SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

Predicting customer churn in Telecom industry

AI & ML Individual Project

Student: Iurii Fedotov

Program: Master in Computer Science and Business Technology

TABLE OF CONTENTS:

2019 | Madrid | AI & ML

0. Important note

This report is intended to answer specific questions set up in the assignment description. Please, review the **fedotov_churn_notebook.ipynb** file too – the notebook contains more details about small technical things, like preprocessing or models choice.

1. Dataset

The dataset used for this assignment is stored in the file ***dataset.csv***[1]. It includes information about 7 thousand customers of anonymous telecom company, and provides the following details:

- A binary column indicating customers who left within the last month – the column is called Churn. It will be the target variable.
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Demographic info about customers – gender, age range, and if they have partners and dependents.

The initial quality of the dataset is very good – it has just few missing values which can be filled with meaningful values easily, using the values from another columns.

2. Python code

The full Python code used in this assignment is in the notebook ***fedotov_notebook.ipynb***. It uses only one dependency which is not installed in Anaconda's environment by default – CatBoost. To run the notebook, please, run **$ conda install catboost** in your anaconda environment. It also requires the latest versions of matplotlib and scikit learn, since it uses some functions which were released very recently.

The notebook contains self-explanatory comments on all important steps.

3. Description of the problem

Telecom markets in each country/region are typically divided between few large players, who are in tough competition with each other. The products are very standardized, but the transactional costs of switching from one provider to another are quite high for customers. To

[1] IBM Sample Data Sets: Customer Churn.
Link: https://www.kaggle.com/blastchar/telco-customer-churn#WA_Fn-UseC_-Telco-Customer-Churn.csv

automate the management of churn rate in large customer bases, telecom companies use so-called **churn models**.
The standard use case of this model is the following:

1. For each active customer (there are tens of millions of them) the model predicts whether this customer is likely to churn or not. Some classifiers predict it directly, while others, like logistic regression, predict the probability of churn $\{0 \leq P_{churn} \leq 1\}$ first, and then convert it to binary prediction.
2. A company can target customers which are likely to churn with special offers, thus attempting to change their decision to leave. It can use SMS mailing, direct calls, emails, and many other methods, each of which is not free.
3. On the other hand, a company can do less marketing efforts for the part of its client base which is not likely to churn.

Overall, the purpose of churn models is to allocate the marketing budget on customers retention effectively. It means investing money in keeping the customers who are likely to churn, and not putting much into those who will stay anyway.

Taking everything into consideration, telecom companies need models which predict customers churn based on their characteristics.

4. Value added by machine learning

In general, there are 3 ways to predict the likelihood of churn for clients:

- Manually investigate the client base, and use gut feeling and personal experience of sales managers to identify those who are likely to churn. This approach is practically impossible, because telecom companies have tens of millions of customers, they come and go daily, thus making manual tracking unfeasible.
- Predict a probability of churn for each client using a pre-defined mathematical formula. For example: “if a customer is on premium plan, then if, else if…”. This approach is at least feasible, since it automates the decision making. However, it does not take advantage of data that a company has, and is in the end equivalent to using gut feeling, since the formula is defined based on managers’ past experience.
- Constantly retrieve the relationships between churn and clients’ characteristics from the past, and use advanced statistical models to predict churn for current clients. This approach is not only feasible, but also, according to common position of industry experts,

demonstrates significantly better accuracy[2] than the second one. It reduces the human error, since past experience of the entire company is considered, and ML models are capable of catching relationships which humans will not consider on their own.

A good model is a model which advises to allocate money in retaining customers who are really thinking of changing a provider, and does not advice to put marketing efforts into those who plan to stay.

5. General approach and important decisions on each step

1. Have a brief look at the data, and deal with missing values – in this case, as I said before, it was possible to replace them with meaningful values using other columns. **Commented in the notebook.**
2. Analyze features individually, decide how to preprocess, aggregate, encode and even delete some of them. **All these decisions are commented in the notebook.**
3. Generate new features – **commented in the notebook.**
4. Create 5 alternative models, tune key hyperparameters of each of them, and compare the performance of optimized versions of 5 models. The objective metric for optimization was the **3-fold shuffled cross-validated precision**. Precision in general is the fraction of relevant instances among the retrieved instances. In this problem, it is a fraction of clients who did churn among those which were selected by the model as likely to churn. Precision reaches its best value at 1 and worst score at 0.

   I'm choosing precision for 2 reasons:

   - To provide business value for telecoms, it is better for this model to have lower recall but higher precision than vice versa. Predicting too much churn would mean losing money on ads which try to get people back while they don't plan to leave. On the other hand, we know that conversion rates of ads targeted to people who really want to churn are quite low, so we should spend them only for those who are really likely to churn.
   - We are facing a problem of imbalanced classes, and accuracy makes no sense in such problems.

5. Cross validation is very important to prevent overfitting during hyperparameter tuning, and making splits shuffled eliminates the risk of data being sorted. Finally, the number of

[2] By accuracy here I mean not a classification metric, but general performance of a group of algorithms.

folds is chosen to provide a good trade-off between computing time and validity of results. **All details of hyperparameters tuning are commented in the notebook.**

6. Make conclusions about which model was the best, and what outcomes might businesses expect while using it.

6. Critical appraisal of the results

The table below compares 5 models. It demonstrates scores for optimized version of each model, which are trained and tested on the entire data set. Please, note, that my approach assumed using cross validation during hyperparameters testing, to avoid parameters which lead to overfitting, and then comparing the models based on their performance of the entire data set.

Table 1. Comparison of 5 models

Model	Precision	Recall	F1	Accuracy	AUC
CatBoost Classifier	0.73	0.58	0.64	0.83	0.75
KNN classifier	0.70	0.38	0.50	0.79	0.66
AdaBoost Classifier	0.67	0.54	0.60	0.81	0.72
Logistic Regression	0.66	0.54	0.59	0.80	0.72
Decision Tree	0.43	0.89	0.58	0.65	0.73

According to our criteria – precision, the best model is **CatBoost Classifier**. It gives us not only the best precision, but the best values for accuracy, AUC and F1 score. The optimal l2_leaf_reg value is 100. 0.73 precision means that this model is the most conservative one – out of all churn predictions, only 23% are wrong.

Figure below provides details on CatBoost Classifier performance. Such figures for all other models are available in the notebook.