Expresso Churn Prediction

Introduction to DS Course Project

Vera Soboleva & Viacheslav Vasilev

25th of October, 2021

Skoltech
Skolkovo Institute of Science and Technology

Presentation Plan

1. Problem Statement
2. Data exploration
3. Data preprocessing
4. Feature engineering
5. Random Forest and Ensembling
6. Gradient Boosting
7. Feature importance
8. Conclusions

Problem statement

Expresso - African telecommunications company that provides customers with airtime and mobile data bundles.

Why churn prediction is important?

- better understanding of future expected revenue
- prevent churn
- understand what preventative steps are necessary
- formation of special offers for regular customers

Classification problem: X, y = CHURN : {0,1}

Evaluation metric: AUC

Data Exploration: Categorical features

The churn dataset 4 categorical variables:

- REGION - the location of each client
- TENURE - duration in the network
- TOP_PACK - the most active packs
- MRG - a client who is going

Data Exploration: Categorical features

The churn dataset 4 categorical variables:

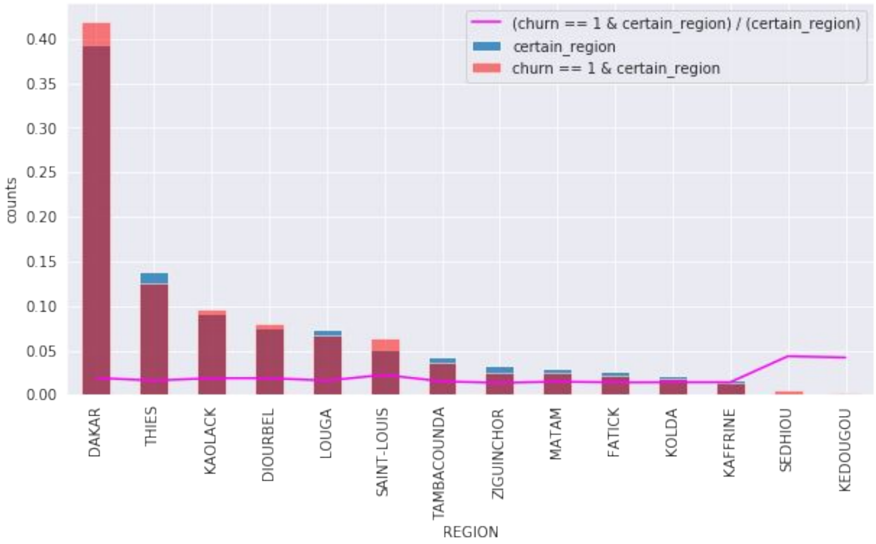
- REGION - the location of each client
- TENURE - duration in the network
- TOP_PACK - the most active packs
- MRG - a client who is going (all values are the same)

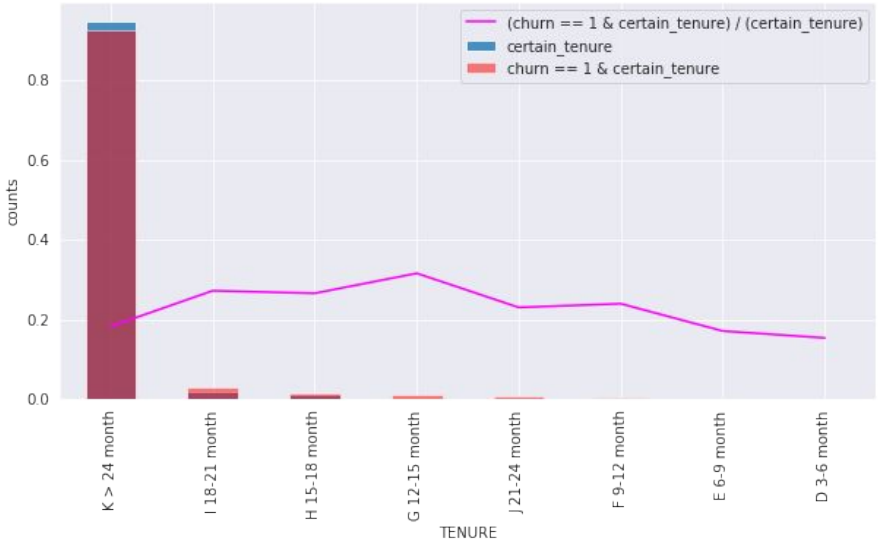
Data Exploration: Categorical features

The churn dataset 4 categorical variables:

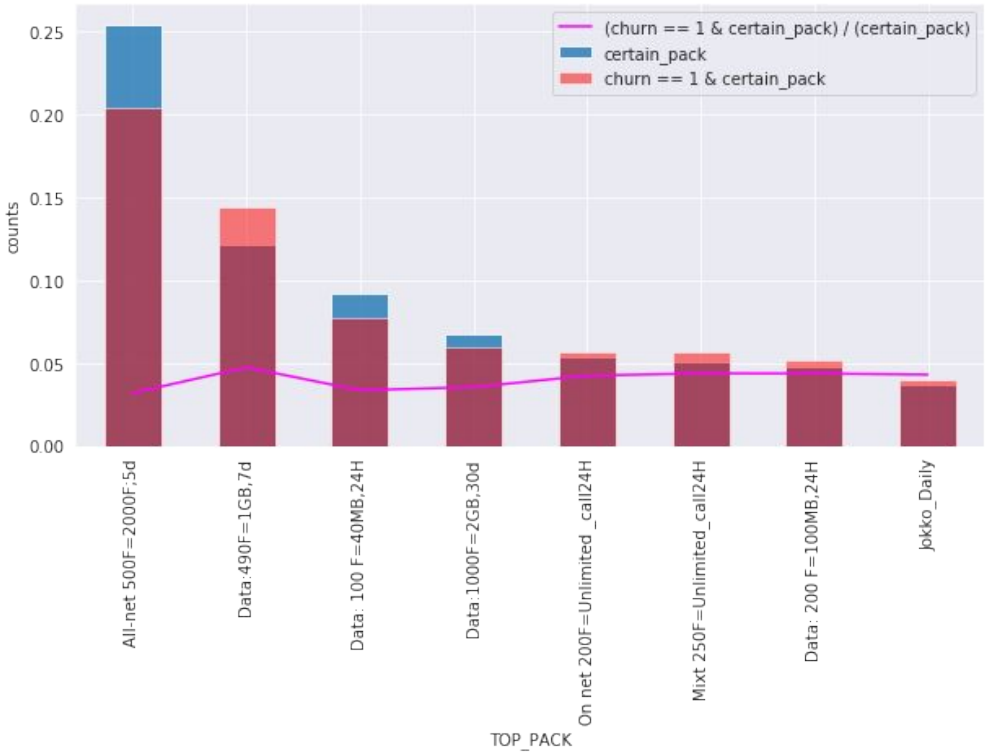
- REGION - the location of each client
- TENURE - duration in the network
- TOP_PACK - the most active packs
- ~~MRG - a client who is going (all values are the same)~~

Data Exploration: Categorical features





Data Exploration: Categorical features



Data Preprocessing

1. **Splitting** of initial train data (2 154 048 objects):
   56% - for training, 14% - for validation, 30% - for local testing
   (the model is **trained again on all train data** before submitting)
2. **Stratification**
3. Working with NaNs. Replace by **mean**, **median, zero or combination**
4. Working with NaNs after splitting to **avoid data leaks**
5. **Normalization** or **scaling** in [0, 1] for numeric features or **not any** standardization
6. Fixed **random_state**

Feature engineering: Add numerical features

1. **Total number of calls** (ON_NET + ORANGE + TIGO + ZONE1 + ZONE2)
   To evaluate the overall user activity
2. **Average top-up amount** (MONTANT / FREQUENCE_RECH)
   “*Reliable” users can put less often, but more*
3. **Connections density** (DATA_VOLUME / REGULARITY) **and**
4. **Call density** (Total number of calls / REGULARITY)
   It should be high for "reliable" users
5. **Income from the user per month minus its top up amount**
   (ARPU_SEGMENT - MONTANT)
   An "unreliable" user may also be unprofitable for the company
6. **Arpu last to average** (ARPU_SEGMENT / REVENUE)

Random Forest: Baseline

We used model with **max_depth=7**, **n_estimators=200** and **max_samples=1**

Data modifications	**AUC**
Replace NaNs by mean	0.896979
Include encoded ‘TENURE’ feature	0.895807
Replace NaNs by median	0.897719
Replace NaNs by zeros	0.897601
Change train size	0.8981661
Without Normalization	0.8981660
MinMaxScaler	0.898169
...	...
Add new numerical features	**0.898878**

Random Forest: Model Selection and Ensembling

- After data manipulation, we use GridSearchCV with AUC-scoring to select the best RF model. Our grids:
  - n_estimators: [150, 200, 250, 300, 350, 400]
  - max_samples: [0.4, 0.6, 0.8, 1.0]
  - max_depth: [7, 9, 12, 15]

  Total: 96 variants of models. The evaluation of hyperparameters was done on a validation dataset.
  Best RF model's parameters: {'max_depth': **7**, 'max_samples': **0.6**, 'n_estimators': **300**}
  AUC on the test: 0.89906

- After that, we took the top 10 models and averaged their probability predictions.
  AUC on the test: 0.8988

Logistic Regression

Best parameters (grid search): l1 regularization, C = 10

Preprocessing: NaN to mean and zeros, Category encoder

Best Validation AUC: 0.928629

Gradient Boosting (Light GBM)

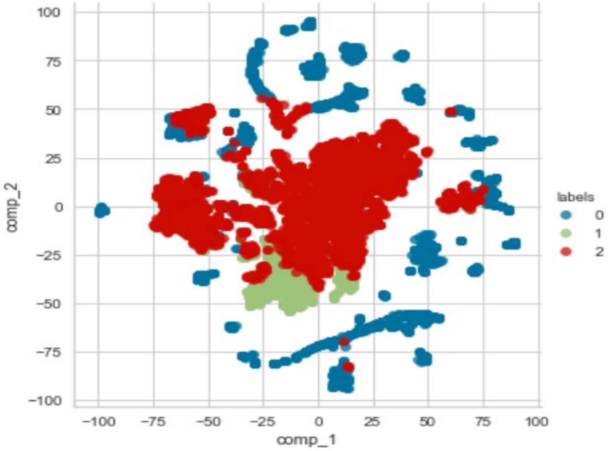
Data modification	**Validation AUC**	**Test Leaderboard AUC**
-	0.93149	
NaN to mean	0.93139	
NaN to mean and zeros	0.93125	
Category encoder	0.9315	
NaN to mean + Category encoder	0.93136	
New features	0.93151	0.93156
New features + Category encoder	**0.93154**	**0.93164**

Best params: n_estimators=200, learning_rate=0.05, min_child_samples=30, num_leaves=127
(by grid search)

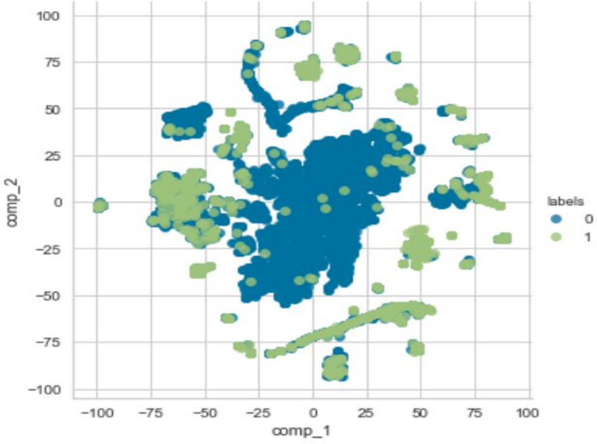
Clustering

tSNE visualization

k-means(k = 3)

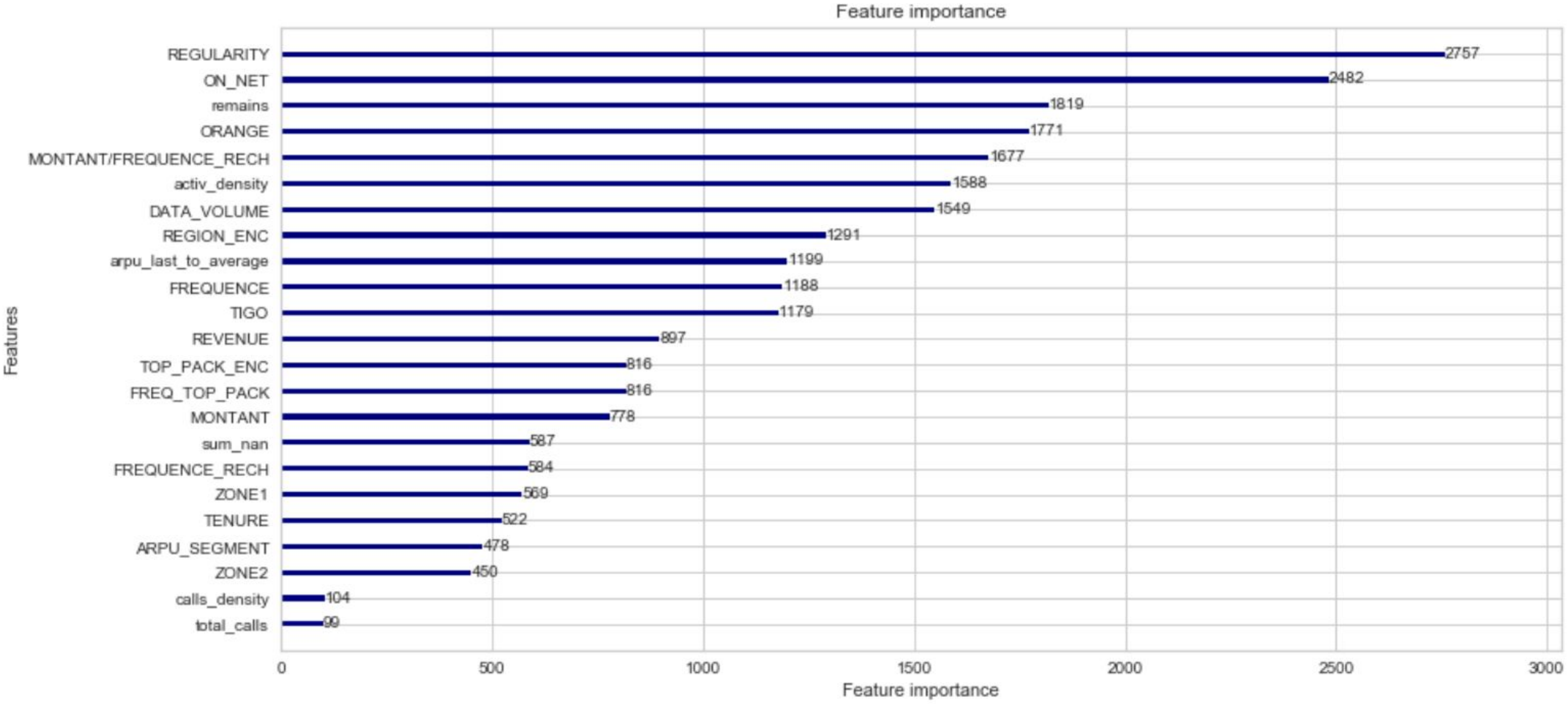


target



	Validation AUC
Additional feature	0.9341
Different models for different clusters	0.877

Feature importance



Conclusions

1. We **carefully examined the data transformations**, checking the work of the transformation efficiency and training a new model.
2. **Categorical features** that were discarded at the beginning, at the end made a significant contribution to the speed of the model.
3. We also investigated simpler and more **interpretable** models like Logistic Regression and found that they work well in this task.
4. Our best score = **0.93164** (public leaderboard position 42, best public leaderboard score 0.93346).

Thank you for attention!