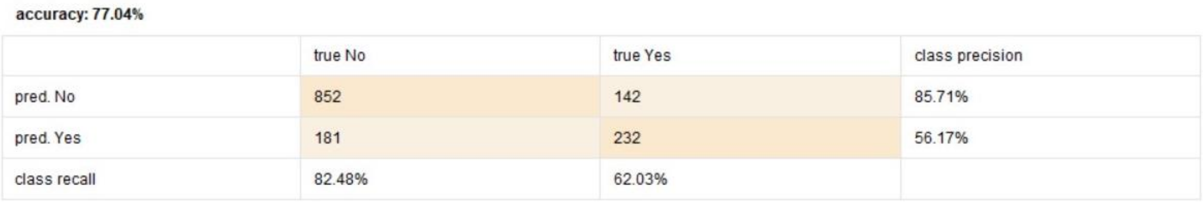
	true No	true Yes	class precision
pred. No	863	134	86.56%
pred. Yes	170	240	58.54%
class recall	83.54%	64.17%	

Table 14 - k-NN; confusion matrix 2

Overall, k-NN works well in this classification problem. However, we preferred Logistic Regression, since the accuracy is similar, but the model is much easier to interpret.

5.5 Decision tree

After importing the cleaned data set (total charge excluded), setting roles (customer id as id and churn as label) and splitting data according to 80/20 ratio, we have first run the decision tree according to the default setting of parameters, the root was the contracts.

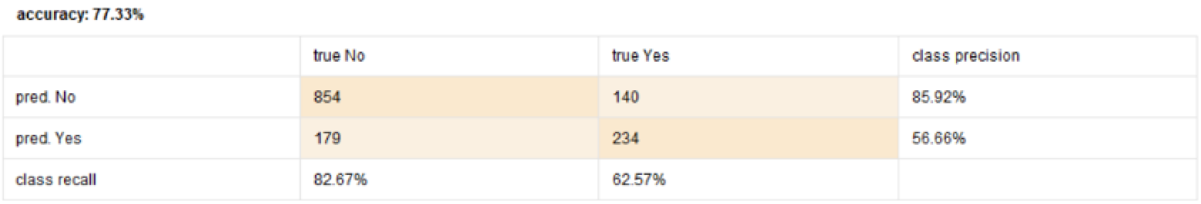


accuracy: 77.04%

	true No	true Yes	class precision
pred. No	852	142	85.71%
pred. Yes	181	232	56.17%
class recall	82.48%	62.03%	

Table 15 - Decision tree; confusion matrix 1

Then we run the model using cleaned data set according to logistics regression results where certain attributes were removed based on examination of standardized coefficients and p-values. We run the model again with the default parameters setting.



accuracy: 77.33%

	true No	true Yes	class precision
pred. No	854	140	85.92%
pred. Yes	179	234	56.66%
class recall	82.67%	62.57%	

Table 16 - Decision tree; confusion matrix 2

Dataset with removed attributes performed slightly better, so we decided to use this set for further applications. Contract attribute was again assigned as a root.

Then, we applied parameters suggested by RapidMiner according to the most frequent usage. 67% chose a value between 20 and 29 for maximal depth, we chose 25. 76% chose a value between 0.2 and 0.4 for confidence, we chose 0.3, and for minimal gain 70% chose between 0.1 and 0.2. For our data set even 0.1 was too high; a higher value of minimal gain results in fewer splits and thus a smaller tree. A value that is too high will completely prevent splitting and a tree with a single node is generated which is what happened in our case, when we applied 0.1. When leaving the last option on default and only apply the first two suggested, the results in confusion matrix were worse than with the default option and also a tree with a maximal depth of 25 is much more difficult to interpret and interpretability is also what we are looking for.

Another solution for the case when there is only one node in the tree (it is very likely that the standard pruning options are preventing the tree from growing), is a very drastic one of deactivating pruning and prepruning. When removing both parameters and leaving the maximal depth on 25, the results were worse than with our best default model so far. When we changed the depth to 10, as the default setting, the result for true yes turned out 0.27%.

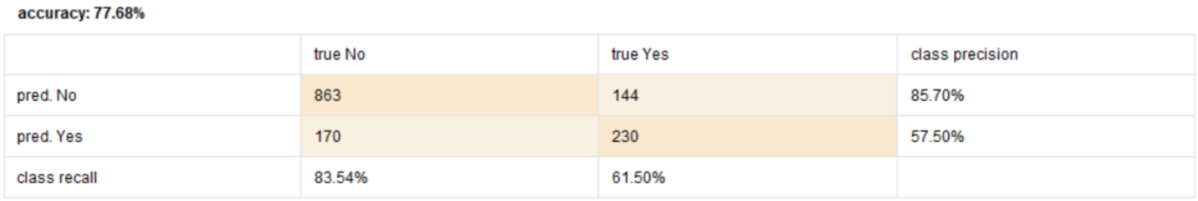
When we applied only prepruning, the results were the same as with pruning and prepruning applied simultaneously. We decided to apply it and leave it on default setting. Pruning is a way to cut away leaves which are not statistically meaningful, after the tree was built. Prepruning prevents that such leaves are being built at all.

According to the RapidMiner community, the two most important settings within pruning and prepruning are minimal gain and confidence.

Minimal gain specifies how good a cut needs to be that it is really executed. A 0 means all cuts are made while a 1 means only cuts which purify the two leaves are executed. The standard setting of 0.1 is considered as a hard requirement, which our data set confirmed as it resulted with a single node tree.

Confidence is a statistical measure based on binominal distribution which branches should be pruned away after building the tree.

After running the data with several different combinations of pruning and prepruning parameters (maximal depth always leaving on 10), the best results of accuracy happened when we set confidence to 0.4 and minimal gain to 0.04.



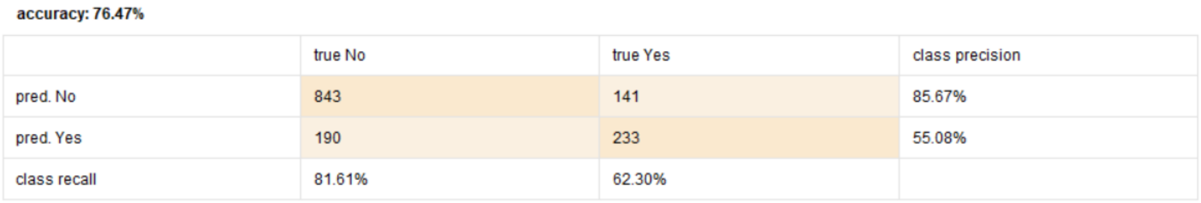
accuracy: 77.68%

	true No	true Yes	class precision
pred. No	863	144	85.70%
pred. Yes	170	230	57.50%
class recall	83.54%	61.50%	

Table 17 - Decision tree; confusion matrix 3

We are looking for a model that can be best interpretable in order to provide best possible assistance to the client. When setting depth to 10, the interpretation is not impossible, but it is also not very smooth and might not be very understandable or helpful to our final customer.

Luckily, when we changed the maximal depth to 6, the model became easily interpretable and what is more important, the accuracy did not worsen significantly. Here are the accuracy results for the model we find best for interpretation (confidence: 0.4, minimal gain: 0.04):



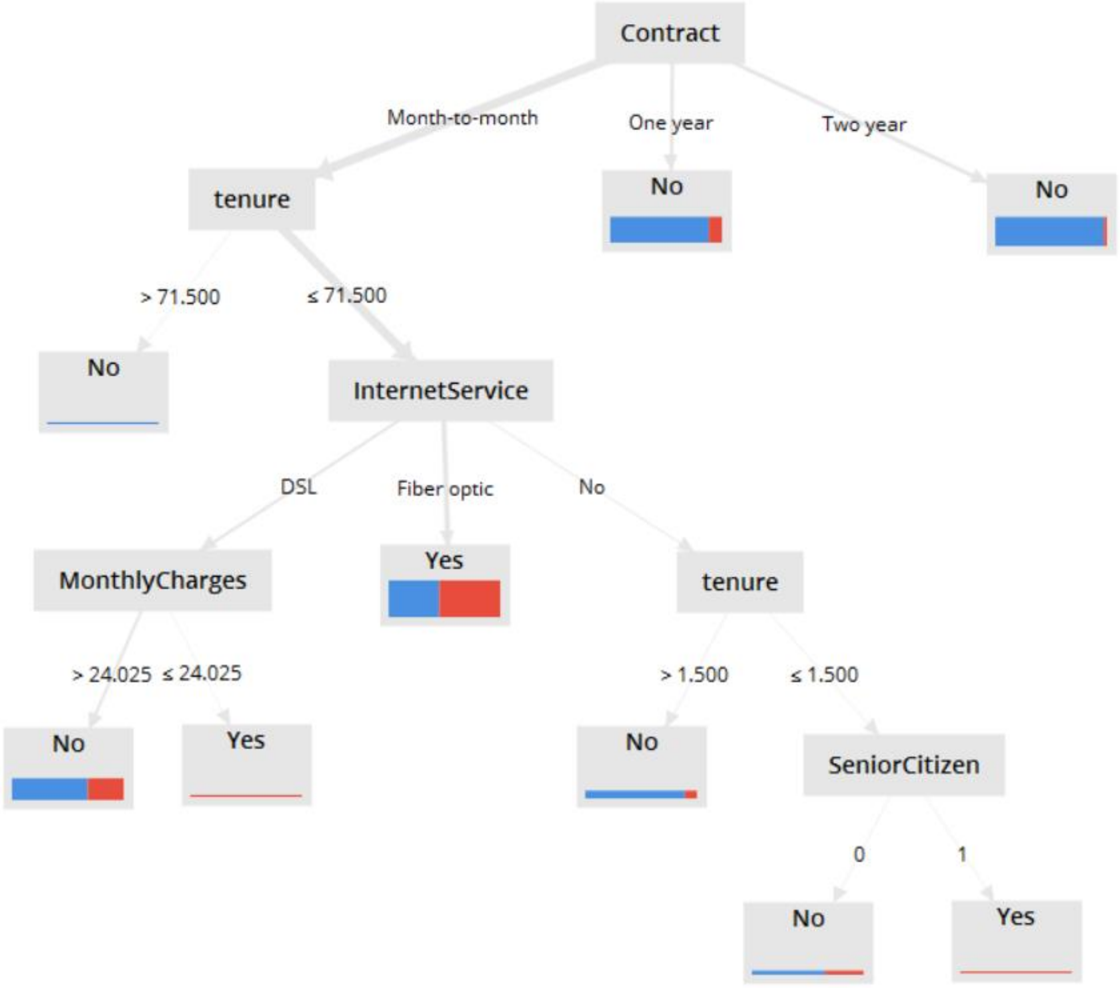
accuracy: 76.47%

	true No	true Yes	class precision
pred. No	843	141	85.67%
pred. Yes	190	233	55.08%
class recall	81.61%	62.30%	

Table 18 - Decision tree; confusion matrix 4

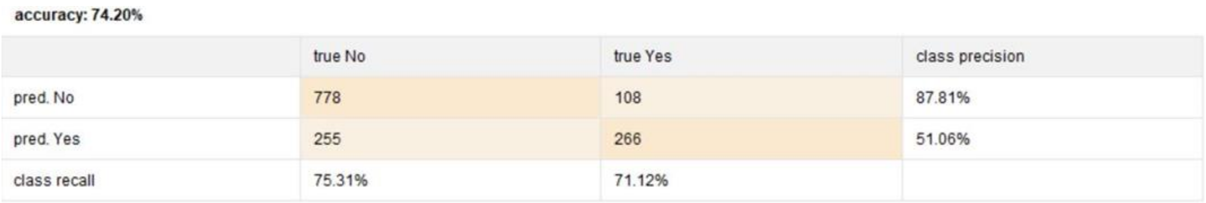
On the graphical model, selected attributes serve as best predictors for our label attribute. Contract is the root, the best predictor of whether or not a customer is going to churn.

With all the combinations so far, precision and recall are higher for “no”. According to our selected tree, customers, who have two year- or one year-contract are very likely not to churn. Those, who have monthly contract and are with the company for less than 71,5 months and have chosen fiber optic as their internet service are the only group showing a possible alarm for churning, as results here are showing 930 yes for churning out of 1704 total.



Graph 3 - Decision tree; graph 1

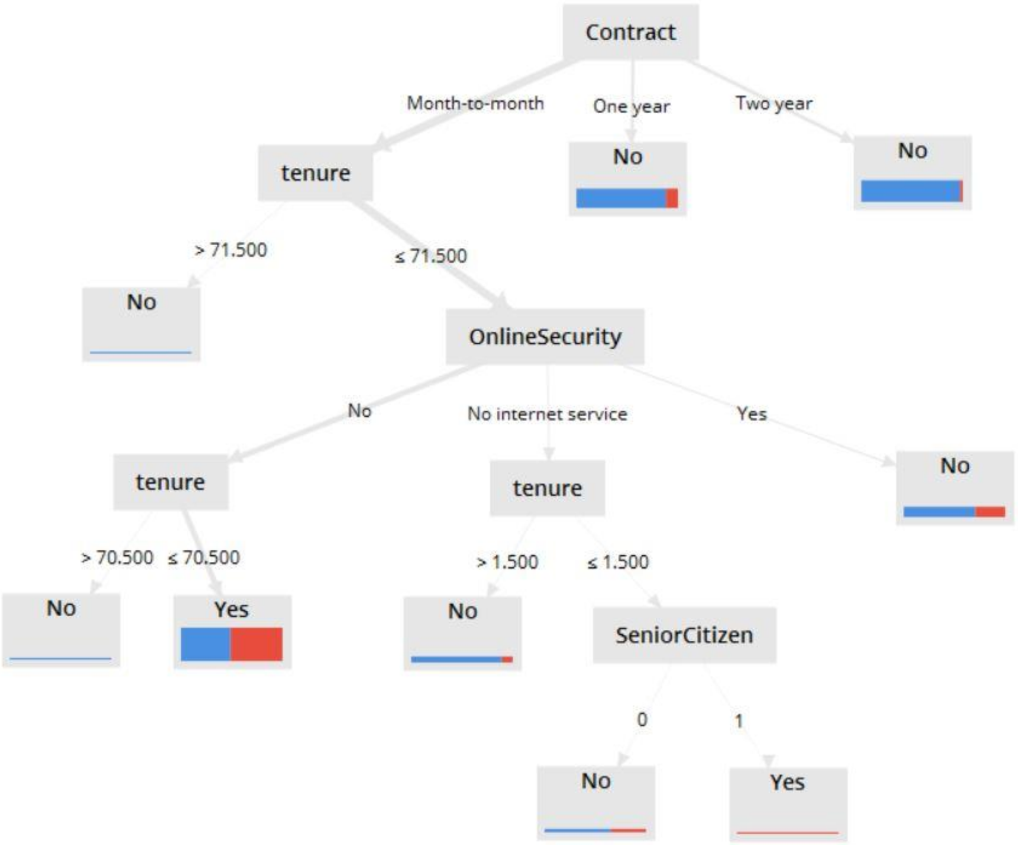
According to the results, the faster the internet service is, more likely the customer will churn. We found these results questionable and we also discovered high correlation between monthly charge and internet service with logistic regression before. Consequently, because of high multicollinearity, we decided to exclude the internet service attribute and run the decision tree again. The accuracy turned out to be 2,27% lower than our previous model.



accuracy: 74.20%

	true No	true Yes	class precision
pred. No	778	108	87.81%
pred. Yes	255	266	51.06%
class recall	75.31%	71.12%	

Table 19 - Decision tree; confusion matrix 5

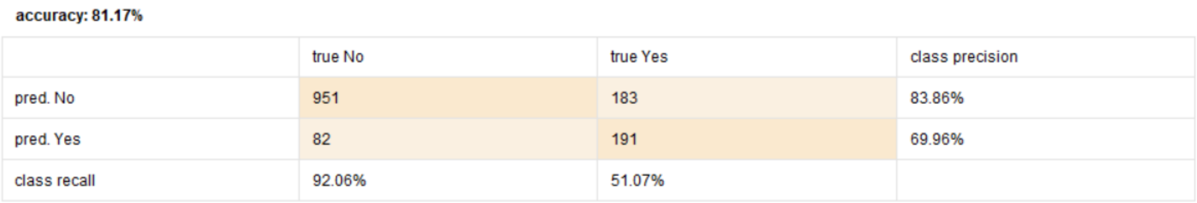


Graph 4 - Decision tree; graph 2

The likeliness for not churning stayed the same, the result for which customers are likely to churn changed, since we removed the internet service attribute. According to the last graph, customers most likely to churn are the ones who have monthly contract, did not subscribe to online security and are with the company for less than 70,5 months. Out of 2107 customers, 1078 will churn, slightly above 50%.

5.6 Random forest

First, we run this algorithm with default setting with clean data set (total charge excluded).

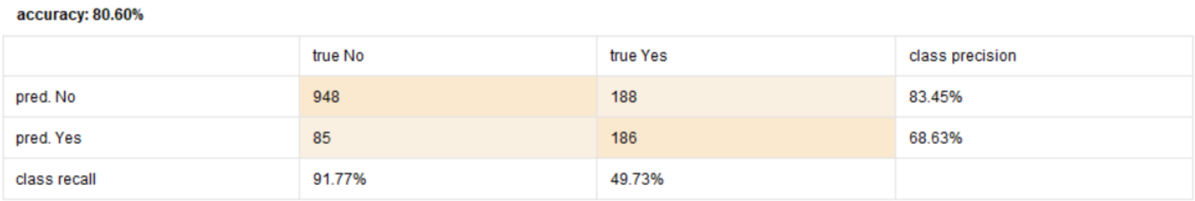


accuracy: 81.17%

	true No	true Yes	class precision
pred. No	951	183	83.86%
pred. Yes	82	191	69.96%
class recall	92.06%	51.07%	

Table 20 - Random forest; confusion matrix 1

Then, we run it with default setting with data set without attributes (removed based on logistics regression).



accuracy: 80.60%

	true No	true Yes	class precision
pred. No	948	188	83.45%
pred. Yes	85	186	68.63%
class recall	91.77%	49.73%	

Table 21 - Random forest; confusion matrix 2

In this case, the data set with all the attributes performed better. The overall accuracy is higher for 3,49% compared with the best accuracy of the decision tree algorithm.

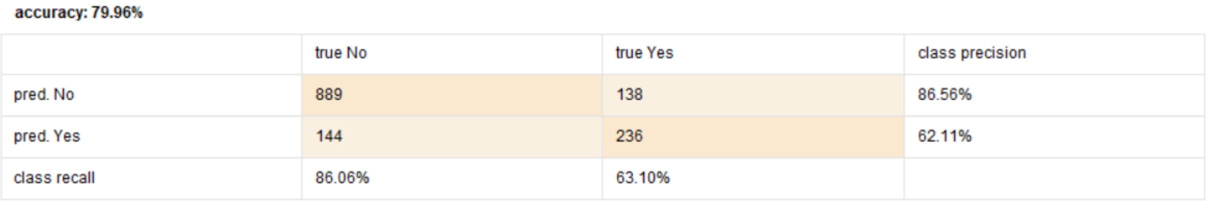
Random forest is a classifier that evolves from decision trees. It creates several random trees on different example subsets. The resulting model is based on voting of all these trees. Due to this difference, it is less prone to overtraining. To classify a new instance, each decision tree provides a classification for input data; random forest collects the classifications and chooses the most voted prediction as the result. The input of each tree is sampled data from the original dataset. In addition, a subset of features is randomly selected from the optional features to grow the tree at each node. Each tree is grown without pruning. Essentially, random forest enables many weak or weakly-correlated classifiers to form a strong classifier.

Unfortunately, it is very difficult to interpret since the default creates 100 different decision trees from which we cannot extract one answer for churning prediction. Although the accuracy is higher, for the sake of the interpretation of the problem, it is wiser to select decision tree algorithm.

5.7 Neural net

The Neural Net operator in RapidMiner cannot handle nominal attributes, so we needed to code the nominal values to numerical values. In RapidMiner there is the “Nominal to Numerical” operator which can do so. By choosing coding type “unique integer”, all the nominal attributes are coded as “0”, “1”, “2”, etc. Because for algorithms based on artificial neural networks, interpretability is not priority, we used automatically coding operator offered by the software and included all the attributes.

At first, we ran the algorithm with all default parameter setting and the performance vector was returned as shown below. In general, the accuracy was satisfying (80%) but we would like to try setting different parameters and see if the confusion matrix can be improved especially for class recall and class precision for predicting “Yes”.

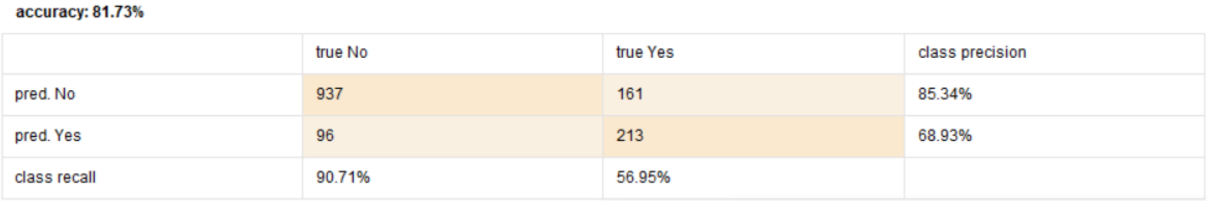


accuracy: 79.96%

	true No	true Yes	class precision
pred. No	889	138	86.56%
pred. Yes	144	236	62.11%
class recall	86.06%	63.10%	

Table 22 - Neural net; confusion matrix 1

So, in the second try we turned on the “decay” parameter which means that the learning rate will decrease during learning. In the result we can see the accuracy increased to almost 82% and the precision of predicting “Yes” increased to 69% but the recall of it decreased to 57%.



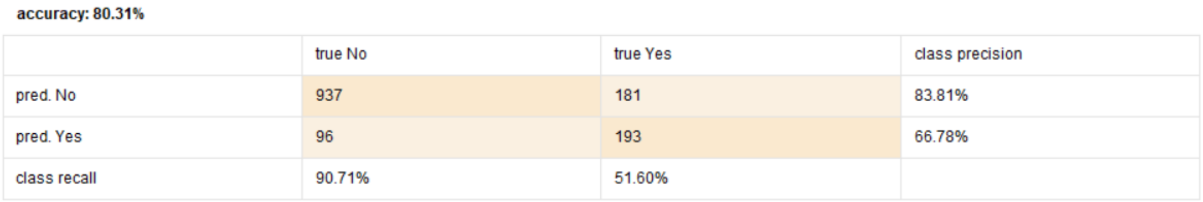
accuracy: 81.73%

	true No	true Yes	class precision
pred. No	937	161	85.34%
pred. Yes	96	213	68.93%
class recall	90.71%	56.95%	

Table 23 - Neural net; confusion matrix 2

Then we tried to set the “learning rate” and “momentum” parameter to different values based on RapidMiner’s “crowd wisdom” function which shows you what values are the most often used by all RapidMiner users. But the results of different settings barely changed and sometimes it even got worse regarding the balance of recall and precision. The default “learning rate” is 0.01 and “momentum” is 0.9, no matter we increased learning rate or decreased momentum, usually the precision of true “Yes” would increase but the recall would drop dramatically, thus the accuracy kept the same or even lower. So, eventually we chose to keep the first model with default parameters.

After having obtained the most optimized parameters, we wanted to test whether the confusion matrix will improve if we only include the attributes with high significance based on the results of Logistic Regression. The overall accuracy did not change, the class precision of “Yes” increased slightly (67%) but the class recall dropped (52%).



accuracy: 80.31%

	true No	true Yes	class precision
pred. No	937	181	83.81%
pred. Yes	96	193	66.78%
class recall	90.71%	51.60%	

Table 24 - Neural net; confusion matrix 3

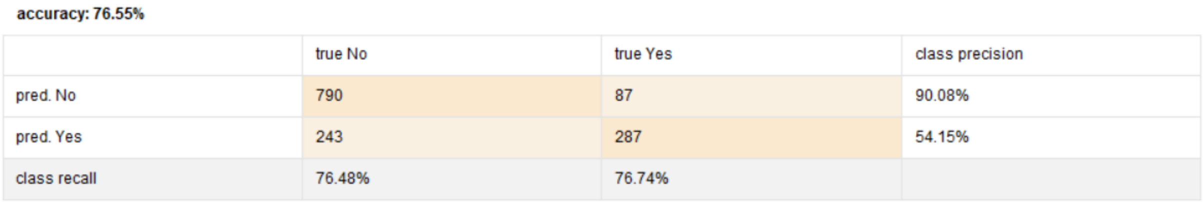
So, the output of Neural Net model did not improve in this case and the first model is still the best one so far.

Overall, Neural Net created a decent churn prediction model with high accuracy and balanced class prediction and recall. But we also need to point out that the interpretability of Neural Net is quite poor.

5.8 Deep learning

Deep Learning is another algorithm based on artificial neural networks. The advantage of Deep Learning comparing to Neural Net is that it can handle nominal attributes since in our datasets a lot of attributes are polynominal. Another advantage is that Deep Learning allows to compute variable importances which makes Deep Learning slightly more interpretable than Neural Net.

With default activation function "rectifier" and parameters, we got a model with surprisingly high class recall of "Yes" (76.7%) but lower precision and overall accuracy. We tried all four activation functions, and they all returned similar matrix. According to RapidMiner recommendation and Bank's churn case by Borges (2018), we decided to keep using the activation function "rectifier".

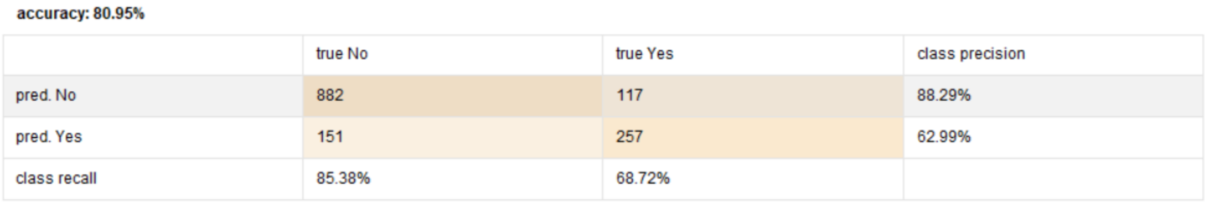


accuracy: 76.55%

	true No	true Yes	class precision
pred. No	790	87	90.08%
pred. Yes	243	287	54.15%
class recall	76.48%	76.74%	

Table 25 - Deep learning; confusion matrix 1

After multiple times of trying different combination of parameters, we found out one model with the satisfying confusion matrix:



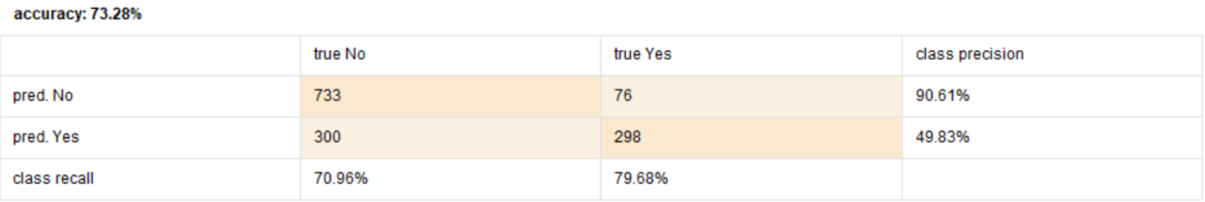
accuracy: 80.95%

	true No	true Yes	class precision
pred. No	882	117	88.29%
pred. Yes	151	257	62.99%
class recall	85.38%	68.72%	

Table 26 - Deep learning; confusion matrix 2

The accuracy is 81% with balanced class precision and recall of true “Yes”. Both are over 60% and the recall is almost 69%. The key parameters change comparing to default setting are: 2 hidden layers with sizes of 300 and 100; increasing epochs from 10 to 20 to iterate the dataset more times; using Huber loss function to decrease the effect of outliers.

Again, we applied “select attributes” to omit attributes with low significance and got this result:

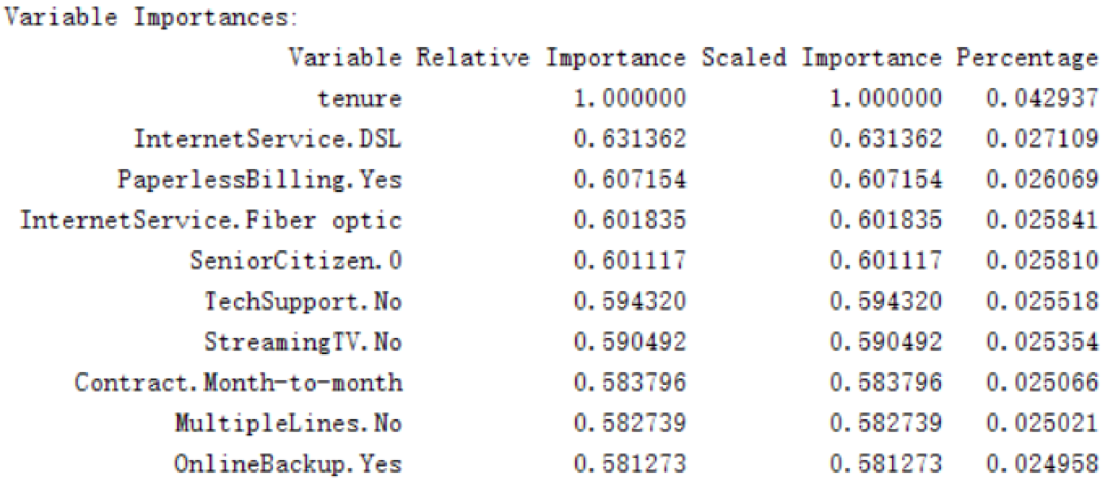


accuracy: 73.28%

	true No	true Yes	class precision
pred. No	733	76	90.61%
pred. Yes	300	298	49.83%
class recall	70.96%	79.68%	

Table 27 - Deep learning; confusion matrix 3

Although class recall of true Yes increased dramatically but accuracy dropped a lot as well as precision of true Yes which even went under 50%. So, we assumed that for artificial network, decreasing attributes might not be very helpful and in Deep Learning the best model is the second one. According to the computed variable importance, “Tenure”, “Internet service: DSL”, “Paperless billing: Yes”, “Internet service: Fiber Optic” and “non-senior citizen” contributed the most to the model. But because we could not tell these determining factors resulting in churning or not, it is still hard to interpret the result.

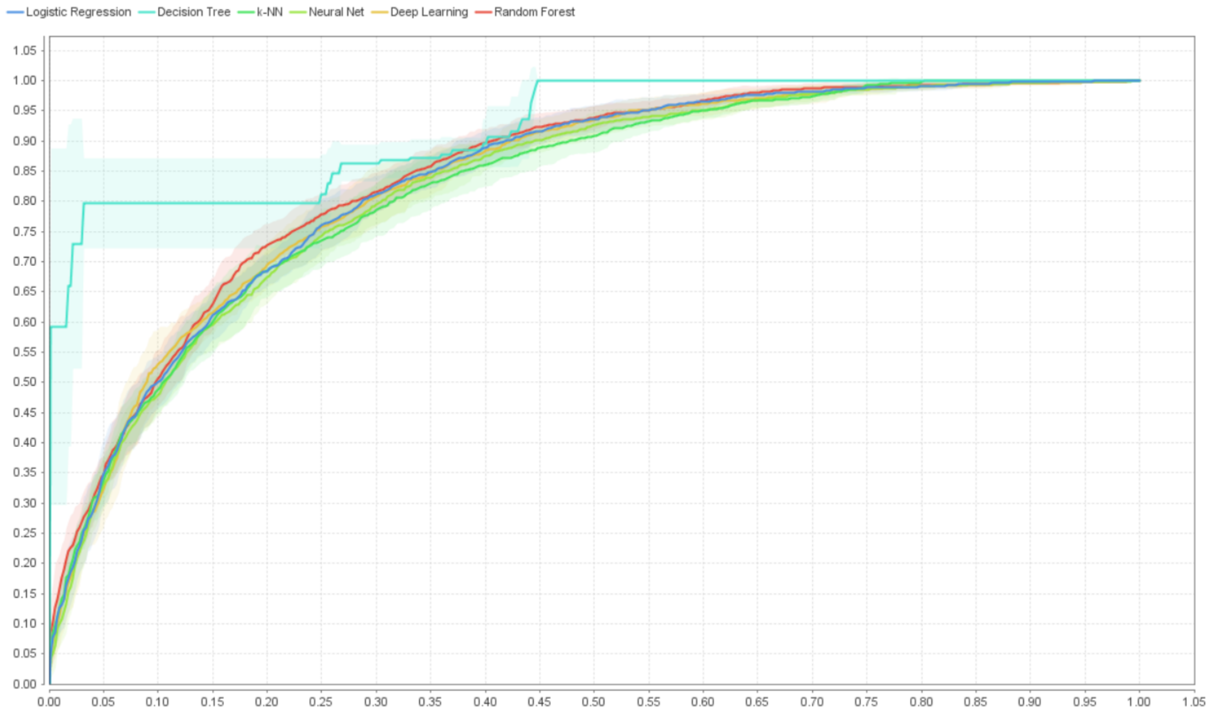


```
Variable Importances:
                   Variable Relative Importance Scaled Importance Percentage
                     tenure            1.000000          1.000000   0.042937
        InternetService.DSL            0.631362          0.631362   0.027109
       PaperlessBilling.Yes            0.607154          0.607154   0.026069
InternetService.Fiber optic            0.601835          0.601835   0.025841
            SeniorCitizen.0            0.601117          0.601117   0.025810
             TechSupport.No            0.594320          0.594320   0.025518
             StreamingTV.No            0.590492          0.590492   0.025354
    Contract.Month-to-month            0.583796          0.583796   0.025066
           MultipleLines.No            0.582739          0.582739   0.025021
           OnlineBackup.Yes            0.581273          0.581273   0.024958
```

Table 28 - Deep learning; output

5.9 ROC curves comparison

After selecting the best model (with preference in interpretability for logistic regression, k-NN, decision tree and preference in accuracy in random forest, neural net, deep learning) of each algorithm, we ran a ROC curves comparison for all of them. It returned out that, for classification in this case, Decision Tree had the best performance, the second best is Random Forest. Afterwards, Logistic Regression and Deep Learning had similar performance, then it is Neural Net. The worst one is k-NN.



Graph 5 - ROC curve result

6 Findings and suggestions for the business case

For our dataset, most of the algorithms ran fairly well in predicting non-churners, but less accurate in predicting churners, and it is hard to get a balance between recall and precision. Algorithms based on artificial neural networks returned confusion matrix with high accuracy and balanced recall and precision in both predicting churners and non-churners, but the interpretability was pretty low. For our business case, we thought that interpretability was equally important to us. Combining the results from ROCs comparison, confusion matrix and interpretability, eventually we decided that the best models for our case is logistic regression and decision tree. Logistic regression can show us which characteristics of customers are most related to churning, and decision tree can help us examine the pattern behind all the customers who churned.

The following suggestions will be based on the results from logistic regression and decision tree, because these two algorithms are the most interpretable.

First, in both logistic regression and decision tree, contract was determinant, resulting in customers to churn. In fact, its standardized coefficient was 0.773, and it also was the root of our decision tree. The customers who have shorter contract term are more likely to churn. There might be two reasons for this finding. One is that customers who have the monthly contracts could perceive that they are spending more money, the other one is that customers who have long term contract, need to consider the legal risk of breaking the contract, so it is less easy for them to churn. Therefore, our suggestion for the company’s future custom is to focus on trying to pursue their customers to choose yearly contract. The company could also launch the loyalty programme. For example, the customers could accumulate points by time or special events and then use the points to exchange for free services or products. But customers with longer contract term can accumulate the points faster and have more chance to get the points. This can be exploited by sales department to convince new customers, or to convince current customers to change their contract type to yearly or longer. This solution can also be justified for another finding we got from logistic regression which is also visible in decision tree. Customers whose tenure is less than 70, which means they stayed with the company less than 70 months, are more likely to churn.

Our second suggestion for the company is to consider lowering the price for their fastest internet service (fiber optic). Even though we found a high correlation between "Monthly Charges" and "Internet Service", the second-last decision tree showed that customers with fiber optic are more likely to churn. Initially we thought it was due to higher prices, but then we tried to find a deeper reason. Our hypothesis is leaning towards a thought that there is fierce competition between local telecommunication companies. It might happen that the competitors are able to offer lower price for the same internet speed. To solve this problem, the company should communicate with the technology and innovation department to find out a more cost-efficient way in offering internet service.

Thirdly, we found a negative coefficient of having paperless billing in logistic regression (-0.472), and this was also one of the variables which contributed the most to the deep learning model. We assumed that customers are more aware of digitalization now and value the convenience brought by it. Thus, in the future the company should invest in developing more convenient services to the customers.

Lastly, after applying the model to find out which customers are likely to churn, the marketing department should target these potential churners with promotions, such as bonus free services, more frequent advertisement, or discount on charges.

7 Authors' contribution

Giacomo: Data cleansing and preparation with Microsoft Excel, logistic regression and k-NN, t-test and linear regression, business case assessment

Eva: Introduction, requirements, decision tree, random forest

Qiao: Data and metadata analysis, neural net, deep learning, ROC curves comparison

Parts not mentioned have been created by the all three.

References

- Borges, D. M. (2018, October 03). Predicting Bank's churn with Artificial Neural Networks. Retrieved from https://medium.com/diogo-menezes-borges/predicting-banks-churn-with-artificial-neural-networks-f48393fb1f9c
- Orac, R. (2019, January 26). Churn prediction. Retrieved from https://towardsdatascience.com/churn-prediction-770d6cb582a5?gi=d18fd63f50e6
- Tsai, C. F., & Lu, Y. H. (2010). Data mining techniques in customer churn prediction. *Recent Patents on Computer Science, 3*(1), 28-32.