Expresso Churn Prediction

Lina Bashaeva
Dmitry Gilyov

Problem description

- Expresso is an African telecommunications company;
- **Objective** - to develop an ML model predicting the likelihood of each customer becoming inactive for 90 days;
- **Solution** can help Expresso to prevent their customers from leaving.

Loading the data

Data files:

- **Train.csv** - contains information about 2 million customers with target value called ‘CHURN’;
- **Test.csv** - similar to train, but without the Churn column, 0.38 mln raws

Dataset preparation: data types

Numerical features:

- MONTANT
- FREQUENCE_RECH
- REVENUE
- ARPU_SEGMENT
- FREQUENCE
- DATA_VOLUME
- ON_NET
- ORANGE
- TIGO
- REGULARITY
- FREQ_TOP_PACK

Categorical features:

- REGION
- TENURE
- TOP_PACK

Dataset preparation: encoding

In [32]:
```
cat_cols = ['REGION','TENURE']

X_cat_cols_train = pd.get_dummies(X_train[cat_cols])
X_cat_cols_test = pd.get_dummies(X_test[cat_cols])
X_cat_cols = pd.get_dummies(X[cat_cols])
X_cat_cols
```

executed in 1.98s, finished 21:01:23 2021-10-24

Out[32]:

Jnknown	REGION_ZIGUINCHOR	TENURE_D 3-6 month	TENURE_E 6-9 month	TENURE_F 9-12 month	TENURE_G 12-15 month	TENURE_H 15-18 month	TENURE_I 18-21 month	TENURE_J 21-24 month	TENURE_K > 24 month
0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...
1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1

Dataset preparation

creating new features:

```
Names = ['=', 'GB', 'd', 'Data', 'Daily', 'MIXT', 'NO_PACK', 'net']
lsts = [[] for _ in range(len(Names))]
values = df.TOP_PACK.unique()
for i in range(len(Names)):
  for j in range(len(values)):
    if Names[i] in values[j]:
      lsts[i].append(values[j])
  df['packs_'+ Names[i]] = df['TOP_PACK'].apply(lambda x: 1 if Names[i] in x else 0)

```

```
pack_cols = ['packs_' + name for name in Names]
pack_cols.append('CHURN')
df[pack_cols].corr()
```

	packs_=	packs_GB	packs_d	packs_Data	packs_Daily	packs_MIXT	packs_NO_PACK	packs_net	CHURN
packs_=	1.000000	0.312418	0.811987	0.435835	-0.226264	0.114699	-0.861783	0.605699	-0.385020
packs_GB	0.312418	1.000000	0.318903	0.718294	-0.059981	-0.066934	-0.273577	-0.225080	-0.101631
packs_d	0.811987	0.318903	1.000000	0.129310	-0.143326	0.200278	-0.791012	0.731455	-0.358982
packs_Data	0.435835	0.718294	0.129310	1.000000	-0.090734	-0.093180	-0.380855	-0.313340	-0.145700
packs_Daily	-0.226264	-0.059981	-0.143326	-0.090734	1.000000	-0.046330	-0.189363	-0.155794	-0.080412
packs_MIXT	0.114699	-0.066934	0.200278	-0.093180	-0.046330	1.000000	-0.176458	0.296524	-0.087780
packs_NO_PACK	-0.861783	-0.273577	-0.791012	-0.380855	-0.189363	-0.176458	1.000000	-0.593382	0.444594
packs_net	0.605699	-0.225080	0.731455	-0.313340	-0.155794	0.296524	-0.593382	1.000000	-0.285122
CHURN	-0.385020	-0.101631	-0.358982	-0.145700	-0.080412	-0.087780	0.444594	-0.285122	1.000000

Dataset preparation: correlation between some features

Dataset preparation: dropped features

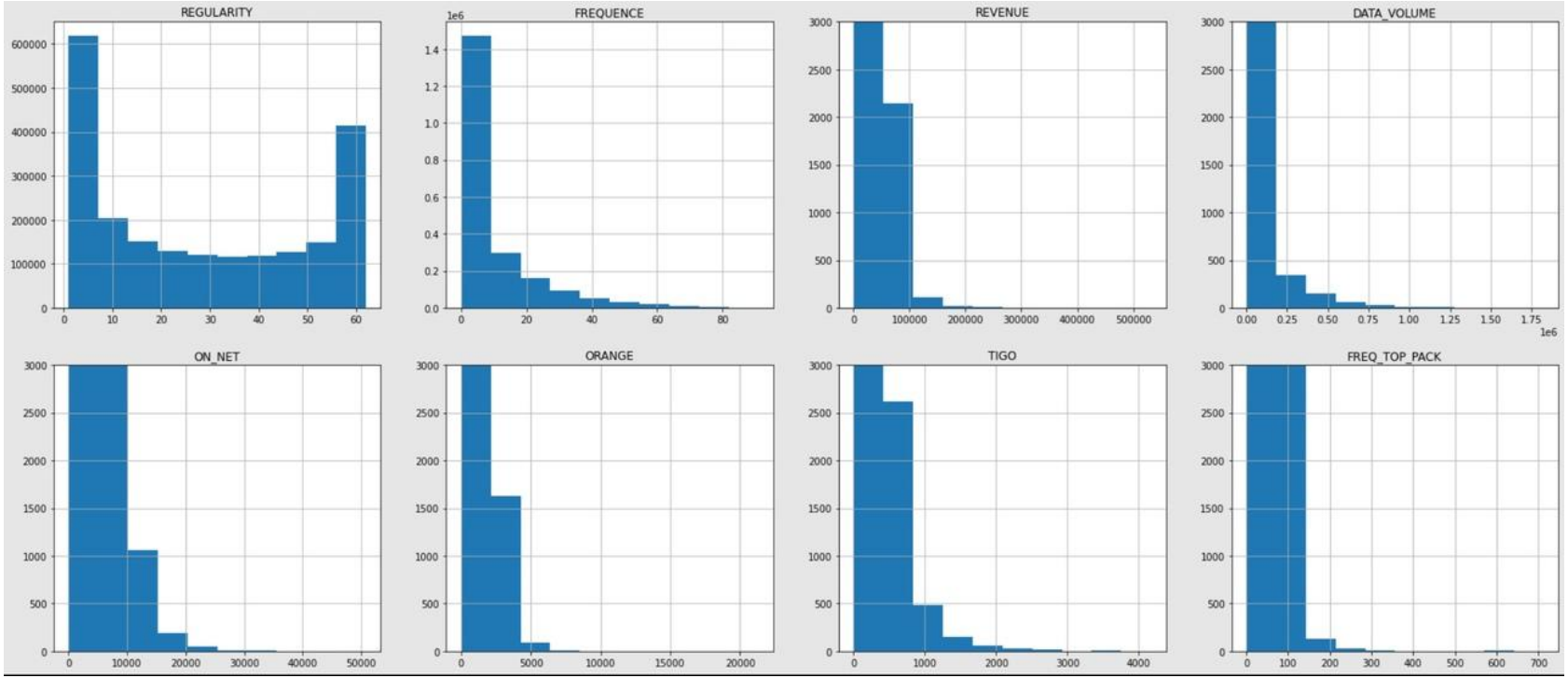
We dropped the following features:

- “MRG” (same value for all clients)
- “Montant” (0.98 correlation with “Revenue”)
- “Frequence_rech” (0.96 correlation with “Frequence”)
- “Arpu_segment” (1.00 correlation with “Revenue”)

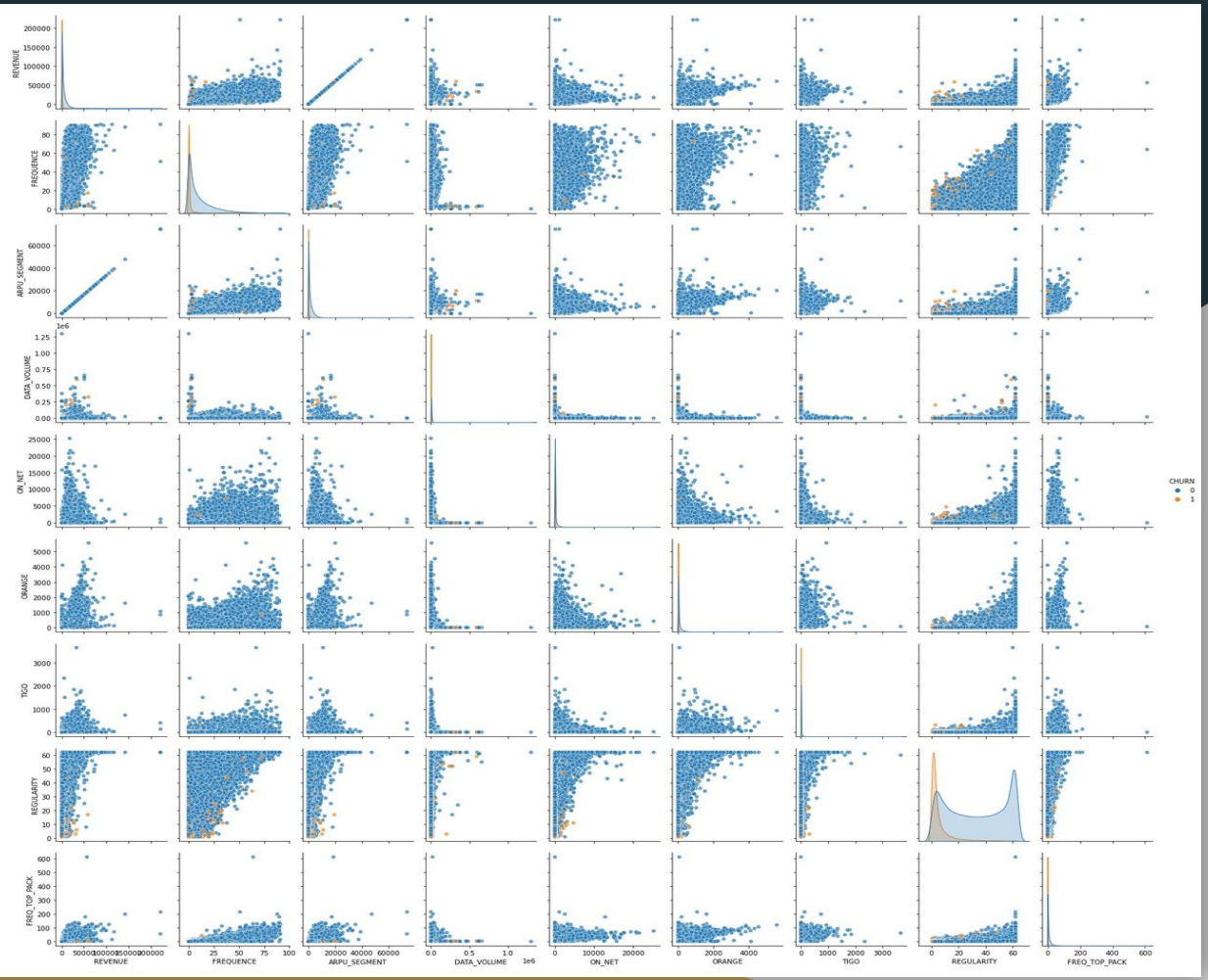
Dataset preprocessing: dealing with NaN values

- We filled **numerical** NaN values with zeros;
- **“Region”** NaNs were replaced with the class ‘Unknown’;
- **“Top_pack”** NaNs were replaced with ‘No_Pack’ for better understanding;

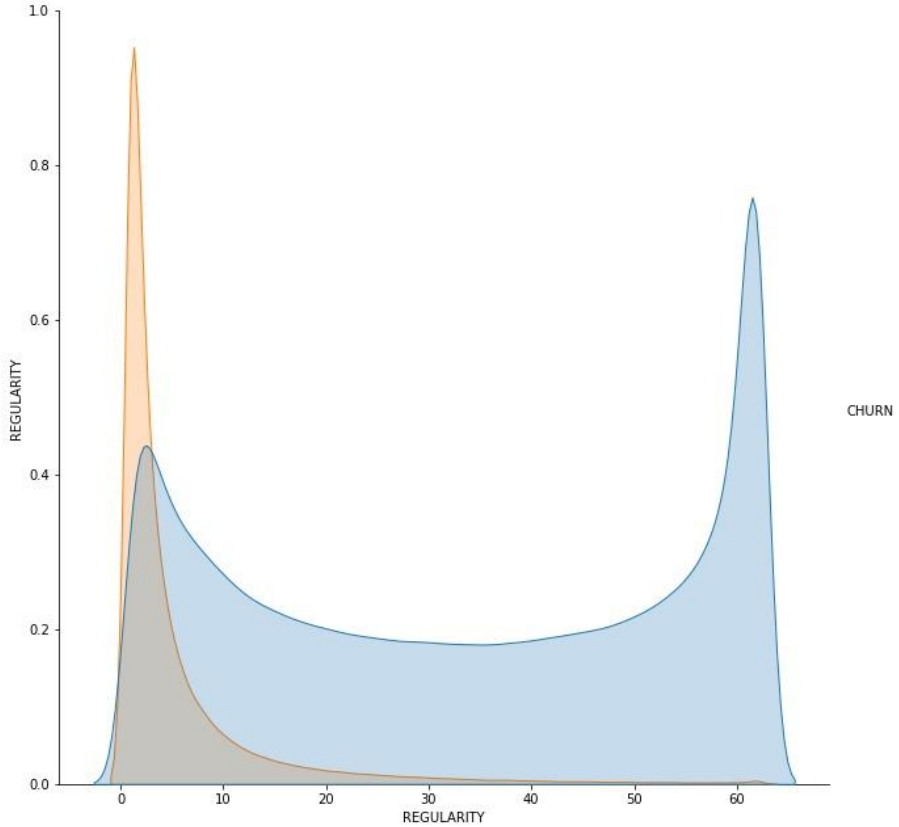
Visualisation: histogram of numerical features



Visualisation: pairplot

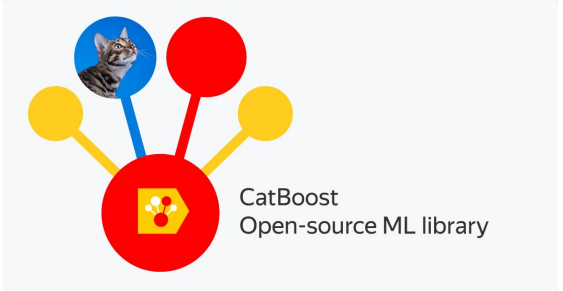


Visualisation: interesting finding



The feature has a non-trivial distribution with two peaks. corresponding to different classes of the target value

Choosing ML model: Catboost



Why Catboost?

We have a classification problem

- Gradient boosting with decision trees as weak learners is appropriate for classification problem;
- The library uses technique called oblivious decision tree, which is more effective than classic decision tree;
- Catboost allows to work with categorical features without one-hot encoding;
- Has a big variety of parameters.

CatBoostClassifier: results

Parameters of used classifier:

- iterations=800;
- learning_rate=0.08;
- depth=5.

Obtained results:

- Logloss = 0.253;
- F1 = 81.245 %;
- AUC-ROC = 79.207% .

Cross-validation procedure:results

	iterations	test-Logloss-mean	test-Logloss-std	train-Logloss-mean	train-Logloss-std
595	595	0.253289	0.000391	0.253161	0.000129
596	596	0.253288	0.000392	0.253159	0.000128
597	597	0.253287	0.000391	0.253158	0.000129
598	598	0.253286	0.000391	0.253157	0.000129
599	599	0.253285	0.000392	0.253156	0.000128

Features' importance

- Here is importance of some features, calculated by our model using CatBoost
- For each feature, it shows how much on average the prediction changes if the feature value changes

Feature name	Importance
Region	43,6
Tenure	0,746
Montant	1,77
Frequence_rech	1,63
Revenue	1,00
Arpu_segment	0,94
Frequence	1,92
Data_volume	4,12
On_net	2,82
Orange	1,17
Tigo	1,06
Zone1	0,73

Summing up

- We preprocessed all input data for the model training;
- We thoroughly explored all features and found all dependencies;
- Engineered a number of new features, correlating with target value;
- Trained a gradient boosting algorithm on decision trees;
- Achieved 79% quality in terms of area under ROC curve;
- The algorithm can be used in real life to prevent clients from leaving.