No-show - different approach: feature engineering, ML, DL, anomaly detection and AUTOML models

Magdalena Cebula [1,2] Jan Tarasiewicz [1] Dominik Stępień [1] Salem Albarudy [1] Adam Skorek [1] Jędrzej Dudzicz [1] Małgorzata Stolarska[1] Mateusz Rymuza[1]

[1]WMMData [2]Orange

Some about data

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	29872499824296.0000	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0	0	0	0	No
1	558997776694438.0000	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	0	0	0	0	No
2	4262962299951.0000	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	0	0	0	0	No
3	867951213174.0000	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0	0	0	0	No
4	8841186448183.0000	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1	0	0	0	No

Figure 1. Dataset.

Feature engineering - first approach

AppointmentID

We use this column for count waitingTime feature and delete it.

Gender

We leave this column with change F/M to 0/1

Age

We leave this column and drop value < 0 or value > 110.

Neighbourhood

We use one hot encoding.

PatientId, Scholarship, Hipertension, Diabets, Alcoholism, Handcap, SMSRecived

We leave those columns as were.

Dates

We create feature like hour, day and month, dayOfWeek from both dates and waitingTime.

Train-validation-test split

We split data totaly randomly. This is wrong approach, because it can lead to data leakage.

Feature engineering - second approach

Neighbourhood

We create feature "neighbourhood ratio", where add no-show/show ratio to each neighbourhood. We create category "OTHER" for this neighbourhood which had less than X visit (where X was param, default 200).

Handcap

We create category "OTHER" for Handcap > 1, because Handcap = 2 has 183 occurrence, Handcap = 3 has 13 occurrence, Handcap = 4 has 3 occurrence.

PatientId, Scholarship, Hipertension, Diabets, Alcoholism, SMSRecived

We leave those columns as were.

Dates

In our second approach we create feature from patient history: "lastShowTime", "sumAppointxD", "DayOfWeekRatio", "NoShowsRatioxD", "RecentNoShows", "sumNoShowsxD", "lastNoShowTime", "sumRecentNoShows"

Feature engineering - second approach

	PatientId	ScheduledDay	AppointmentDay	No-show	_waitingTime	_lastShowTime	_lastNoShowTime	_sumRecentNoShows	_sumAppoint_xD	_sumNoShows_xD
45137	9715135545613	2016-05-04 16:05:24	2016-05-04	1	0	29	29	0	0	0
45138	9715135545613	2016-05-04 16:08:03	2016-05-04	1	0	29	0	1	1	1
45139	9715135545613	2016-05-04 16:10:18	2016-05-04	1	0	29	0	2	2	2
45140	9715135545613	2016-05-06 19:12:17	2016-05-06	0	0	29	2	3	3	3
45141	9715135545613	2016-05-10 13:31:46	2016-05-10	0	0	4	6	0	4	3
45142	9715135545613	2016-05-10 13:36:24	2016-05-10	0	0	0	6	0	5	3
45143	9715135545613	2016-05-11 07:47:24	2016-05-11	1	0	1	7	0	6	3
45144	9715135545613	2016-05-11 07:55:08	2016-05-11	1	0	1	0	1	7	4
45145	9715135545613	2016-05-11 16:09:40	2016-05-11	1	0	1	0	2	8	5
45146	9715135545613	2016-05-12 18:49:20	2016-05-12	0	0	2	1	3	9	6

Figure 2. Sample client after transformation and feature engineering.

We thing about how long should be window of history (param xD). Because if we get all history, the new patient has less records/visits than older patient (with a lot of visits).

Train-validation-test split

We draw PatientId which was selected to train, validation, test dataset.

Problem with unbalanced data

In this dataset, we have unbalanced classes.

- **No:** 0.7981
- **Yes:** 0.2019

We test a few methods to balance data: Random-Over Sampling, Random-Under Sampling, Nearmiss, ADASYN, SMOTETomek

The best result get SMOTETomek, so we used them in final pipeline. We test and validate data also in dataset without balancing.

We test anomaly dateection approach too.

Other transformation

We use pipelines and fit-tranform technic. In our pipeline we have out custom transforer, imputer, scaler and sampler and model.

```
pipeline = Pipeline([
    ("custom", MedicalAppointmentsTransformer(neighbourhood_min_count = 200)),
    ("imputer", SimpleImputer(strategy="mean")),
    ("scaler", StandardScaler()),
    ("oversampling", SMOTETomek()),
    ("model", XGBClassifier(random_state=99))
])
```

Figure 3. Example pipeline.

	Scholarship	Hipertension	Diabetes	Alcoholism	SMS_received	_waitingTime	_lastShowTime	_sumAppoint_xD	_NeighbourhoodRatio
0	-0.3327	-0.4902	-0.2743	-0.1722	-0.6384	-0.4850	0.6162	-0.3533	1.7965

_DayOfWeekRatio	_NoShowsRatio_xD	_RecentNoShows	_Age	_Gender	_Handcap	_sumNoShows_xD	_lastNoShowTime	_sumRecentNoShows
1.8167	-0.3123	-0.3161	0.2903	0.7287	-0.1451	-0.2999	0.3113	-0.2703

Figure 4. Example row after transformation.

Machine learning and Deep learning models

We test a few ML models: Logistic Regression, Decision Tree, Random Forest, XGBoost, CatBoost, LigthGBM, SVC, KNN, Naive Bayes and MLP. We select F1 as our metric to evaluate and compare models.

Model	F1	Accuracy	Precision	Recall	ROC
XGBoost	0.45	0.70	0.35	0.62	0.67
MLP	0.44	0.65	0.32	0.69	0.66
SVC	0.43	0.65	0.32	0.68	0.66
Logistig Regression	0.41	0.66	0.31	0.60	0.64
Naive Bayes	0.40	0.61	0.29	0.65	0.63

Table 1. Top 5 models

Anomaly detection

We test approach to detect no show as anomaly. Unfortunatelly, Isolation forest and autoencoder get worse scores than classic ml models.

F1: 0.25 Accuracy: 0.73 Precision: 0.23 Recall: 0.28

AutoML

We test a few AutoML tools, the best was Azure AutoML. In our case the best model was RandomForest with StandardScalerWrapper - with f1 = 0.54.

This AutoML have explanations dashboard and we can build responsible AI dashboard.

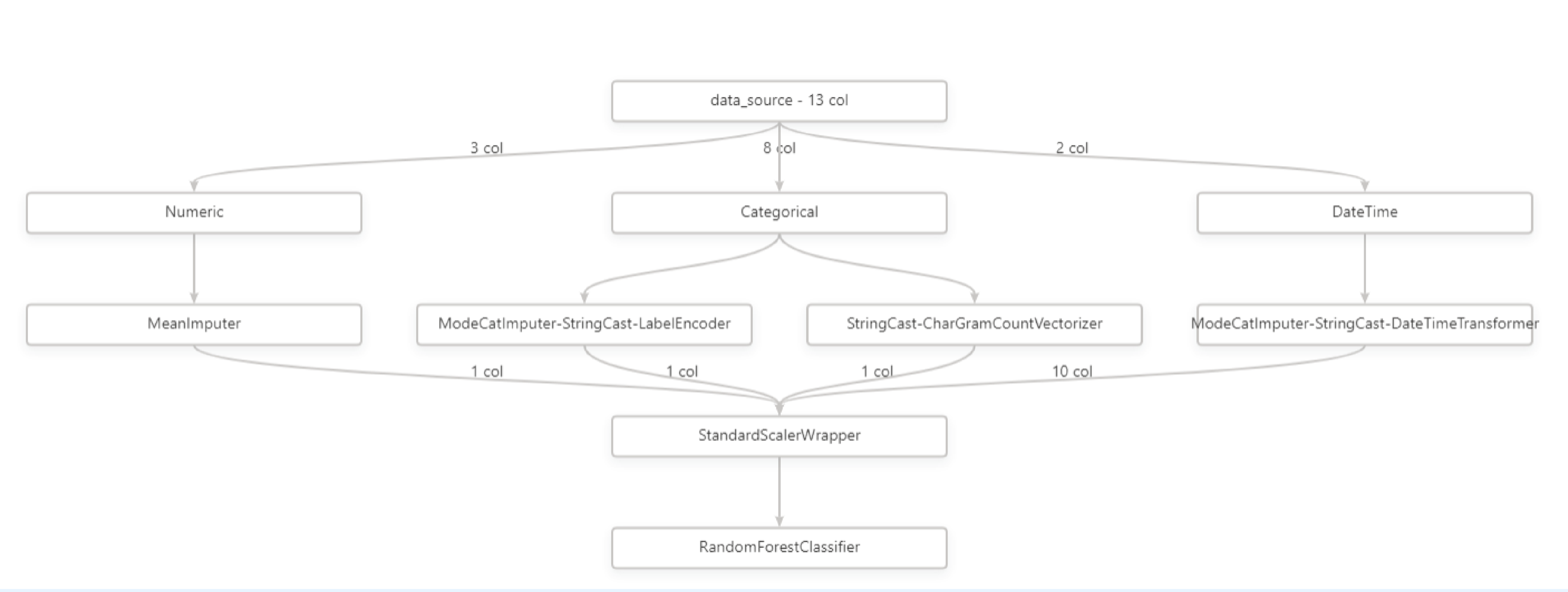

Figure 5. Data transformation from AutoML.

References

https://github.com/DataWorkshop-Foundation/warsaw-project-2

https://www.kaggle.com/datasets/joniarroba/noshowappointments?resource=download

 magdalena.oliwia.cebula@gmail.com