homecredit_machine_learning

August 30, 2023

1 The analysis of the Home Credit Group dataset (II part)

1.0.1 Importing libraries

The main libraries which will be used for the manipulation with data are pandas, suckb and numpy. Matplotlib, seaborn and yellowbrick will be used for data visualization. Scipy, Statsmodels, Researchpy, Math, Random will be used for conducting statistical tests, calculating confidence intervals. Sklearn and imblearn will be used for spliting data into training and testing samples, building and testing machine learning models. Optuna will be used for hyperparameter tuning. Tensorflow library will be used for deep learning.

The functions which will be created in the process of analysis will be uploaded from a file "home-credit_utils.py".

Importing the initially preprocessed datasets The files which were created as the result of exploratory analysis and initial preprocessing of the Home Credit Club data (see the file "home-credit_exploratory_analysis.ipynb" are imported and saved into pandas dataframes.

1.0.2 Machine learning

Randomly selecting the data for machine learning In order to train the model which would predict probabilities if clients of the Home Credit Club are risky (that is, have difficulties in paying loans in time), the data was randomly sampled from the fulldata_train dataset.

Spliting the data into training, validation, and test datasets First, the data were split into feature variables and a target variable as well as into training, validation and test datasets.

Building a machine learning pipelines Next, pipelines of transforming data were constructed. First, names of binary, categorical (with multiple values) and numerical normalized and other numerical features were identified and saved into separate lists.

For separation of binary features the functions 'binary_numeric" and 'get_binary_numeric" were applied.

lists of features:

```
"NAME_CONTRACT_TYPE", "FLAG_OWN_REALTY", "FLAG_EMP_PHONE", "FLAG_WORK_PHONE",
"FLAG_CONT_MOBILE", "FLAG_PHONE", "FLAG_EMAIL", "REG_REGION_NOT_LIVE_REGION",
"REG_REGION_NOT_WORK_REGION", "LIVE_REGION_NOT_WORK_REGION",
"REG_CITY_NOT_LIVE_CITY", "REG_CITY_NOT_WORK_CITY", "LIVE_CITY_NOT_WORK_CITY",
```

"ORGANIZATION_TYPE_Advertising", "ORGANIZATION_TYPE_Agriculture",
"ORGANIZATION_TYPE_Bank", "ORGANIZATION_TYPE_Business Entity Type 1",
"ORGANIZATION_TYPE_Business Entity Type 2", "ORGANIZATION_TYPE_Business Entity
Type 3", "ORGANIZATION_TYPE_Cleaning", "ORGANIZATION_TYPE_Construction",
"ORGANIZATION_TYPE_Culture", "ORGANIZATION_TYPE_Electricity",
"ORGANIZATION_TYPE_Emergency", "ORGANIZATION_TYPE_Government",
"ORGANIZATION_TYPE_Hotel", "ORGANIZATION_TYPE_Housing",
"ORGANIZATION_TYPE_Industry: type 1", "ORGANIZATION_TYPE_Industry: type 10",
"ORGANIZATION_TYPE_Industry: type 11", "ORGANIZATION_TYPE_Industry: type 12",
"ORGANIZATION_TYPE_Industry: type 13", "ORGANIZATION_TYPE_Industry: type 2",
"ORGANIZATION_TYPE_Industry: type 3", "ORGANIZATION_TYPE_Industry: type 4",
"ORGANIZATION_TYPE_Industry: type 5", "ORGANIZATION_TYPE_Industry: type 6",
"ORGANIZATION_TYPE_Industry: type 7", "ORGANIZATION_TYPE_Industry: type 8",
"ORGANIZATION_TYPE_Industry: type 9", "ORGANIZATION_TYPE_Insurance",
"ORGANIZATION_TYPE_Kindergarten", "ORGANIZATION_TYPE_Legal Services",
"ORGANIZATION_TYPE_Medicine", "ORGANIZATION_TYPE_Military",
"ORGANIZATION_TYPE_Mobile", "ORGANIZATION_TYPE_Other",
"ORGANIZATION_TYPE_Police", "ORGANIZATION_TYPE_Postal",
"ORGANIZATION_TYPE_Realtor", "ORGANIZATION_TYPE_Religion",
"ORGANIZATION_TYPE_Restaurant", "ORGANIZATION_TYPE_School",
"ORGANIZATION_TYPE_Security", "ORGANIZATION_TYPE_Security Ministries",
"ORGANIZATION_TYPE_Self-employed", "ORGANIZATION_TYPE_Services",
"ORGANIZATION_TYPE_Telecom", "ORGANIZATION_TYPE_Trade: type 1",
"ORGANIZATION_TYPE_Trade: type 2", "ORGANIZATION_TYPE_Trade: type 3",
"ORGANIZATION_TYPE_Trade: type 4", "ORGANIZATION_TYPE_Trade: type 5",
"ORGANIZATION_TYPE_Trade: type 6", "ORGANIZATION_TYPE_Trade: type 7",
"ORGANIZATION_TYPE_Transport: type 1", "ORGANIZATION_TYPE_Transport: type 2",
"ORGANIZATION_TYPE_Transport: type 3", "ORGANIZATION_TYPE_Transport: type 4",
"ORGANIZATION_TYPE_University", "GENDER_F", "GENDER_M",
"NAME_GOODS_CATEGORY_House Construction", "NAME_CASH_LOAN_PURPOSE_Refusal to
name the goal", "CREDIT_ACTIVE_Bad debt", "CREDIT_TYPE_Loan for purchase of
shares (margin lending)", "CREDIT_TYPE_Loan for the purchase of equipment",
"CREDIT_TYPE_Mobile operator loan", "AVG_NAME_CONTRACT_STATUS_Amortized_debt",
"AVG_NAME_CONTRACT_STATUS_CC_Approved"

"REGION_POPULATION_RELATIVE", "FLAG_MOBIL", "FLAG_EMP_PHONE", "FLAG_WORK_PHONE",
"FLAG_CONT_MOBILE", "FLAG_PHONE", "FLAG_EMAIL", "REG_REGION_NOT_LIVE_REGION",
"REG_REGION_NOT_WORK_REGION", "LIVE_REGION_NOT_WORK_REGION",
"REG_CITY_NOT_LIVE_CITY", "REG_CITY_NOT_WORK_CITY", "LIVE_CITY_NOT_WORK_CITY",
"EXT_SOURCE_1", "EXT_SOURCE_2", "EXT_SOURCE_3", "TOTALAREA_MODE",
"ORGANIZATION_TYPE_Advertising", "ORGANIZATION_TYPE_Agriculture",
"ORGANIZATION_TYPE_Bank", "ORGANIZATION_TYPE_Business Entity Type 1",
"ORGANIZATION_TYPE_Business Entity Type 2", "ORGANIZATION_TYPE_Business Entity
Type 3", "ORGANIZATION_TYPE_Cleaning", "ORGANIZATION_TYPE_Construction",
"ORGANIZATION_TYPE_Culture", "ORGANIZATION_TYPE_Electricity",
"ORGANIZATION_TYPE_Emergency", "ORGANIZATION_TYPE_Government",
"ORGANIZATION_TYPE_Hotel", "ORGANIZATION_TYPE_Housing",
"ORGANIZATION_TYPE_Industry: type 1", "ORGANIZATION_TYPE_Industry: type 10",

"ORGANIZATION_TYPE_Industry: type 11", "ORGANIZATION_TYPE_Industry: type 12",
"ORGANIZATION_TYPE_Industry: type 13", "ORGANIZATION_TYPE_Industry: type 2",
"ORGANIZATION_TYPE_Industry: type 3", "ORGANIZATION_TYPE_Industry: type 4",
"ORGANIZATION_TYPE_Industry: type 5", "ORGANIZATION_TYPE_Industry: type 6",
"ORGANIZATION_TYPE_Industry: type 7", "ORGANIZATION_TYPE_Industry: type 8",
"ORGANIZATION_TYPE_Industry: type 9", "ORGANIZATION_TYPE_Insurance",
"ORGANIZATION_TYPE_Kindergarten", "ORGANIZATION_TYPE_Legal Services",
"ORGANIZATION_TYPE_Medicine", "ORGANIZATION_TYPE_Military",
"ORGANIZATION_TYPE_Mobile", "ORGANIZATION_TYPE_Other",
"ORGANIZATION_TYPE_Police", "ORGANIZATION_TYPE_Postal",
"ORGANIZATION_TYPE_Realtor", "ORGANIZATION_TYPE_Religion",
"ORGANIZATION_TYPE_Restaurant", "ORGANIZATION_TYPE_School",
"ORGANIZATION_TYPE_Security", "ORGANIZATION_TYPE_Security Ministries",
"ORGANIZATION_TYPE_Self-employed", "ORGANIZATION_TYPE_Services",
"ORGANIZATION_TYPE_Telecom", "ORGANIZATION_TYPE_Trade: type 1",
"ORGANIZATION_TYPE_Trade: type 2", "ORGANIZATION_TYPE_Trade: type 3",
"ORGANIZATION_TYPE_Trade: type 4", "ORGANIZATION_TYPE_Trade: type 5",
"ORGANIZATION_TYPE_Trade: type 6", "ORGANIZATION_TYPE_Trade: type 7",
"ORGANIZATION_TYPE_Transport: type 1", "ORGANIZATION_TYPE_Transport: type 2",
"ORGANIZATION_TYPE_Transport: type 3", "ORGANIZATION_TYPE_Transport: type 4",
"ORGANIZATION_TYPE_University", "GENDER_F", "GENDER_M", "NAME_CONTRACT_TYPE_Cash
loans", "NAME_CONTRACT_TYPE_Consumer loans", "NAME_CONTRACT_TYPE_Revolving
loans", "FLAG_LAST_APPL_PER_CONTRACT_N", "FLAG_LAST_APPL_PER_CONTRACT_Y",
"NFLAG_LAST_APPL_IN_DAY_0", "NFLAG_LAST_APPL_IN_DAY_1",
"NAME_CONTRACT_STATUS_Approved", "NAME_CONTRACT_STATUS_Canceled",
"NAME_CONTRACT_STATUS_Refused", "NAME_CONTRACT_STATUS_Unused offer",
"NAME_PAYMENT_TYPE_Cash through the bank", "NAME_PAYMENT_TYPE_Cashless from the
account of the employer", "NAME_PAYMENT_TYPE_Non-cash from your account",
"CODE_REJECT_REASON_CLIENT", "CODE_REJECT_REASON_HC",
"CODE_REJECT_REASON_LIMIT", "CODE_REJECT_REASON_SCO",
"CODE_REJECT_REASON_SCOFR", "CODE_REJECT_REASON_SYSTEM",
"CODE_REJECT_REASON_VERIF", "NAME_TYPE_SUITE_Children",
"NAME_TYPE_SUITE_Family", "NAME_TYPE_SUITE_Group of people",
"NAME_TYPE_SUITE_Other_A", "NAME_TYPE_SUITE_Other_B", "NAME_TYPE_SUITE_Spouse,
partner", "NAME_TYPE_SUITE_Unaccompanied", "NAME_CLIENT_TYPE_New",
"NAME_CLIENT_TYPE_Refreshed", "NAME_CLIENT_TYPE_Repeater",
"NAME_PORTFOLIO_Cards", "NAME_PORTFOLIO_Cars", "NAME_PORTFOLIO_Cash",
"NAME_PORTFOLIO_POS", "NAME_PRODUCT_TYPE_walk-in", "NAME_PRODUCT_TYPE_x-sell",
"CHANNEL_TYPE_AP+ (Cash loan)", "CHANNEL_TYPE_Car dealer", "CHANNEL_TYPE_Channel
of corporate sales", "CHANNEL_TYPE_Contact center", "CHANNEL_TYPE_Country-wide",
"CHANNEL_TYPE_Credit and cash offices", "CHANNEL_TYPE_Regional / Local",
"CHANNEL_TYPE_Stone", "NAME_YIELD_GROUP_high", "NAME_YIELD_GROUP_low_action",
"NAME_YIELD_GROUP_low_normal", "NAME_YIELD_GROUP_middle",
"NFLAG_INSURED_ON_APPROVAL_0.0", "NFLAG_INSURED_ON_APPROVAL_1.0",
"NAME_GOODS_CATEGORY_Additional Service", "NAME_GOODS_CATEGORY_Animals",
"NAME_GOODS_CATEGORY_Audio/Video", "NAME_GOODS_CATEGORY_Auto Accessories",
"NAME_GOODS_CATEGORY_Clothing and Accessories", "NAME_GOODS_CATEGORY_Computers",
"NAME_GOODS_CATEGORY_Construction Materials", "NAME_GOODS_CATEGORY_Consumer

Electronics", "NAME_GOODS_CATEGORY_Direct Sales",
"NAME_GOODS_CATEGORY_Education", "NAME_GOODS_CATEGORY_Fitness",
"NAME_GOODS_CATEGORY_Furniture", "NAME_GOODS_CATEGORY_Gardening",
"NAME_GOODS_CATEGORY_Homewares", "NAME_GOODS_CATEGORY_House Construction",
"NAME_GOODS_CATEGORY_Insurance", "NAME_GOODS_CATEGORY_Jewelry",
"NAME_GOODS_CATEGORY_Medical Supplies", "NAME_GOODS_CATEGORY_Medicine",
"NAME_GOODS_CATEGORY_Mobile", "NAME_GOODS_CATEGORY_Office Appliances",
"NAME_GOODS_CATEGORY_Other", "NAME_GOODS_CATEGORY_Photo / Cinema Equipment",
"NAME_GOODS_CATEGORY_Sport and Leisure", "NAME_GOODS_CATEGORY_Tourism",
"NAME_GOODS_CATEGORY_Vehicles", "NAME_GOODS_CATEGORY_Weapon",
"NAME_CASH_LOAN_PURPOSE_Building a house or an annex",
"NAME_CASH_LOAN_PURPOSE_Business development", "NAME_CASH_LOAN_PURPOSE_Buying a
garage", "NAME_CASH_LOAN_PURPOSE_Buying a holiday home / land",
"NAME_CASH_LOAN_PURPOSE_Buying a home", "NAME_CASH_LOAN_PURPOSE_Buying a new
car", "NAME_CASH_LOAN_PURPOSE_Buying a used car", "NAME_CASH_LOAN_PURPOSE_Car
repairs", "NAME_CASH_LOAN_PURPOSE_Education", "NAME_CASH_LOAN_PURPOSE_Everyday
expenses", "NAME_CASH_LOAN_PURPOSE_Furniture",
"NAME_CASH_LOAN_PURPOSE_Gasification / water supply",
"NAME_CASH_LOAN_PURPOSE_Hobby", "NAME_CASH_LOAN_PURPOSE_Journey",
"NAME_CASH_LOAN_PURPOSE_Medicine", "NAME_CASH_LOAN_PURPOSE_Money for a third
person", "NAME_CASH_LOAN_PURPOSE_Other", "NAME_CASH_LOAN_PURPOSE_Payments on
other loans", "NAME_CASH_LOAN_PURPOSE_Purchase of electronic equipment",
"NAME_CASH_LOAN_PURPOSE_Refusal to name the goal",
"NAME_CASH_LOAN_PURPOSE_Repairs", "NAME_CASH_LOAN_PURPOSE_Urgent needs",
"NAME_CASH_LOAN_PURPOSE_Wedding / gift / holiday", "PRODUCT_COMBINATION_Card
Street", "PRODUCT_COMBINATION_Card X-Sell", "PRODUCT_COMBINATION_Cash",
"PRODUCT_COMBINATION_Cash Street: high", "PRODUCT_COMBINATION_Cash Street: low",
"PRODUCT_COMBINATION_Cash Street: middle", "PRODUCT_COMBINATION_Cash X-Sell:
high", "PRODUCT_COMBINATION_Cash X-Sell: low", "PRODUCT_COMBINATION_Cash X-Sell:
middle", "PRODUCT_COMBINATION_POS household with interest",
"PRODUCT_COMBINATION_POS household without interest", "PRODUCT_COMBINATION_POS
industry with interest", "PRODUCT_COMBINATION_POS industry without interest",
"PRODUCT_COMBINATION_POS mobile with interest", "PRODUCT_COMBINATION_POS mobile
without interest", "PRODUCT_COMBINATION_POS other with interest",
"PRODUCT_COMBINATION_POS others without interest", "WWAVG_RATE_DOWN_PAYMENT",
"WWAVG_RATE_INTEREST_PRIMARY", "WWAVG_RATE_INTEREST_PRIVILEGED",
"CREDIT_ACTIVE_Active", "CREDIT_ACTIVE_Bad debt", "CREDIT_ACTIVE_Closed",
"CREDIT_ACTIVE_Sold", "CREDIT_CURRENCY_currency 1", "CREDIT_CURRENCY_currency
2", "CREDIT_CURRENCY_currency 3", "CREDIT_CURRENCY_currency 4",
"CREDIT_TYPE_Another type of loan", "CREDIT_TYPE_Car loan", "CREDIT_TYPE_Cash
loan (non-earmarked)", "CREDIT_TYPE_Consumer credit", "CREDIT_TYPE_Credit card",
"CREDIT_TYPE_Loan for business development", "CREDIT_TYPE_Loan for purchase of
shares (margin lending)", "CREDIT_TYPE_Loan for the purchase of equipment",
"CREDIT_TYPE_Loan for working capital replenishment", "CREDIT_TYPE_Microloan",
"CREDIT_TYPE_Mobile operator loan", "CREDIT_TYPE_Mortgage", "CREDIT_TYPE_Real
estate loan", "CREDIT_TYPE_Unknown type of loan", "WAVG_AVG_STATUS_1",
"WAVG_AVG_STATUS_2", "WAVG_AVG_STATUS_3", "WAVG_AVG_STATUS_4",
"WAVG_AVG_STATUS_5", "AVG_NAME_CONTRACT_STATUS_Active",

```
"AVG_NAME_CONTRACT_STATUS_Amortized_debt", "AVG_NAME_CONTRACT_STATUS_Approved",
"AVG_NAME_CONTRACT_STATUS_Canceled", "AVG_NAME_CONTRACT_STATUS_Completed",
"AVG_NAME_CONTRACT_STATUS_Demand",
"AVG_NAME_CONTRACT_STATUS_Returned_to_the_store",
"AVG_NAME_CONTRACT_STATUS_Signed", "AVG_NAME_CONTRACT_STATUS_CC_Active",
"AVG_NAME_CONTRACT_STATUS_CC_Approved", "AVG_NAME_CONTRACT_STATUS_CC_Completed",
"AVG_NAME_CONTRACT_STATUS_CC_Demand", "AVG_NAME_CONTRACT_STATUS_CC_Refused",
"AVG_NAME_CONTRACT_STATUS_CC_Sent_proposal",
"AVG_NAME_CONTRACT_STATUS_CC_Signed"
```

```
"SK_ID_CURR", "CNT_CHILDREN", "AMT_INCOME_TOTAL", "AMT_CREDIT", "AMT_ANNUITY",
"AMT_GOODS_PRICE", "DAYS_BIRTH", "DAYS_EMPLOYED", "DAYS_REGISTRATION",
"DAYS_ID_PUBLISH", "CNT_FAM_MEMBERS", "REGION_RATING_CLIENT",
"REGION_RATING_CLIENT_W_CITY", "OBS_60_CNT_SOCIAL_CIRCLE",
"DAYS_LAST_PHONE_CHANGE", "EDUCATION", "CAR_OWN", "LIVING_CONDITIONS_1",
"LIVING_CONDITIONS_2", "CB_enquiries_1", "CB_enquiries_2", "WWAVG_AMT_ANNUITY",
"WWAVG_AMT_APPLICATION", "WWAVG_AMT_CREDIT", "WWAVG_AMT_DOWN_PAYMENT",
"WWAVG_AMT_GOODS_PRICE", "WAVG_CREDIT_END_LATE", "WAVG_CREDIT_DAY_OVERDUE",
"WAVG_AMT_CREDIT_MAX_OVERDUE", "WAVG_CNT_CREDIT_PROLONG", "WAVG_AMT_CREDIT_SUM",
"WAVG_AMT_CREDIT_SUM_DEBT", "WAVG_AMT_CREDIT_SUM_LIMIT",
"WAVG_AMT_CREDIT_SUM_OVERDUE", "WAVG_DAYS_CREDIT_UPDATE", "WAVG_AVG_STATUS_0",
"WAVG_AVG_STATUS_C", "WAVG_AVG_STATUS_X", "sums_of_days_late",
"sums_of_days_in_time", "sums_of_amounts_late", "sums_of_amounts_in_time",
"CNT_INSTALMENT_WAVG", "CNT_INSTALMENT_FUTURE_WAVG", "SK_DPD_WAVG",
"SK_DPD_DEF_WAVG", "WWAVG_AMT_BALANCE", "WWAVG_AMT_CREDIT_LIMIT_ACTUAL",
"WWAVG_AMT_DRAWINGS_ATM_CURRENT", "WWAVG_AMT_DRAWINGS_CURRENT",
"WWAVG_AMT_DRAWINGS_OTHER_CURRENT", "WWAVG_AMT_DRAWINGS_POS_CURRENT",
"WWAVG_AMT_INST_MIN_REGULARITY", "WWAVG_AMT_DRAWINGS_POS_CURRENT_2",
"WWAVG_AMT_PAYMENT_TOTAL_CURRENT", "WWAVG_AMT_RECEIVABLE_PRINCIPAL",
"WWAVG_AMT_RECIVABLE", "WWAVG_AMT_TOTAL_RECEIVABLE",
"WWAVG_CNT_DRAWINGS_ATM_CURRENT", "WWAVG_CNT_DRAWINGS_CURRENT",
"WWAVG_CNT_DRAWINGS_OTHER_CURRENT", "WWAVG_CNT_DRAWINGS_POS_CURRENT",
"WWAVG_CNT_INSTALMENT_MATURE_CUM", "WWAVG_SK_DPD", "WWAVG_SK_DPD_DEF"
```

```
"NAME_CONTRACT_TYPE", "FLAG_OWN_REALTY", "NAME_TYPE_SUITE", "NAME_INCOME_TYPE",
"NAME_FAMILY_STATUS", "NAME_HOUSING_TYPE", "OCCUPATION_TYPE",
"FONDKAPREMONT_MODE", "HOUSETYPE_MODE", "WALLSMATERIAL_MODE",
"EMERGENCYSTATE_MODE"
```

Normalised numerical features were identified with the help of the function which selects those feature names from the list X.columns which are not present in other lists.

Next, classes for different types of features and lists of instances of those classes were created.

lists of transformers: Separate lists for various types of transformers were created in order to simplify the process of pipeline and feature engeneering (examining the effects of different transformers on the metrics of various machine learning models)

list of classifiers: Various classifiers were included into a list of classifiers. SVC classifier was excluded from the list in the intermediary stage of modelling, as it was observed that to train the SVC classifier takes long time while the classifier does not perform well.

Function for transformer pipelines: Pipelines of various transformers for different types of features were combined into a single function "get_transformers".

Running the machine learning pipeline The function for fitting classifiers, predicting the target variable in the validation dataset, cross-validation, getting metrics, printing metrics outputs and appending outputs into dictionnaries was created.

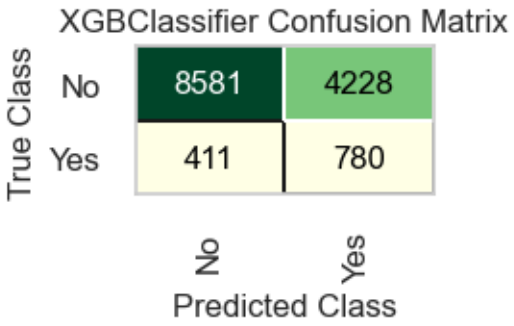
A function for building empty dictionnaries to save modelling outputs with keys for each type of data was created.

The function to loop over lists of features, pipelines of transformers and classifiers and to save data into dictionaries (to be transformed into pandas dataframes) and models into local files was created.

The function (presented above) was run on the mentioned lists. Outputs present metrics for different classifiers, combinations of features and transformers which were used for training the models are also mentioned in the output.

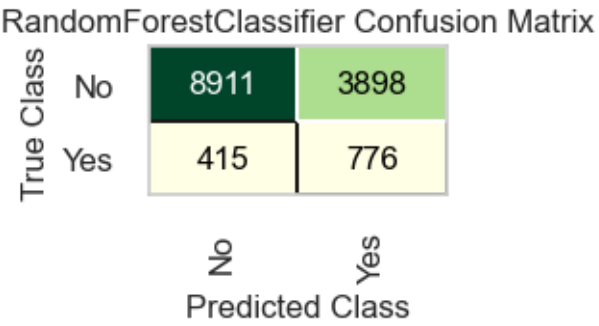
Initially, models were trained on the total number of available features (315).

```
Parameters for the dataset and transformers: OneHotEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.67      0.79     12809
         Yes       0.16      0.65      0.25      1191

    accuracy                           0.67     14000
```

```
   macro avg       0.56      0.66      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.66035714 0.67892857 0.66035714 0.65571429 0.67071429]
Accuracy score (average): 0.6652142857142858
F1 scores for 'Yes' values: [0.24102155 0.24135021 0.22367347 0.22006472
0.23294509]
Average F1 score: 0.519
ROC-AUC score: 0.662
PR-AUC score: 0.131
Log-loss: 0.726


Execution time: 59.361830949783325

RandomForestClassifier
```

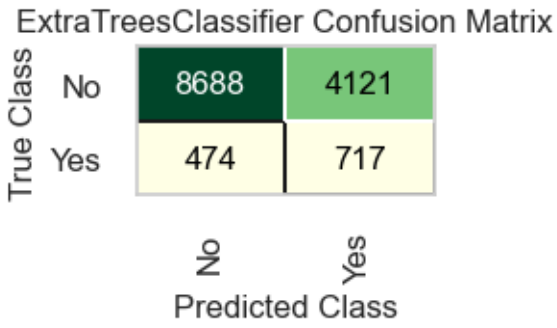



```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.70      0.81     12809
         Yes       0.17      0.65      0.26      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.65892857 0.68178571 0.68       0.68       0.68642857]
Accuracy score (average): 0.6774285714285715
F1 scores for 'Yes' values: [0.23782921 0.2338779  0.24705882 0.23287671
0.2417962 ]
Average F1 score: 0.535
ROC-AUC score: 0.674
```

PR-AUC score: 0.138
Log-loss: 0.614

Execution time: 58.41139578819275

ExtraTreesClassifier



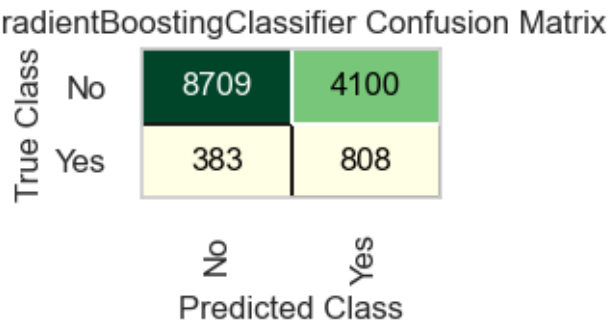
Classification Report:

	precision	recall	f1-score	support
No	0.95	0.68	0.79	12809
Yes	0.15	0.60	0.24	1191
accuracy			0.67	14000
macro avg	0.55	0.64	0.51	14000
weighted avg	0.88	0.67	0.74	14000

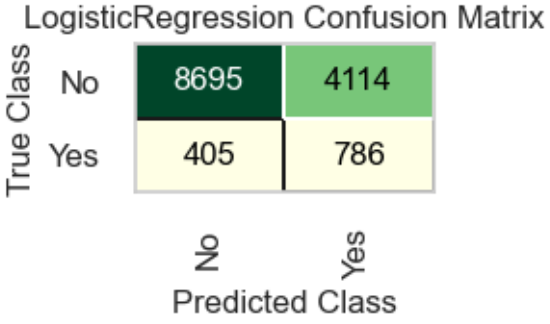
Cross-validation
Accuracy scores: [0.65857143 0.65892857 0.65392857 0.67607143 0.65428571]
Accuracy score (average): 0.6603571428571429
F1 scores for 'Yes' values: [0.22778675 0.21913328 0.22417934 0.22544833
0.22061192]
Average F1 score: 0.514
ROC-AUC score: 0.64
PR-AUC score: 0.123
Log-loss: 0.625

Execution time: 58.527832984924316

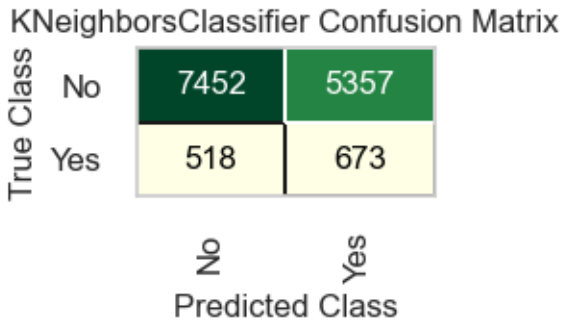
GradientBoostingClassifier




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     12809
         Yes       0.16      0.68      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.66571429 0.68357143 0.70178571 0.69035714 0.68035714]
Accuracy score (average): 0.6843571428571428
F1 scores for 'Yes' values: [0.2512     0.24273504 0.25246195 0.24543081
0.24726661]
Average F1 score: 0.53
ROC-AUC score: 0.679
PR-AUC score: 0.139
Log-loss: 0.599


Execution time: 64.75929999351501

LogisticRegression
```

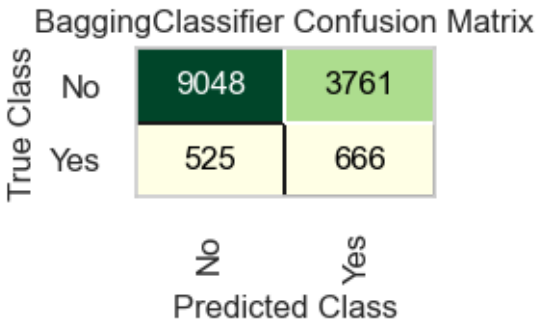


```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.79     12809
         Yes       0.16      0.66      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.66321429 0.67964286 0.67964286 0.69821429 0.66857143]
Accuracy score (average): 0.6778571428571428
F1 scores for 'Yes' values: [0.24257028 0.2531224  0.24558452 0.24079066
0.23558484]
Average F1 score: 0.526
ROC-AUC score: 0.669
PR-AUC score: 0.135
Log-loss: 0.625


Execution time: 58.36784625053406

KNeighborsClassifier
```



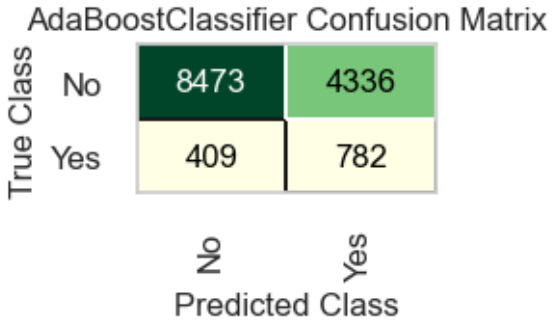

```
Classification Report:
              precision    recall  f1-score   support

          No       0.94      0.58      0.72     12809
         Yes       0.11      0.57      0.19      1191

    accuracy                           0.58     14000
```

```
   macro avg       0.52      0.57      0.45     14000
weighted avg       0.86      0.58      0.67     14000

Cross-validation
Accuracy scores: [0.53       0.54607143 0.54535714 0.56107143 0.56928571]
Accuracy score (average): 0.5503571428571429
F1 scores for 'Yes' values: [0.1754386  0.1654629  0.16194865 0.15879535
0.15899582]
Average F1 score: 0.452
ROC-AUC score: 0.573
PR-AUC score: 0.1
Log-loss: 1.736


Execution time: 57.74232506752014

BaggingClassifier
```



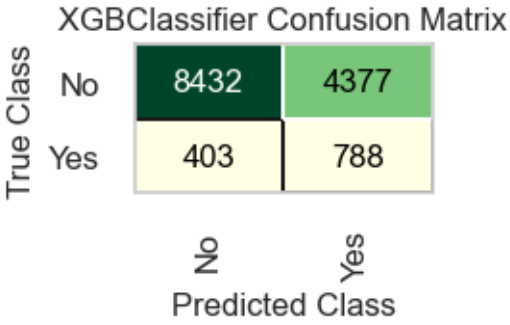

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.71      0.81     12809
         Yes       0.15      0.56      0.24      1191

    accuracy                           0.69     14000
   macro avg       0.55      0.63      0.52     14000
weighted avg       0.88      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.67714286 0.68857143 0.6825     0.695      0.71142857]
Accuracy score (average): 0.6909285714285713
F1 scores for 'Yes' values: [0.22068966 0.20727273 0.22493461 0.22363636
0.23484848]
Average F1 score: 0.523
ROC-AUC score: 0.633
```

```
PR-AUC score: 0.122
Log-loss: 0.942


Execution time: 59.1973180770874

AdaBoostClassifier
```



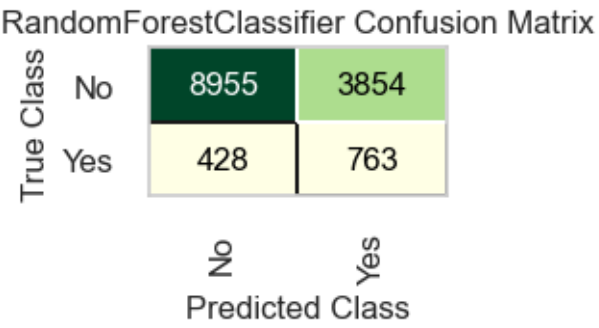

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.66      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.51     14000
weighted avg       0.89      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.63642857 0.685      0.65714286 0.6825     0.6725    ]
Accuracy score (average): 0.6667142857142857
F1 scores for 'Yes' values: [0.22878788 0.23965517 0.23688394 0.22762815
0.24402308]
Average F1 score: 0.515
ROC-AUC score: 0.659
PR-AUC score: 0.13
Log-loss: 0.689


Execution time: 58.820992946624756

Parameters for the dataset and transformers: WOEEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
```

XGBClassifier

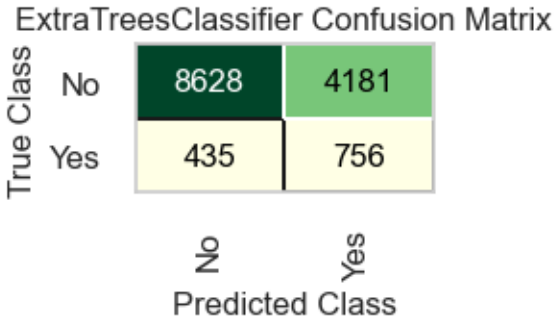



```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.66      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.51     14000
weighted avg       0.89      0.66      0.73     14000

Cross-validation
Accuracy scores: [0.6575     0.6725     0.67535714 0.66142857 0.66107143]
Accuracy score (average): 0.6655714285714286
F1 scores for 'Yes' values: [0.24189723 0.23005877 0.24186822 0.22801303
0.22908205]
Average F1 score: 0.514
ROC-AUC score: 0.66
PR-AUC score: 0.13
Log-loss: 0.743


Execution time: 52.551705837249756
```

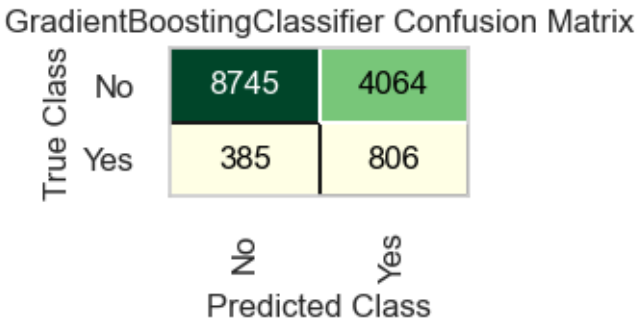
RandomForestClassifier




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.70      0.81     12809
         Yes       0.17      0.64      0.26      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.66285714 0.67428571 0.68535714 0.68642857 0.66678571]
Accuracy score (average): 0.675142857142857
F1 scores for 'Yes' values: [0.24358974 0.2284264  0.24377682 0.228471
0.23461854]
Average F1 score: 0.535
ROC-AUC score: 0.67
PR-AUC score: 0.136
Log-loss: 0.612


Execution time: 51.995522260665894

ExtraTreesClassifier
```

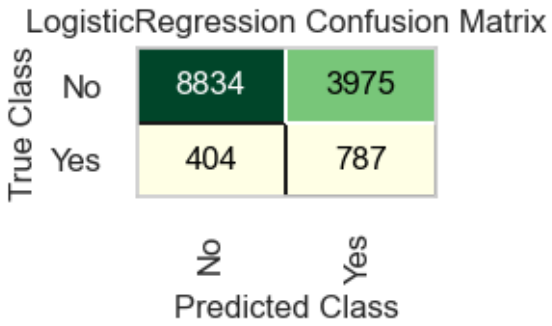


```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.67      0.79     12809
         Yes       0.15      0.63      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.55      0.65      0.52     14000
weighted avg       0.88      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65821429 0.655      0.65178571 0.67       0.67035714]
Accuracy score (average): 0.6610714285714285
F1 scores for 'Yes' values: [0.23378703 0.22096774 0.23046567 0.21694915
0.23275145]
Average F1 score: 0.518
ROC-AUC score: 0.654
PR-AUC score: 0.128
Log-loss: 0.627


Execution time: 51.88545489311218

GradientBoostingClassifier
```



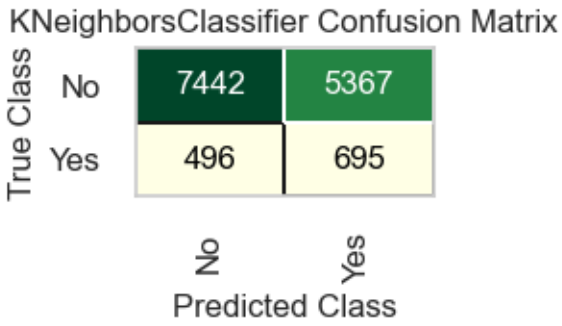

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     12809
         Yes       0.17      0.68      0.27      1191

    accuracy                           0.68     14000
```

```
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.65785714 0.69928571 0.69928571 0.69035714 0.69035714]
Accuracy score (average): 0.6874285714285714
F1 scores for 'Yes' values: [0.2515625  0.26398601 0.25486726 0.24279476
0.25451419]
Average F1 score: 0.532
ROC-AUC score: 0.68
PR-AUC score: 0.14
Log-loss: 0.601


Execution time: 58.42783999443054

LogisticRegression
```

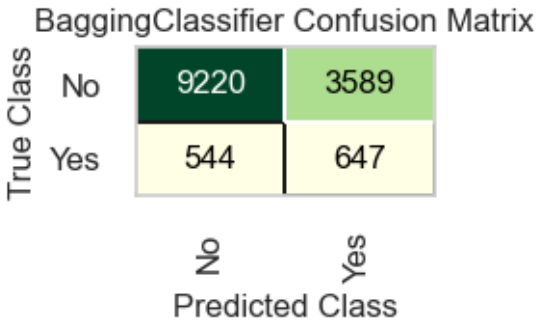



```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.69      0.80     12809
         Yes       0.17      0.66      0.26      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.66928571 0.69357143 0.69       0.69607143 0.66964286]
Accuracy score (average): 0.6837142857142857
F1 scores for 'Yes' values: [0.25322581 0.25520833 0.25557461 0.24623561
0.23363712]
Average F1 score: 0.533
ROC-AUC score: 0.675
```

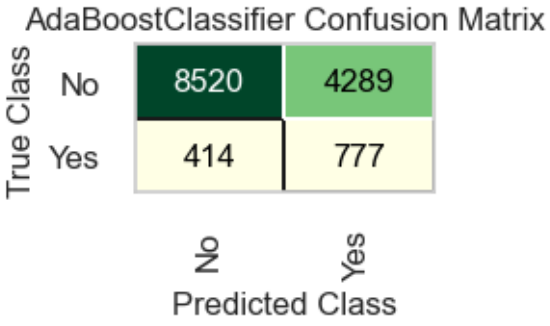
```
PR-AUC score: 0.138
Log-loss: 0.61


Execution time: 51.07777118682861

KNeighborsClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.94      0.58      0.72     12809
         Yes       0.11      0.58      0.19      1191

    accuracy                           0.58     14000
   macro avg       0.53      0.58      0.45     14000
weighted avg       0.87      0.58      0.67     14000

Cross-validation
Accuracy scores: [0.56785714 0.52714286 0.55321429 0.54571429 0.57714286]
Accuracy score (average): 0.5542142857142858
F1 scores for 'Yes' values: [0.17462483 0.15776081 0.17643186 0.16205534
0.15669516]
Average F1 score: 0.455
ROC-AUC score: 0.582
PR-AUC score: 0.102
Log-loss: 1.726


Execution time: 50.99345803260803

BaggingClassifier
```



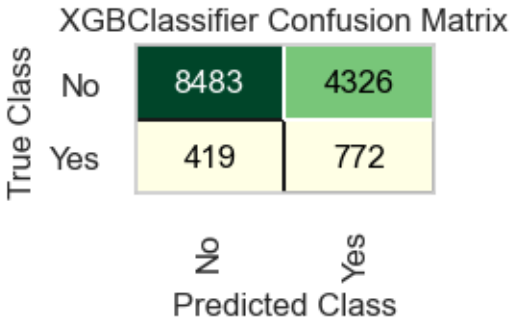

```
Classification Report:
              precision    recall  f1-score   support

          No       0.94      0.72      0.82     12809
         Yes       0.15      0.54      0.24      1191

    accuracy                           0.70     14000
   macro avg       0.55      0.63      0.53     14000
weighted avg       0.88      0.70      0.77     14000

Cross-validation
Accuracy scores: [0.68107143 0.70428571 0.71678571 0.71642857 0.69892857]
Accuracy score (average): 0.7035
F1 scores for 'Yes' values: [0.22144725 0.2247191  0.23233301 0.23359073
0.21288515]
Average F1 score: 0.528
ROC-AUC score: 0.632
PR-AUC score: 0.122
Log-loss: 0.936


Execution time: 52.09188723564148

AdaBoostClassifier
```



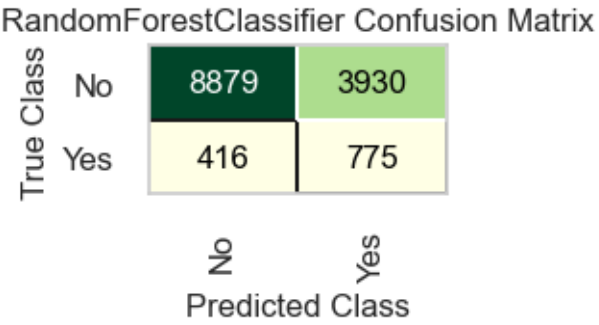
```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.67      0.78     12809
         Yes       0.15      0.65      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.52     14000
weighted avg       0.89      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.63392857 0.67821429 0.65928571 0.68285714 0.66392857]
Accuracy score (average): 0.6636428571428572
F1 scores for 'Yes' values: [0.21934501 0.23837701 0.22439024 0.23448276
0.23183673]
Average F1 score: 0.516
ROC-AUC score: 0.659
PR-AUC score: 0.13
Log-loss: 0.688


Execution time: 52.365379095077515
```

Next, models are trained on a limited number of features (150). Features are selected by step ('selectKBest', SelectKBest(score_func=mutual_info_classif, k=features)) in the pipeline.

```
Parameters for the dataset and transformers: OneHotEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
XGBClassifier
```

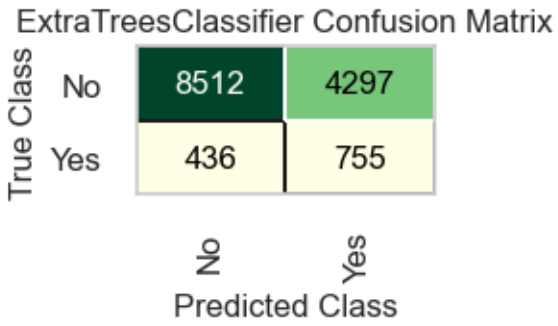


```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.65      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.51     14000
weighted avg       0.88      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.64571429 0.65642857 0.65964286 0.66714286 0.66571429]
Accuracy score (average): 0.6589285714285713
F1 scores for 'Yes' values: [0.23338485 0.22916667 0.23453815 0.21150592
0.24271845]
Average F1 score: 0.513
ROC-AUC score: 0.655
PR-AUC score: 0.128
Log-loss: 0.734


Execution time: 57.58500599861145

RandomForestClassifier
```



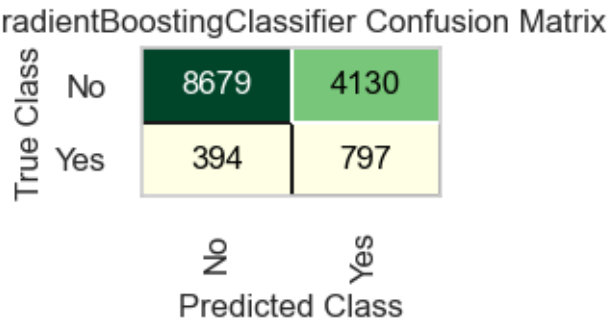

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.69      0.80     12809
         Yes       0.16      0.65      0.26      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.69      0.76     14000
```

```
Cross-validation
Accuracy scores: [0.66321429 0.68714286 0.68892857 0.69142857 0.67821429]
Accuracy score (average): 0.6817857142857142
F1 scores for 'Yes' values: [0.25099285 0.24090121 0.24457936 0.23674912
0.24476111]
Average F1 score: 0.533
ROC-AUC score: 0.672
PR-AUC score: 0.137
Log-loss: 0.609


Execution time: 58.03208017349243

ExtraTreesClassifier
```

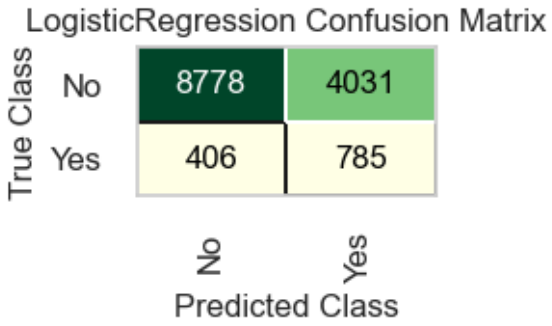



```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.63      0.24      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.65      0.51     14000
weighted avg       0.88      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.65571429 0.64821429 0.66357143 0.68357143 0.66178571]
Accuracy score (average): 0.6625714285714286
F1 scores for 'Yes' values: [0.23492063 0.21638823 0.23909532 0.23090278
0.21929101]
Average F1 score: 0.512
ROC-AUC score: 0.649
PR-AUC score: 0.126
Log-loss: 0.63
```

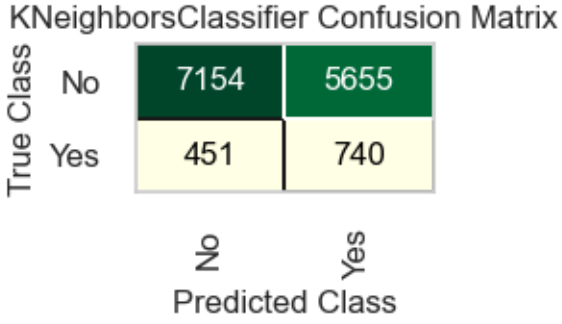
```
Execution time: 57.81312084197998

GradientBoostingClassifier
```



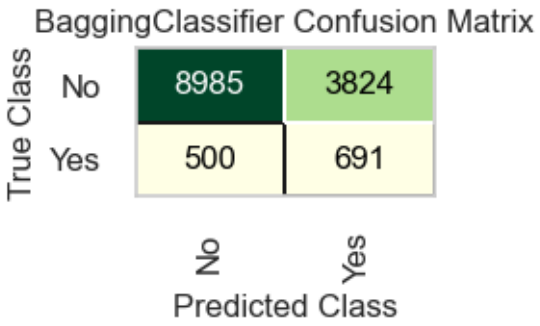

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.79     12809
         Yes       0.16      0.67      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.65714286 0.6975     0.69785714 0.67964286 0.66678571]
Accuracy score (average): 0.6797857142857143
F1 scores for 'Yes' values: [0.24528302 0.25242718 0.24733096 0.23659574
0.23461854]
Average F1 score: 0.527
ROC-AUC score: 0.673
PR-AUC score: 0.136
Log-loss: 0.604


Execution time: 515.3310532569885

LogisticRegression
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.69      0.80     12809
         Yes       0.16      0.66      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.67321429 0.68785714 0.67607143 0.70214286 0.68928571]
Accuracy score (average): 0.6857142857142856
F1 scores for 'Yes' values: [0.25910931 0.25171233 0.23845508 0.24456522
0.23684211]
Average F1 score: 0.53
ROC-AUC score: 0.672
PR-AUC score: 0.136
Log-loss: 0.604


Execution time: 57.79024791717529

KNeighborsClassifier
```

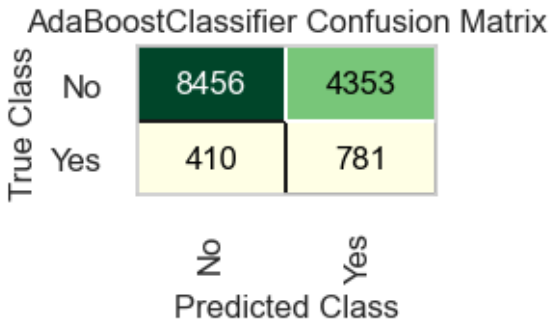


```
Classification Report:
              precision    recall  f1-score   support

          No       0.94      0.56      0.70     12809
         Yes       0.12      0.62      0.20      1191

    accuracy                           0.56     14000
   macro avg       0.53      0.59      0.45     14000
weighted avg       0.87      0.56      0.66     14000

Cross-validation
Accuracy scores: [0.555      0.56142857 0.54142857 0.53285714 0.55035714]
Accuracy score (average): 0.5482142857142858
F1 scores for 'Yes' values: [0.15810811 0.16348774 0.16297262 0.1462141
0.16567263]
Average F1 score: 0.448
ROC-AUC score: 0.59
PR-AUC score: 0.104
Log-loss: 2.035


Execution time: 57.606423139572144

BaggingClassifier
```



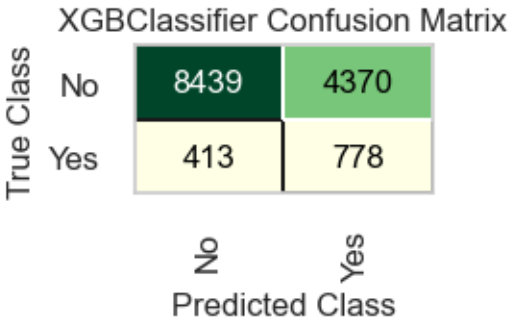

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.70      0.81     12809
         Yes       0.15      0.58      0.24      1191

    accuracy                           0.69     14000
```

```
   macro avg       0.55      0.64      0.52     14000
weighted avg       0.88      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.66607143 0.695      0.69607143 0.69714286 0.7075    ]
Accuracy score (average): 0.6923571428571428
F1 scores for 'Yes' values: [0.23297785 0.22080292 0.21711132 0.22344322
0.23813953]
Average F1 score: 0.524
ROC-AUC score: 0.641
PR-AUC score: 0.125
Log-loss: 1.013


Execution time: 58.07448720932007

AdaBoostClassifier
```



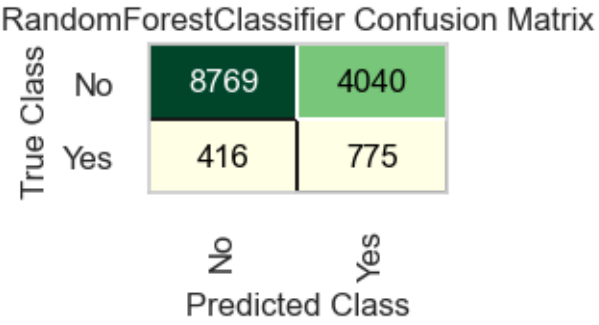

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.66      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.51     14000
weighted avg       0.89      0.66      0.73     14000

Cross-validation
Accuracy scores: [0.65714286 0.67607143 0.66214286 0.66857143 0.66857143]
Accuracy score (average): 0.6665
F1 scores for 'Yes' values: [0.24409449 0.23845508 0.22838499 0.23178808
0.23305785]
Average F1 score: 0.514
ROC-AUC score: 0.658
```

```
PR-AUC score: 0.129
Log-loss: 0.689


Execution time: 64.80393409729004

Parameters for the dataset and transformers: WOEEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
XGBClassifier
```

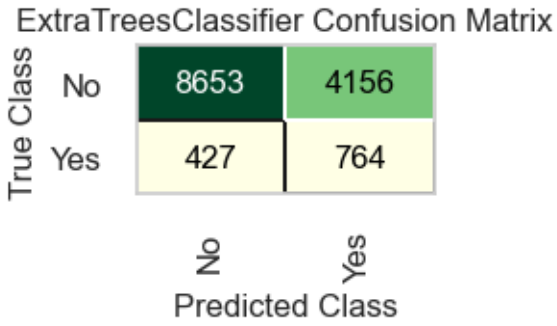



```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.65      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.51     14000
weighted avg       0.89      0.66      0.73     14000

Cross-validation
Accuracy scores: [0.65821429 0.66714286 0.66392857 0.66107143 0.65571429]
Accuracy score (average): 0.6612142857142856
F1 scores for 'Yes' values: [0.24107851 0.22975207 0.23308883 0.22149303
0.23248408]
Average F1 score: 0.512
ROC-AUC score: 0.656
PR-AUC score: 0.128
Log-loss: 0.747


Execution time: 53.73718500137329
```

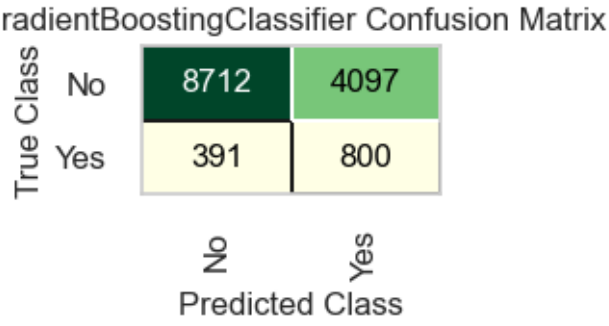
RandomForestClassifier



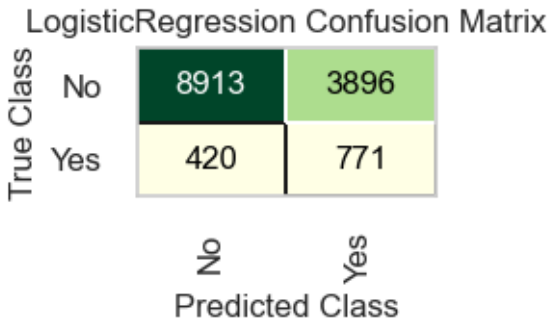

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.68      0.80     12809
         Yes       0.16      0.65      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.67071429 0.69178571 0.675      0.68785714 0.68214286]
Accuracy score (average): 0.6815
F1 scores for 'Yes' values: [0.24673203 0.24496938 0.23141892 0.24
0.24190801]
Average F1 score: 0.528
ROC-AUC score: 0.668
PR-AUC score: 0.134
Log-loss: 0.615


Execution time: 52.83474087715149

ExtraTreesClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.68      0.79     12809
         Yes       0.16      0.64      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.55      0.66      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65607143 0.67       0.66857143 0.67642857 0.65357143]
Accuracy score (average): 0.6649285714285714
F1 scores for 'Yes' values: [0.23993686 0.23762376 0.23432343 0.22959184
0.23015873]
Average F1 score: 0.52
ROC-AUC score: 0.659
PR-AUC score: 0.13
Log-loss: 0.621


Execution time: 52.443227767944336

GradientBoostingClassifier
```

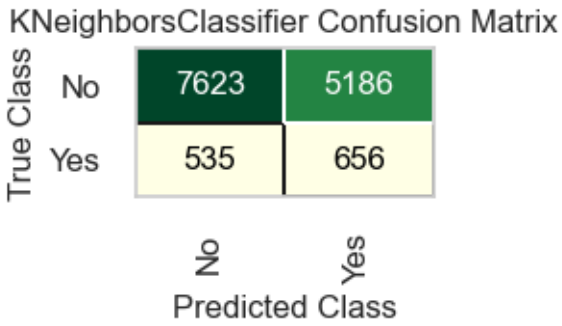


```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     12809
         Yes       0.16      0.67      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.66607143 0.69107143 0.695      0.69214286 0.69464286]
Accuracy score (average): 0.6877857142857142
F1 scores for 'Yes' values: [0.25852498 0.24454148 0.24424779 0.23172906
0.24802111]
Average F1 score: 0.529
ROC-AUC score: 0.676
PR-AUC score: 0.138
Log-loss: 0.605


Execution time: 56.264875173568726

LogisticRegression
```



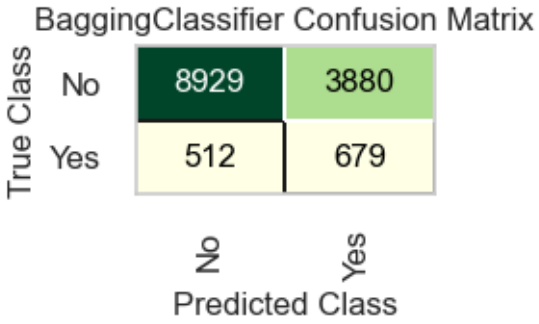

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.70      0.81     12809
         Yes       0.17      0.65      0.26      1191

    accuracy                           0.69     14000
```

```
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.67857143 0.68714286 0.69178571 0.70142857 0.67535714]
Accuracy score (average): 0.6868571428571429
F1 scores for 'Yes' values: [0.26829268 0.25383305 0.25667528 0.24955117
0.23031329]
Average F1 score: 0.534
ROC-AUC score: 0.672
PR-AUC score: 0.137
Log-loss: 0.608


Execution time: 51.36406493186951

KNeighborsClassifier
```

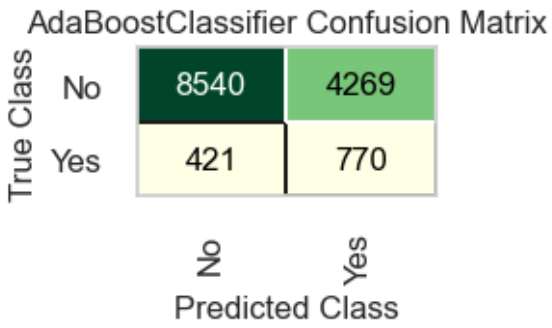



```
Classification Report:
              precision    recall  f1-score   support

          No       0.93      0.60      0.73     12809
         Yes       0.11      0.55      0.19      1191

    accuracy                           0.59     14000
   macro avg       0.52      0.57      0.46     14000
weighted avg       0.86      0.59      0.68     14000

Cross-validation
Accuracy scores: [0.54964286 0.58035714 0.5575     0.585      0.59857143]
Accuracy score (average): 0.5742142857142858
F1 scores for 'Yes' values: [0.17202889 0.17078335 0.16       0.16161616
0.17956204]
Average F1 score: 0.457
ROC-AUC score: 0.573
```

```
PR-AUC score: 0.1
Log-loss: 1.639


Execution time: 51.14260387420654

BaggingClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.70      0.80     12809
         Yes       0.15      0.57      0.24      1191

    accuracy                           0.69     14000
   macro avg       0.55      0.63      0.52     14000
weighted avg       0.88      0.69      0.75     14000

Cross-validation
Accuracy scores: [0.67321429 0.695      0.69321429 0.68392857 0.69964286]
Accuracy score (average): 0.689
F1 scores for 'Yes' values: [0.23430962 0.21072089 0.20389249 0.20911528
0.21767442]
Average F1 score: 0.519
ROC-AUC score: 0.634
PR-AUC score: 0.121
Log-loss: 1.024


Execution time: 51.60221004486084

AdaBoostClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.67      0.78     12809
         Yes       0.15      0.65      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.55      0.66      0.52     14000
weighted avg       0.88      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65607143 0.68107143 0.67142857 0.66071429 0.67214286]
Accuracy score (average): 0.6682857142857143
F1 scores for 'Yes' values: [0.24113475 0.23740393 0.23333333 0.22258592
0.23880597]
Average F1 score: 0.516
ROC-AUC score: 0.657
PR-AUC score: 0.129
Log-loss: 0.689


Execution time: 52.7132363319397
```

Metrics of different models can be observed in the dataframes 'scores1' (appended with the output data from the first run) and 'scores2' (from the second run) and in the bar plots.

[156]:

model	index	precision_score	recall_score	\
GradientBoostingClassifier 11	11	0.562	0.680	
GradientBoostingClassifier 3	3	0.561	0.679	
LogisticRegression 12	12	0.561	0.675	
RandomForestClassifier 1	1	0.561	0.674	
RandomForestClassifier 9	9	0.560	0.670	
LogisticRegression 4	4	0.558	0.669	

XGBClassifier 0	0	0.555	0.662
XGBClassifier 8	8	0.553	0.660
AdaBoostClassifier 7	7	0.553	0.659
AdaBoostClassifier 15	15	0.554	0.659
ExtraTreesClassifier 10	10	0.553	0.654
ExtraTreesClassifier 2	2	0.548	0.640
BaggingClassifier 6	6	0.548	0.633
BaggingClassifier 14	14	0.549	0.632
KNeighborsClassifier 13	13	0.526	0.582
KNeighborsClassifier 5	5	0.523	0.573

model	model_name	a_score	f1_score	\
GradientBoostingClassifier 11	GradientBoostingClassifier	0.682	0.532	
GradientBoostingClassifier 3	GradientBoostingClassifier	0.680	0.530	
LogisticRegression 12	LogisticRegression	0.687	0.533	
RandomForestClassifier 1	RandomForestClassifier	0.692	0.535	
RandomForestClassifier 9	RandomForestClassifier	0.694	0.535	
LogisticRegression 4	LogisticRegression	0.677	0.526	
XGBClassifier 0	XGBClassifier	0.669	0.519	
XGBClassifier 8	XGBClassifier	0.659	0.514	
AdaBoostClassifier 7	AdaBoostClassifier	0.661	0.515	
AdaBoostClassifier 15	AdaBoostClassifier	0.664	0.516	
ExtraTreesClassifier 10	ExtraTreesClassifier	0.670	0.518	
ExtraTreesClassifier 2	ExtraTreesClassifier	0.672	0.514	
BaggingClassifier 6	BaggingClassifier	0.694	0.523	
BaggingClassifier 14	BaggingClassifier	0.705	0.528	
KNeighborsClassifier 13	KNeighborsClassifier	0.581	0.455	
KNeighborsClassifier 5	KNeighborsClassifier	0.580	0.452	

model	ROC_AUC_score	PR_AUC_score	loss	exec_time	\
GradientBoostingClassifier 11	0.680	0.140	0.601	58.427840	
GradientBoostingClassifier 3	0.679	0.139	0.599	64.759300	
LogisticRegression 12	0.675	0.138	0.610	51.077771	
RandomForestClassifier 1	0.674	0.138	0.614	58.411396	
RandomForestClassifier 9	0.670	0.136	0.612	51.995522	
LogisticRegression 4	0.669	0.135	0.625	58.367846	
XGBClassifier 0	0.662	0.131	0.726	59.361831	
XGBClassifier 8	0.660	0.130	0.743	52.551706	
AdaBoostClassifier 7	0.659	0.130	0.689	58.820993	
AdaBoostClassifier 15	0.659	0.130	0.688	52.365379	
ExtraTreesClassifier 10	0.654	0.128	0.627	51.885455	
ExtraTreesClassifier 2	0.640	0.123	0.625	58.527833	
BaggingClassifier 6	0.633	0.122	0.942	59.197318	
BaggingClassifier 14	0.632	0.122	0.936	52.091887	
KNeighborsClassifier 13	0.582	0.102	1.726	50.993458	

```
KNeighborsClassifier 5                0.573         0.100  1.736  57.742325

                                    encoders      cimputers      nimputers  \
model
GradientBoostingClassifier 11     WOEEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 3   OneHotEncoder  SimpleImputer  SimpleImputer
LogisticRegression 12             WOEEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 1       OneHotEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 9          WOEEncoder  SimpleImputer  SimpleImputer
LogisticRegression 4           OneHotEncoder  SimpleImputer  SimpleImputer
XGBClassifier 0                OneHotEncoder  SimpleImputer  SimpleImputer
XGBClassifier 8                   WOEEncoder  SimpleImputer  SimpleImputer
AdaBoostClassifier 7           OneHotEncoder  SimpleImputer  SimpleImputer
AdaBoostClassifier 15             WOEEncoder  SimpleImputer  SimpleImputer
ExtraTreesClassifier 10           WOEEncoder  SimpleImputer  SimpleImputer
ExtraTreesClassifier 2         OneHotEncoder  SimpleImputer  SimpleImputer
BaggingClassifier 6            OneHotEncoder  SimpleImputer  SimpleImputer
BaggingClassifier 14              WOEEncoder  SimpleImputer  SimpleImputer
KNeighborsClassifier 13           WOEEncoder  SimpleImputer  SimpleImputer
KNeighborsClassifier 5         OneHotEncoder  SimpleImputer  SimpleImputer

                                     scalers        num_features  \
model
GradientBoostingClassifier 11  StandardScaler  numeric_features_1
GradientBoostingClassifier 3   StandardScaler  numeric_features_1
LogisticRegression 12          StandardScaler  numeric_features_1
RandomForestClassifier 1       StandardScaler  numeric_features_1
RandomForestClassifier 9       StandardScaler  numeric_features_1
LogisticRegression 4           StandardScaler  numeric_features_1
XGBClassifier 0                StandardScaler  numeric_features_1
XGBClassifier 8                StandardScaler  numeric_features_1
AdaBoostClassifier 7           StandardScaler  numeric_features_1
AdaBoostClassifier 15          StandardScaler  numeric_features_1
ExtraTreesClassifier 10        StandardScaler  numeric_features_1
ExtraTreesClassifier 2         StandardScaler  numeric_features_1
BaggingClassifier 6            StandardScaler  numeric_features_1
BaggingClassifier 14           StandardScaler  numeric_features_1
KNeighborsClassifier 13        StandardScaler  numeric_features_1
KNeighborsClassifier 5         StandardScaler  numeric_features_1

                                          cat_features       bin_features  \
model
GradientBoostingClassifier 11  categorical_features_1  binary_features_1
GradientBoostingClassifier 3   categorical_features_1  binary_features_1
LogisticRegression 12          categorical_features_1  binary_features_1
RandomForestClassifier 1       categorical_features_1  binary_features_1
RandomForestClassifier 9       categorical_features_1  binary_features_1
```

```
LogisticRegression 4          categorical_features_1  binary_features_1
XGBClassifier 0               categorical_features_1  binary_features_1
XGBClassifier 8               categorical_features_1  binary_features_1
AdaBoostClassifier 7          categorical_features_1  binary_features_1
AdaBoostClassifier 15         categorical_features_1  binary_features_1
ExtraTreesClassifier 10       categorical_features_1  binary_features_1
ExtraTreesClassifier 2        categorical_features_1  binary_features_1
BaggingClassifier 6           categorical_features_1  binary_features_1
BaggingClassifier 14          categorical_features_1  binary_features_1
KNeighborsClassifier 13       categorical_features_1  binary_features_1
KNeighborsClassifier 5        categorical_features_1  binary_features_1

                                 other_features
model
GradientBoostingClassifier 11  other_features_1
GradientBoostingClassifier 3   other_features_1
LogisticRegression 12          other_features_1
RandomForestClassifier 1       other_features_1
RandomForestClassifier 9       other_features_1
LogisticRegression 4           other_features_1
XGBClassifier 0                other_features_1
XGBClassifier 8                other_features_1
AdaBoostClassifier 7           other_features_1
AdaBoostClassifier 15          other_features_1
ExtraTreesClassifier 10        other_features_1
ExtraTreesClassifier 2         other_features_1
BaggingClassifier 6            other_features_1
BaggingClassifier 14           other_features_1
KNeighborsClassifier 13        other_features_1
KNeighborsClassifier 5         other_features_1
```

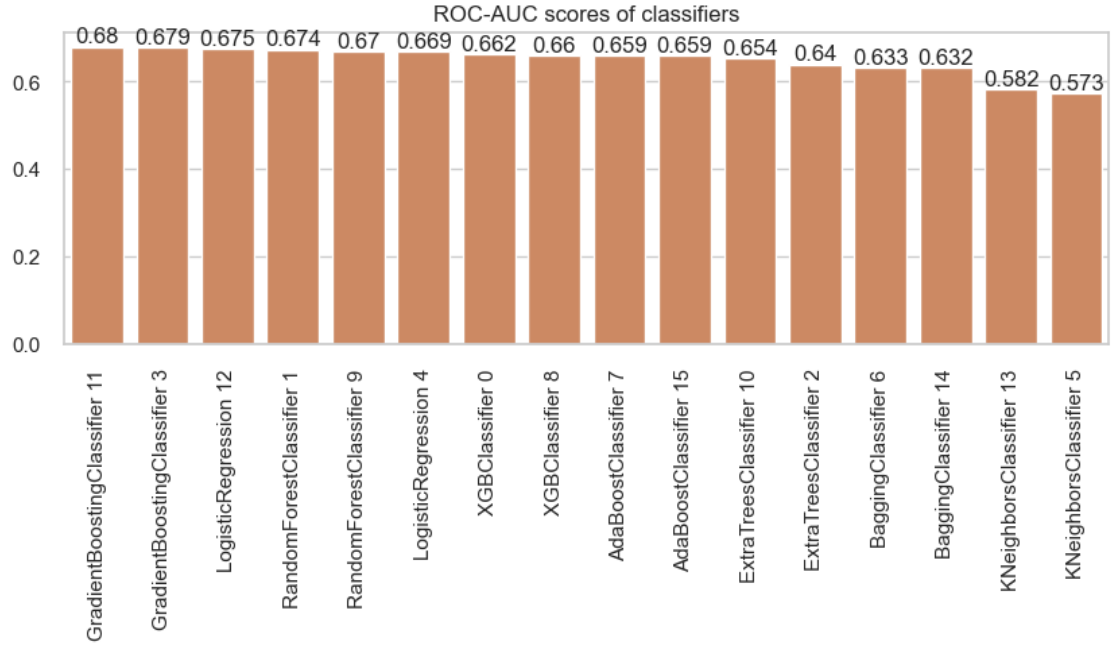
[158]:

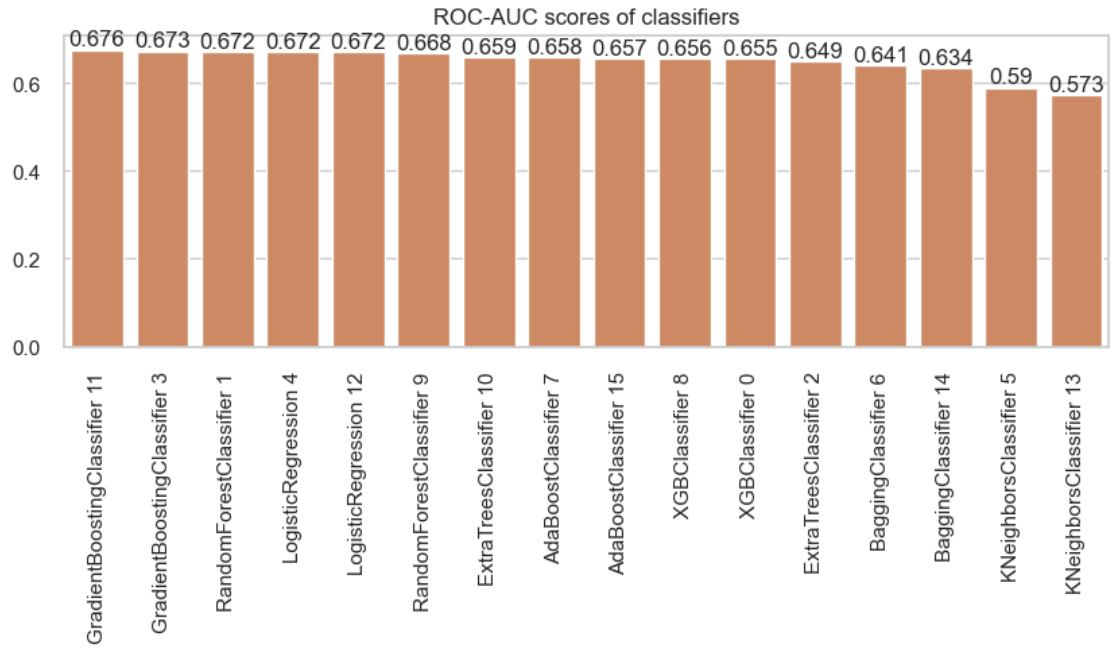
model	index	precision_score	recall_score	\
GradientBoostingClassifier 11	11	0.560	0.676	
GradientBoostingClassifier 3	3	0.559	0.673	
RandomForestClassifier 1	1	0.560	0.672	
LogisticRegression 4	4	0.559	0.672	
LogisticRegression 12	12	0.560	0.672	
RandomForestClassifier 9	9	0.558	0.668	
ExtraTreesClassifier 10	10	0.554	0.659	
AdaBoostClassifier 7	7	0.553	0.658	
AdaBoostClassifier 15	15	0.553	0.657	
XGBClassifier 8	8	0.552	0.656	
XGBClassifier 0	0	0.552	0.655	
ExtraTreesClassifier 2	2	0.550	0.649	
BaggingClassifier 6	6	0.550	0.641	
BaggingClassifier 14	14	0.547	0.634	

```
KNeighborsClassifier 5             5            0.528         0.590
KNeighborsClassifier 13           13            0.523         0.573

                                                 model_name  a_score  f1_score  \
model
GradientBoostingClassifier 11  GradientBoostingClassifier    0.679     0.529
GradientBoostingClassifier 3   GradientBoostingClassifier    0.677     0.527
RandomForestClassifier 1           RandomForestClassifier    0.690     0.533
LogisticRegression 4                   LogisticRegression    0.683     0.530
LogisticRegression 12                  LogisticRegression    0.692     0.534
RandomForestClassifier 9           RandomForestClassifier    0.682     0.528
ExtraTreesClassifier 10              ExtraTreesClassifier    0.673     0.520
AdaBoostClassifier 7                   AdaBoostClassifier    0.660     0.514
AdaBoostClassifier 15                  AdaBoostClassifier    0.665     0.516
XGBClassifier 8                             XGBClassifier    0.658     0.512
XGBClassifier 0                             XGBClassifier    0.661     0.513
ExtraTreesClassifier 2               ExtraTreesClassifier    0.662     0.512
BaggingClassifier 6                     BaggingClassifier    0.691     0.524
BaggingClassifier 14                    BaggingClassifier    0.686     0.519
KNeighborsClassifier 5               KNeighborsClassifier    0.564     0.448
KNeighborsClassifier 13              KNeighborsClassifier    0.591     0.457

                               ROC_AUC_score  PR_AUC_score   loss   exec_time  \
model
GradientBoostingClassifier 11          0.676         0.138  0.605   56.264875
GradientBoostingClassifier 3           0.673         0.136  0.604  515.331053
RandomForestClassifier 1               0.672         0.137  0.609   58.032080
LogisticRegression 4                   0.672         0.136  0.604   57.790248
LogisticRegression 12                  0.672         0.137  0.608   51.364065
RandomForestClassifier 9               0.668         0.134  0.615   52.834741
ExtraTreesClassifier 10                0.659         0.130  0.621   52.443228
AdaBoostClassifier 7                   0.658         0.129  0.689   64.803934
AdaBoostClassifier 15                  0.657         0.129  0.689   52.713236
XGBClassifier 8                        0.656         0.128  0.747   53.737185
XGBClassifier 0                        0.655         0.128  0.734   57.585006
ExtraTreesClassifier 2                 0.649         0.126  0.630   57.813121
BaggingClassifier 6                    0.641         0.125  1.013   58.074487
BaggingClassifier 14                   0.634         0.121  1.024   51.602210
KNeighborsClassifier 5                 0.590         0.104  2.035   57.606423
KNeighborsClassifier 13                0.573         0.100  1.639   51.142604

                                    encoders      cimputers      nimputers  \
model
GradientBoostingClassifier 11     WOEEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 3   OneHotEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 1       OneHotEncoder  SimpleImputer  SimpleImputer
LogisticRegression 4           OneHotEncoder  SimpleImputer  SimpleImputer
```

```
LogisticRegression 12             WOEEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 9          WOEEncoder  SimpleImputer  SimpleImputer
ExtraTreesClassifier 10           WOEEncoder  SimpleImputer  SimpleImputer
AdaBoostClassifier 7           OneHotEncoder  SimpleImputer  SimpleImputer
AdaBoostClassifier 15             WOEEncoder  SimpleImputer  SimpleImputer
XGBClassifier 8                   WOEEncoder  SimpleImputer  SimpleImputer
XGBClassifier 0                OneHotEncoder  SimpleImputer  SimpleImputer
ExtraTreesClassifier 2         OneHotEncoder  SimpleImputer  SimpleImputer
BaggingClassifier 6            OneHotEncoder  SimpleImputer  SimpleImputer
BaggingClassifier 14              WOEEncoder  SimpleImputer  SimpleImputer
KNeighborsClassifier 5         OneHotEncoder  SimpleImputer  SimpleImputer
KNeighborsClassifier 13           WOEEncoder  SimpleImputer  SimpleImputer

                                       scalers        num_features  \
model
GradientBoostingClassifier 11  StandardScaler  numeric_features_1
GradientBoostingClassifier 3   StandardScaler  numeric_features_1
RandomForestClassifier 1       StandardScaler  numeric_features_1
LogisticRegression 4           StandardScaler  numeric_features_1
LogisticRegression 12          StandardScaler  numeric_features_1
RandomForestClassifier 9       StandardScaler  numeric_features_1
ExtraTreesClassifier 10        StandardScaler  numeric_features_1
AdaBoostClassifier 7           StandardScaler  numeric_features_1
AdaBoostClassifier 15          StandardScaler  numeric_features_1
XGBClassifier 8                StandardScaler  numeric_features_1
XGBClassifier 0                StandardScaler  numeric_features_1
ExtraTreesClassifier 2         StandardScaler  numeric_features_1
BaggingClassifier 6            StandardScaler  numeric_features_1
BaggingClassifier 14           StandardScaler  numeric_features_1
KNeighborsClassifier 5         StandardScaler  numeric_features_1
KNeighborsClassifier 13        StandardScaler  numeric_features_1

                                          cat_features       bin_features  \
model
GradientBoostingClassifier 11  categorical_features_1  binary_features_1
GradientBoostingClassifier 3   categorical_features_1  binary_features_1
RandomForestClassifier 1       categorical_features_1  binary_features_1
LogisticRegression 4           categorical_features_1  binary_features_1
LogisticRegression 12          categorical_features_1  binary_features_1
RandomForestClassifier 9       categorical_features_1  binary_features_1
ExtraTreesClassifier 10        categorical_features_1  binary_features_1
AdaBoostClassifier 7           categorical_features_1  binary_features_1
AdaBoostClassifier 15          categorical_features_1  binary_features_1
XGBClassifier 8                categorical_features_1  binary_features_1
XGBClassifier 0                categorical_features_1  binary_features_1
ExtraTreesClassifier 2         categorical_features_1  binary_features_1
BaggingClassifier 6            categorical_features_1  binary_features_1
```

```
BaggingClassifier 14           categorical_features_1  binary_features_1
KNeighborsClassifier 5         categorical_features_1  binary_features_1
KNeighborsClassifier 13        categorical_features_1  binary_features_1

                                 other_features
model
GradientBoostingClassifier 11  other_features_1
GradientBoostingClassifier 3   other_features_1
RandomForestClassifier 1       other_features_1
LogisticRegression 4           other_features_1
LogisticRegression 12          other_features_1
RandomForestClassifier 9       other_features_1
ExtraTreesClassifier 10        other_features_1
AdaBoostClassifier 7           other_features_1
AdaBoostClassifier 15          other_features_1
XGBClassifier 8                other_features_1
XGBClassifier 0                other_features_1
ExtraTreesClassifier 2         other_features_1
BaggingClassifier 6            other_features_1
BaggingClassifier 14           other_features_1
KNeighborsClassifier 5         other_features_1
KNeighborsClassifier 13        other_features_1
```





It can be seen the best performance (the highest roc-auc score 0.68 which is higher that random guessing (score 0.5)) was achieved by the Gradient Boosting classifier with the WOE encoder and 315 features. Also, quite high scores for this classifier were achieved also with the combination of 150 features and one-hot encoder for categorical variables.

It can be observed that all classifiers, with exception of KNeighbors classifier which performed worse, were able to achieve quite similar performance (roc-auc score between 0.6 and 0.7).

The models predict the value 0 (clients who do not have payment difficulties) (max 1 score for "No" - 0.8) much better than the value 1 (persons with payment difficulties) (max f1 score for "Yes" - 0.26).

Bayesian optimization with the Optuna library Hyperparameter tuning of model parameter was conducted by using the Bayesian optimization (Optuna). Two functions were created - 'set_objective' function which sets parameters for hyperparameter tuning and 'run_optuna' function which runs Optuna's function 'objective' with Optuna's study in loops of lists of feature combinations and transformer pipelines. These functions are helpful when there is a need to run Bayesian optimization many times by trying different transformers or combinations of features.

The 'run_optuna' function was executed and it suggested the XGBoos classifier with the parameters (n_estimators: 1000, max_depth: 9, learning_rate: 0.024246395212299744, subsample: 0.8) as the most optimal, generating the highres roc-auc score - 0.692. It can be seen that this score is higher than any score of the previously trained classifiers.

```
Best ROC-AUC score by the Bayesian optimization (Optuna):  0.6922968992402028
Best parameters of classifiers:
  classifier: XGB
  n_estimators: 1000
```

```
  max_depth: 9
  learning_rate: 0.024246395212299744
  subsample: 0.8
Execution time for the Bayesian optimization (Optuna): 1372.8275692462921
```

Random feature selection The last feature selection approach which was used was the random feature selection. The function 'find_inputs' was created which randomly selects feature combinations from all features in the dataset in a loop for a high number of times, trains models on these combinations, calculates metrics and saves them in a dictionary (with feature combinations as keys and metrics as values); then it selects the feature combination which generated the highest score.

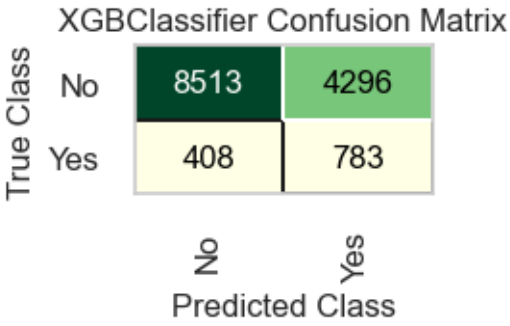
As the executing of this function is time consuming, the function was run on the Logistic regression classifier with certain parameters (the parameters were suggested in one of Optuna studies which outputs are not presented here).

```
Combinations of independent variables for the model with the highest roc-auc
scores:
[2, 3, 4, 5, 9, 14, 22, 23, 25, 29, 30, 31, 32, 33, 34, 35, 36, 40, 42, 43, 44,
45, 49, 51, 55, 58, 59, 62, 63, 66, 67, 68, 76, 79, 82, 84, 86, 92, 96, 98, 99,
102, 104, 105, 107, 114, 118, 121, 123, 124, 127, 129, 132, 133, 135, 139, 141,
145, 146, 147, 149, 153, 154, 157, 158, 159, 162, 164, 167, 169, 171, 176, 177,
178, 179, 185, 186, 188, 189, 190, 191, 193, 198, 199, 202, 210, 211, 213, 214,
216, 217, 219, 222, 223, 224, 225, 228, 230, 232, 234, 238, 241, 244, 245, 246,
247, 251, 252, 255, 258, 261, 263, 265, 266, 274, 275, 276, 278, 280, 282, 286,
288, 289, 290, 293, 297, 298, 300, 301, 302, 303, 304]
Roc-auc score:
0.687
```

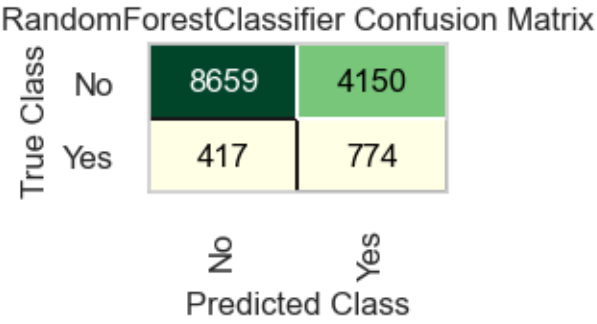
The lists of different types of features were created on the basis of the selected combination. Also, two new lists of classifiers were created - in one of them the best performing classifiers such as the Gradient Boosting, Random Forest and XGBoost (with parameters suggested by the Bayesian optimization) as well as the Logistic Regression classifier were included; in the second one, only the XGBoost classifier with suggested parameters was included.

The classifiers from the first list were trained on the feature combination suggested by the random feature selection. The XGBoost classifier with suggested parameters (in the second list) was also trained on all features (to observe if the choice of a number of features affects scores).

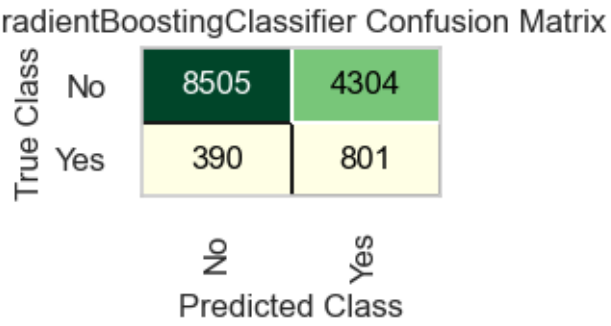
```
Parameters for the dataset and transformers: OneHotEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_5, binary_features_5,
categorical_features_5, other_features_5
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.66      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.52     14000
weighted avg       0.89      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.6425     0.65642857 0.6675     0.66928571 0.67285714]
Accuracy score (average): 0.6617142857142857
F1 scores for 'Yes' values: [0.24224073 0.23162939 0.22738589 0.22184874
0.23666667]
Average F1 score: 0.517
ROC-AUC score: 0.661
PR-AUC score: 0.13
Log-loss: 0.713


Execution time: 40.78691220283508

RandomForestClassifier
```

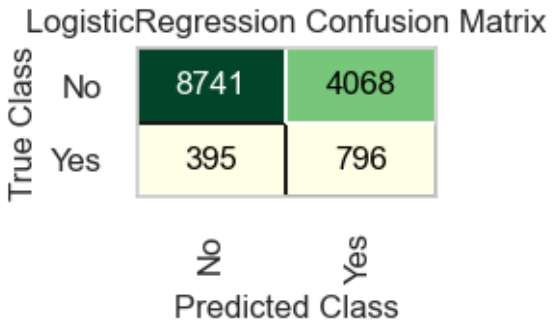


```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.68      0.79     12809
         Yes       0.16      0.65      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.56      0.66      0.52     14000
weighted avg       0.89      0.67      0.75     14000

Cross-validation
Accuracy scores: [0.65785714 0.67642857 0.67107143 0.69392857 0.67964286]
Accuracy score (average): 0.6757857142857142
F1 scores for 'Yes' values: [0.2444795  0.22827939 0.21617021 0.24890447
0.23398804]
Average F1 score: 0.522
ROC-AUC score: 0.663
PR-AUC score: 0.132
Log-loss: 0.62


Execution time: 22.84815216064453

GradientBoostingClassifier
```



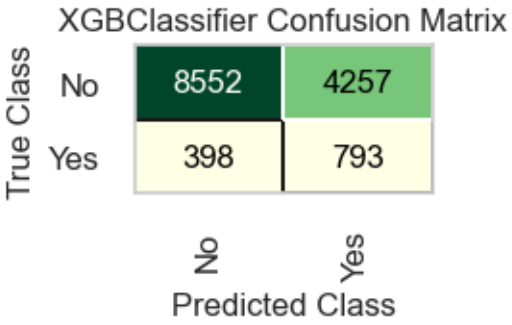

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.66      0.78     12809
         Yes       0.16      0.67      0.25      1191

    accuracy                           0.66     14000
```

```
   macro avg       0.56      0.67      0.52     14000
weighted avg       0.89      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.65714286 0.67857143 0.69321429 0.67857143 0.67964286]
Accuracy score (average): 0.6774285714285715
F1 scores for 'Yes' values: [0.25465839 0.23728814 0.23779947 0.23339012
0.22605695]
Average F1 score: 0.519
ROC-AUC score: 0.668
PR-AUC score: 0.133
Log-loss: 0.612


Execution time: 25.304349899291992

LogisticRegression
```



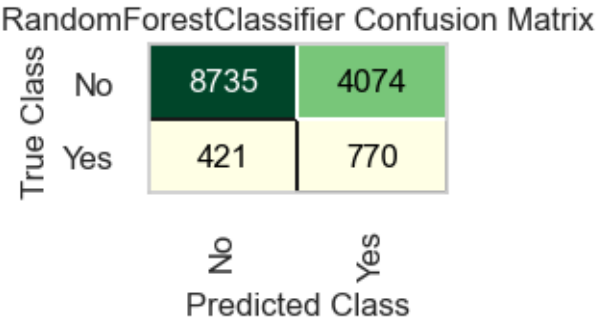

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     12809
         Yes       0.16      0.67      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.67857143 0.68142857 0.6825     0.7025     0.68392857]
Accuracy score (average): 0.6857857142857143
F1 scores for 'Yes' values: [0.25619835 0.24662162 0.24340426 0.24615385
0.23903697]
Average F1 score: 0.53
ROC-AUC score: 0.675
```

```
PR-AUC score: 0.138
Log-loss: 0.617


Execution time: 21.44376492500305

Parameters for the dataset and transformers: WOEEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_5, binary_features_5,
categorical_features_5, other_features_5
XGBClassifier
```

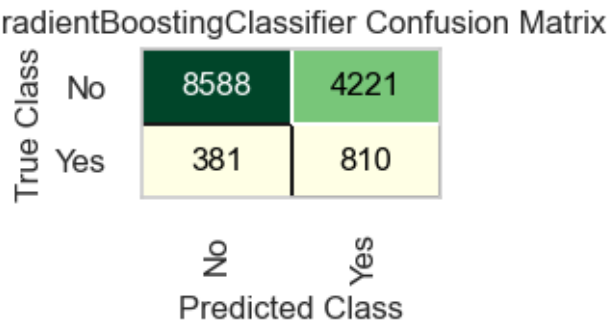



```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.67      0.79     12809
         Yes       0.16      0.67      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.56      0.67      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65178571 0.65428571 0.68392857 0.67357143 0.675     ]
Accuracy score (average): 0.6677142857142858
F1 scores for 'Yes' values: [0.24942263 0.21935484 0.24680851 0.22934233
0.23271501]
Average F1 score: 0.52
ROC-AUC score: 0.667
PR-AUC score: 0.133
Log-loss: 0.713


Execution time: 36.944725036621094
```

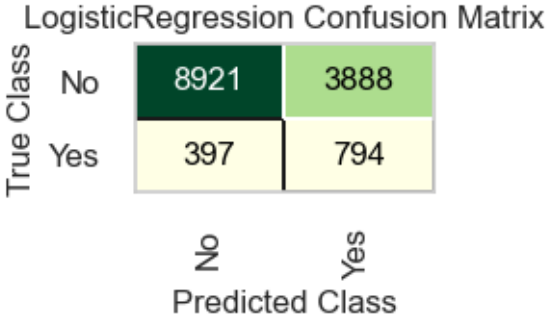
RandomForestClassifier



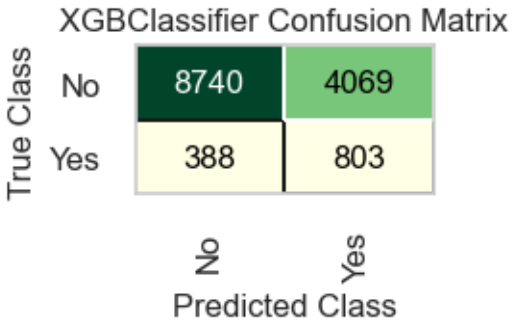

```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.68      0.80     12809
         Yes       0.16      0.65      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.66      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.66071429 0.68964286 0.67357143 0.69428571 0.69107143]
Accuracy score (average): 0.6818571428571428
F1 scores for 'Yes' values: [0.24363057 0.24631396 0.23450586 0.24381625
0.2432196 ]
Average F1 score: 0.525
ROC-AUC score: 0.664
PR-AUC score: 0.133
Log-loss: 0.617


Execution time: 18.920604944229126

GradientBoostingClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.67      0.79     12809
         Yes       0.16      0.68      0.26      1191

    accuracy                           0.67     14000
   macro avg       0.56      0.68      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.66821429 0.68678571 0.69964286 0.6825     0.68678571]
Accuracy score (average): 0.6847857142857142
F1 scores for 'Yes' values: [0.25857941 0.25740898 0.24438455 0.23690987
0.23805387]
Average F1 score: 0.525
ROC-AUC score: 0.675
PR-AUC score: 0.137
Log-loss: 0.611


Execution time: 21.79262900352478

LogisticRegression
```

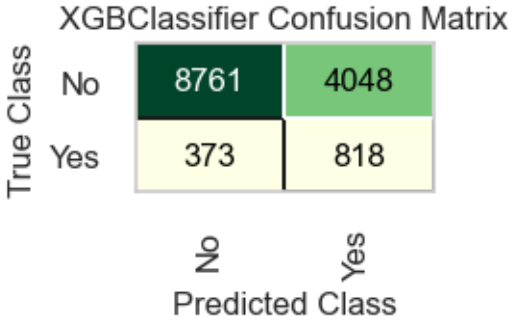


```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.70      0.81     12809
         Yes       0.17      0.67      0.27      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.68      0.54     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.68107143 0.70214286 0.68178571 0.70535714 0.68071429]
Accuracy score (average): 0.6902142857142857
F1 scores for 'Yes' values: [0.26014913 0.26584507 0.24299065 0.25608656
0.23589744]
Average F1 score: 0.538
ROC-AUC score: 0.682
PR-AUC score: 0.141
Log-loss: 0.609


Execution time: 17.518097162246704


Parameters for the dataset and transformers: OneHotEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support
```

```
          No       0.96      0.68      0.80     12809
         Yes       0.16      0.67      0.26      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.66821429 0.6925     0.69607143 0.68928571 0.68714286]
Accuracy score (average): 0.6866428571428572
F1 scores for 'Yes' values: [0.25976096 0.25583405 0.24623561 0.24740484
0.26262626]
Average F1 score: 0.531
ROC-AUC score: 0.678
PR-AUC score: 0.139
Log-loss: 0.69


Execution time: 86.07579493522644

Parameters for the dataset and transformers: WOEEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_1, binary_features_1,
categorical_features_1, other_features_1
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     12809
         Yes       0.17      0.69      0.27      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.69      0.53     14000
weighted avg       0.89      0.68      0.75     14000
```

```
Cross-validation
Accuracy scores: [0.66714286 0.68142857 0.69035714 0.68892857 0.68892857]
Accuracy score (average): 0.6833571428571428
F1 scores for 'Yes' values: [0.2544     0.2440678  0.24674196 0.24457936
0.25619129]
Average F1 score: 0.534
ROC-AUC score: 0.685
PR-AUC score: 0.142
Log-loss: 0.689


Execution time: 81.58130836486816
```

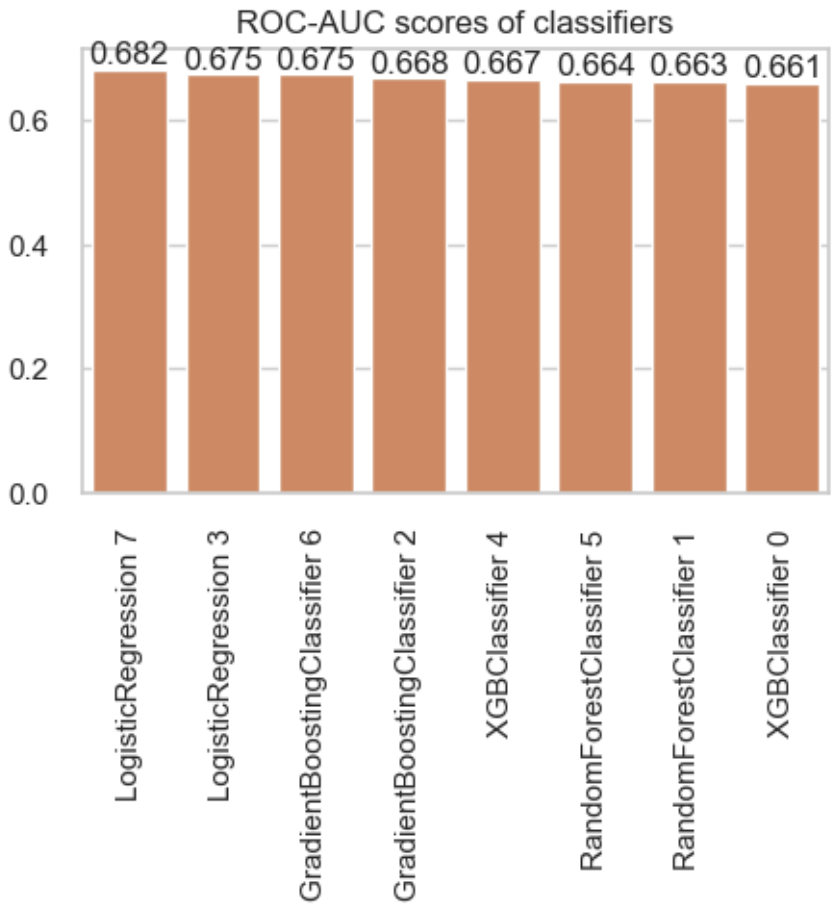
[712]:

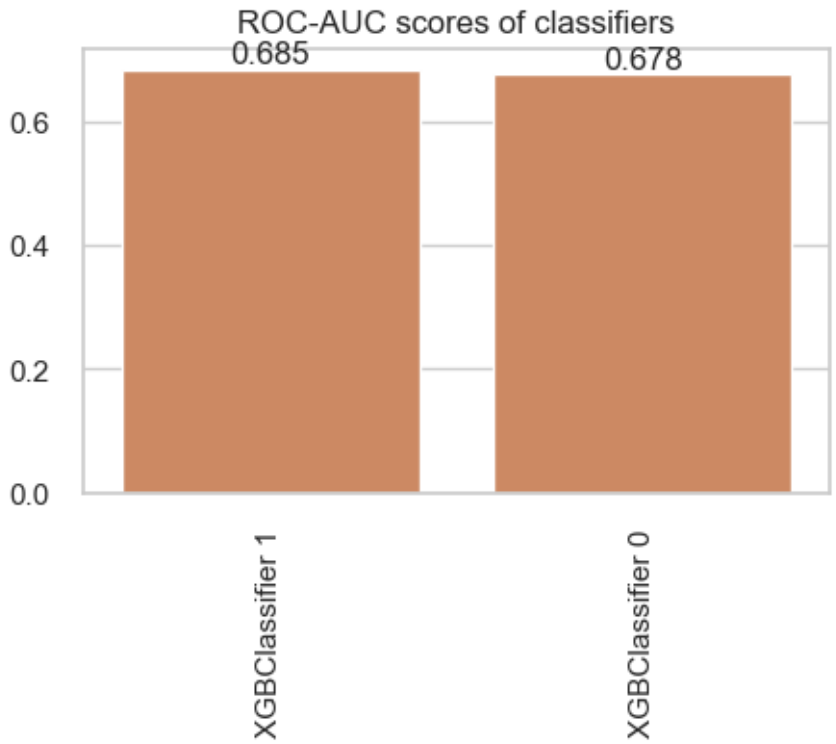
model	index	precision_score	recall_score	model_name	a_score	f1_score	ROC_AUC_score	PR_AUC_score	loss	exec_time
LogisticRegression 7	7	0.563	0.682	LogisticRegression	0.694	0.538	0.682	0.141	0.609	17.518097
LogisticRegression 3	3	0.560	0.675	LogisticRegression	0.681	0.530	0.675	0.138	0.617	21.443765
GradientBoostingClassifier 6	6	0.559	0.675	GradientBoostingClassifier	0.671	0.525	0.675	0.137	0.611	21.792629
GradientBoostingClassifier 2	2	0.557	0.668	GradientBoostingClassifier	0.665	0.519	0.668	0.133	0.612	25.304350
XGBClassifier 4	4	0.556	0.667	XGBClassifier	0.668	0.520	0.667	0.133	0.713	36.944725
RandomForestClassifier 5	5	0.556	0.664	RandomForestClassifier	0.679	0.525	0.664	0.133	0.617	18.920605
RandomForestClassifier 1	1	0.556	0.663	RandomForestClassifier	0.674	0.522	0.663	0.132	0.620	22.848152
XGBClassifier 0	0	0.554	0.661	XGBClassifier	0.664	0.517	0.661	0.130	0.713	40.786912

```
                                    encoders      cimputers      nimputers  \
model
LogisticRegression 7                   WOEEncoder  SimpleImputer  SimpleImputer
LogisticRegression 3                OneHotEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 6           WOEEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 2        OneHotEncoder  SimpleImputer  SimpleImputer
XGBClassifier 4                        WOEEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 5               WOEEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 1            OneHotEncoder  SimpleImputer  SimpleImputer
XGBClassifier 0                     OneHotEncoder  SimpleImputer  SimpleImputer

                                         scalers        num_features  \
model
LogisticRegression 7                StandardScaler  numeric_features_5
LogisticRegression 3                StandardScaler  numeric_features_5
GradientBoostingClassifier 6        StandardScaler  numeric_features_5
GradientBoostingClassifier 2        StandardScaler  numeric_features_5
XGBClassifier 4                     StandardScaler  numeric_features_5
RandomForestClassifier 5            StandardScaler  numeric_features_5
RandomForestClassifier 1            StandardScaler  numeric_features_5
XGBClassifier 0                     StandardScaler  numeric_features_5

                                              cat_features       bin_features  \
model
LogisticRegression 7                categorical_features_5  binary_features_5
LogisticRegression 3                categorical_features_5  binary_features_5
GradientBoostingClassifier 6        categorical_features_5  binary_features_5
GradientBoostingClassifier 2        categorical_features_5  binary_features_5
XGBClassifier 4                     categorical_features_5  binary_features_5
RandomForestClassifier 5            categorical_features_5  binary_features_5
RandomForestClassifier 1            categorical_features_5  binary_features_5
XGBClassifier 0                     categorical_features_5  binary_features_5

                                    other_features
model
LogisticRegression 7                other_features_5
LogisticRegression 3                other_features_5
GradientBoostingClassifier 6        other_features_5
GradientBoostingClassifier 2        other_features_5
XGBClassifier 4                     other_features_5
RandomForestClassifier 5            other_features_5
RandomForestClassifier 1            other_features_5
XGBClassifier 0                     other_features_5
```

[714]:
```
                    index  precision_score  recall_score     model_name  a_score  \
model
```

```
XGBClassifier 1      1           0.564         0.685  XGBClassifier    0.684
XGBClassifier 0      0           0.561         0.678  XGBClassifier    0.682

                f1_score  ROC_AUC_score  PR_AUC_score   loss  exec_time  \
model
XGBClassifier 1    0.534          0.685         0.142  0.689  81.581308
XGBClassifier 0    0.531          0.678         0.139  0.690  86.075795

                     encoders      cimputers      nimputers         scalers  \
model
XGBClassifier 1    WOEEncoder  SimpleImputer  SimpleImputer  StandardScaler
XGBClassifier 0  OneHotEncoder  SimpleImputer  SimpleImputer  StandardScaler

                      num_features            cat_features  \
model
XGBClassifier 1  numeric_features_1  categorical_features_1
XGBClassifier 0  numeric_features_1  categorical_features_1

                     bin_features    other_features
model
XGBClassifier 1  binary_features_1  other_features_1
XGBClassifier 0  binary_features_1  other_features_1
```

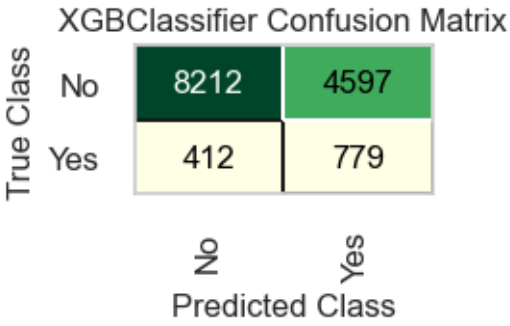

ROC-AUC scores of classifiers
0.682
0.675
0.675
0.668
0.667
0.664
0.663
0.661
0.6
0.4
0.2
0.0
LogisticRegression 7
LogisticRegression 3
GradientBoostingClassifier 6
GradientBoostingClassifier 2
XGBClassifier 4
RandomForestClassifier 5
RandomForestClassifier 1
XGBClassifier 0



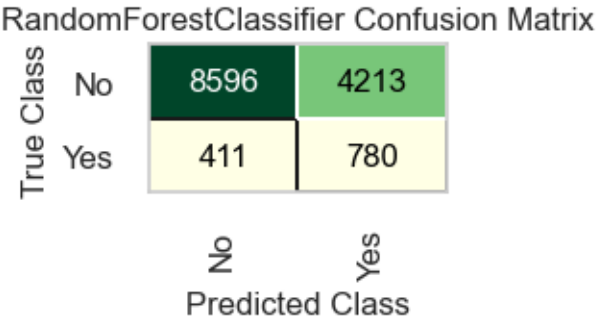
It can be observed that the XGBoost classifier with the suggested parameters (n_estimators: 1000, max_depth: 9, learning_rate: 0.024246395212299744, subsample: 0.8) generates the best roc-auc scores comparing to other trained models, but the number of features does not affect scores (the roc-auc scores for the model trained with 315 and 132 features are the same). Also, it can be seen that the choice of the WOE encoder instead of One-hot encoder generates slighly better roc-auc scores.

Feature selection based on the results of exploratory analysis Another approach was to select features based on the results of exploratory analysis. These features for which statistically significant differences in means (for numerical variables) and proportions (for binary and other categorical variables) were identified, were included in the list of selected features. Models with tuned parameters were trained on this combination of features. Results are presented bellow.

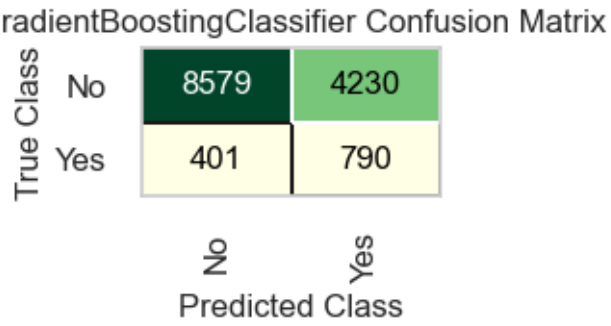
```
Parameters for the dataset and transformers: OneHotEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_6, binary_features_6,
categorical_features_6, other_features_6
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.64      0.77     12809
         Yes       0.14      0.65      0.24      1191

    accuracy                           0.64     14000
   macro avg       0.55      0.65      0.50     14000
weighted avg       0.88      0.64      0.72     14000

Cross-validation
Accuracy scores: [0.62642857 0.64535714 0.66464286 0.67214286 0.66035714]
Accuracy score (average): 0.6537857142857143
F1 scores for 'Yes' values: [0.22056632 0.21252974 0.23844282 0.22466216
0.22620016]
Average F1 score: 0.502
ROC-AUC score: 0.648
PR-AUC score: 0.124
Log-loss: 0.755


Execution time: 33.960461139678955

RandomForestClassifier
```

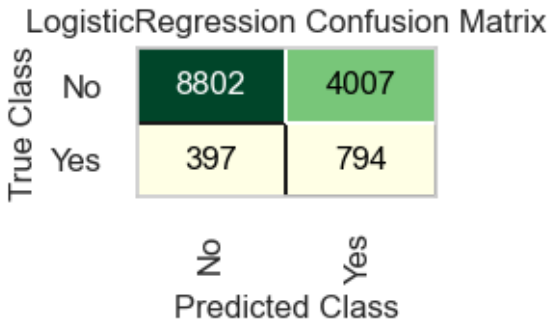


```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.67      0.79     12809
         Yes       0.16      0.65      0.25      1191

    accuracy                           0.67     14000
   macro avg       0.56      0.66      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65428571 0.66642857 0.68964286 0.69642857 0.67321429]
Accuracy score (average): 0.6759999999999999
F1 scores for 'Yes' values: [0.23659306 0.22166667 0.23838738 0.24778761
0.2394015 ]
Average F1 score: 0.52
ROC-AUC score: 0.663
PR-AUC score: 0.132
Log-loss: 0.619


Execution time: 22.791489124298096

GradientBoostingClassifier
```



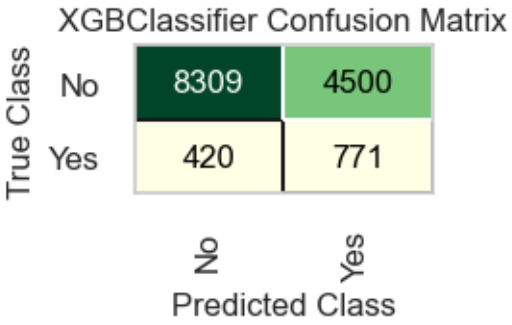

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.67      0.79     12809
         Yes       0.16      0.66      0.25      1191

    accuracy                           0.67     14000
```

```
   macro avg       0.56      0.67      0.52     14000
weighted avg       0.89      0.67      0.74     14000

Cross-validation
Accuracy scores: [0.65785714 0.66964286 0.68       0.68535714 0.67892857]
Accuracy score (average): 0.6743571428571429
F1 scores for 'Yes' values: [0.24208861 0.22334173 0.23418803 0.23590633
0.2322801 ]
Average F1 score: 0.521
ROC-AUC score: 0.667
PR-AUC score: 0.133
Log-loss: 0.611


Execution time: 23.507110118865967

LogisticRegression
```



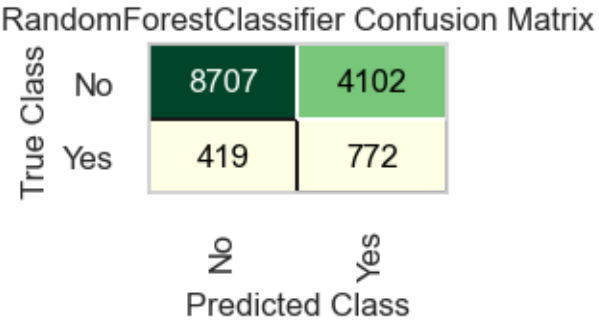

```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.69      0.80     12809
         Yes       0.17      0.67      0.27      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.68      0.53     14000
weighted avg       0.89      0.69      0.75     14000

Cross-validation
Accuracy scores: [0.67       0.68357143 0.67964286 0.69035714 0.68392857]
Accuracy score (average): 0.6815
F1 scores for 'Yes' values: [0.24632953 0.23488774 0.23398804 0.24279476
0.23509075]
Average F1 score: 0.532
ROC-AUC score: 0.677
```

```
PR-AUC score: 0.139
Log-loss: 0.605


Execution time: 21.406705141067505

Parameters for the dataset and transformers: WOEEncoder,
SimpleImputer(strategy='median'), SimpleImputer(fill_value=0,
strategy='constant'), StandardScaler, numeric_features_6, binary_features_6,
categorical_features_6, other_features_6
XGBClassifier
```

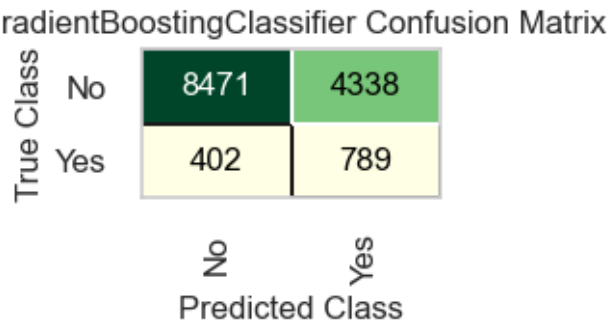



```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.65      0.77     12809
         Yes       0.15      0.65      0.24      1191

    accuracy                           0.65     14000
   macro avg       0.55      0.65      0.51     14000
weighted avg       0.88      0.65      0.73     14000

Cross-validation
Accuracy scores: [0.66142857 0.65821429 0.675      0.65928571 0.66678571]
Accuracy score (average): 0.6641428571428571
F1 scores for 'Yes' values: [0.2428115  0.2313253  0.23657718 0.21416804
0.22314738]
Average F1 score: 0.505
ROC-AUC score: 0.648
PR-AUC score: 0.125
Log-loss: 0.73


Execution time: 30.536390781402588
```

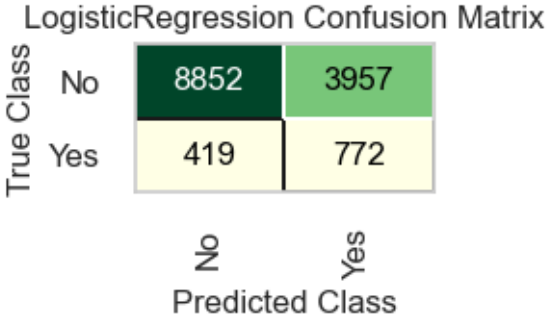
RandomForestClassifier




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.68      0.79     12809
         Yes       0.16      0.65      0.25      1191

    accuracy                           0.68     14000
   macro avg       0.56      0.66      0.52     14000
weighted avg       0.89      0.68      0.75     14000

Cross-validation
Accuracy scores: [0.65428571 0.67571429 0.69321429 0.7        0.6775    ]
Accuracy score (average): 0.6801428571428572
F1 scores for 'Yes' values: [0.24727838 0.23697479 0.22959641 0.23913043
0.2418136 ]
Average F1 score: 0.524
ROC-AUC score: 0.664
PR-AUC score: 0.133
Log-loss: 0.622


Execution time: 18.686326026916504

GradientBoostingClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.66      0.78     12809
         Yes       0.15      0.66      0.25      1191

    accuracy                           0.66     14000
   macro avg       0.55      0.66      0.52     14000
weighted avg       0.89      0.66      0.74     14000

Cross-validation
Accuracy scores: [0.66607143 0.68321429 0.68571429 0.69428571 0.67642857]
Accuracy score (average): 0.6811428571428572
F1 scores for 'Yes' values: [0.25020048 0.24123182 0.22942207 0.24514991
0.23608769]
Average F1 score: 0.516
ROC-AUC score: 0.662
PR-AUC score: 0.131
Log-loss: 0.617


Execution time: 19.799463272094727

LogisticRegression
```



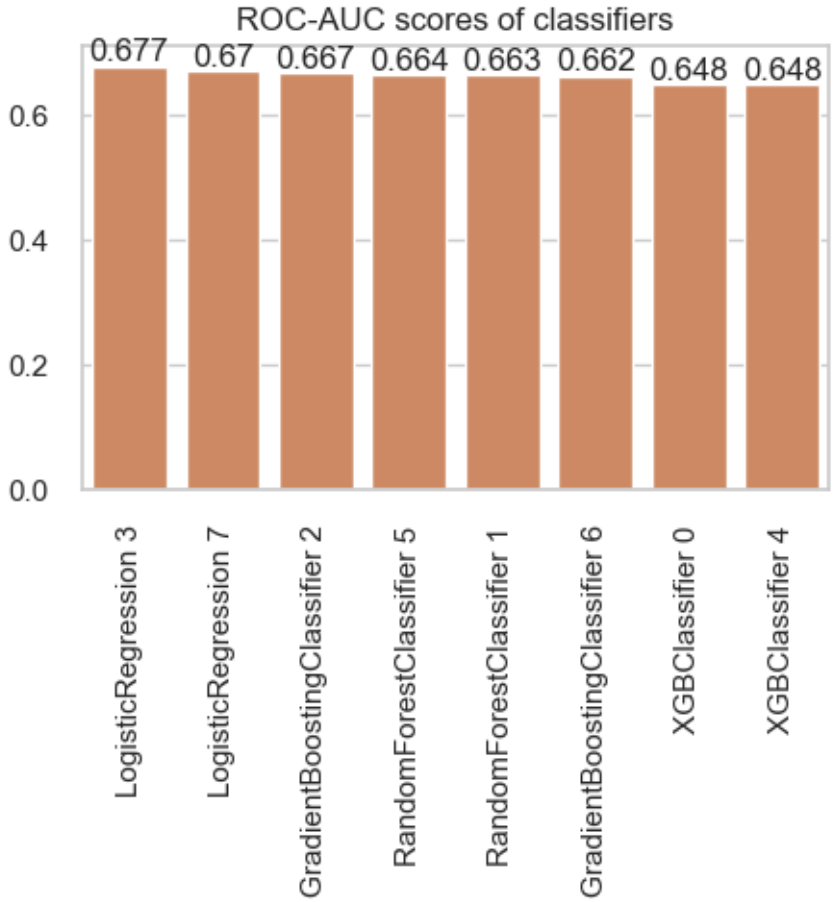
```
Classification Report:
              precision    recall  f1-score   support

          No       0.95      0.69      0.80     12809
         Yes       0.16      0.65      0.26      1191

    accuracy                           0.69     14000
   macro avg       0.56      0.67      0.53     14000
weighted avg       0.89      0.69      0.76     14000

Cross-validation
Accuracy scores: [0.67035714 0.6975     0.6825     0.7075     0.69821429]
Accuracy score (average): 0.6912142857142858
F1 scores for 'Yes' values: [0.25504439 0.24171889 0.23690987 0.26149684
0.2435094 ]
Average F1 score: 0.531
ROC-AUC score: 0.67
PR-AUC score: 0.136
Log-loss: 0.606


Execution time: 17.27903389930725
```

[707]:

model	index	precision_score	recall_score	model_name	a_score	f1_score
LogisticRegression 3	3	0.561	0.677	LogisticRegression	0.685	0.532
LogisticRegression 7	7	0.559	0.670	LogisticRegression	0.687	0.531
GradientBoostingClassifier 2	2	0.556	0.667	GradientBoostingClassifier	0.669	0.521
RandomForestClassifier 5	5	0.556	0.664	RandomForestClassifier	0.677	0.524
RandomForestClassifier 1	1	0.555	0.663	RandomForestClassifier	0.670	0.520
GradientBoostingClassifier 6	6	0.554	0.662	GradientBoostingClassifier	0.661	0.516
XGBClassifier 0	0	0.549	0.648	XGBClassifier	0.642	0.502
XGBClassifier 4	4	0.549	0.648	XGBClassifier	0.649	0.505

```
                            ROC_AUC_score  PR_AUC_score   loss  exec_time  \
model
LogisticRegression 3                0.677         0.139  0.605  21.406705
LogisticRegression 7                0.670         0.136  0.606  17.279034
GradientBoostingClassifier 2        0.667         0.133  0.611  23.507110
RandomForestClassifier 5            0.664         0.133  0.622  18.686326
RandomForestClassifier 1            0.663         0.132  0.619  22.791489
GradientBoostingClassifier 6        0.662         0.131  0.617  19.799463
XGBClassifier 0                     0.648         0.124  0.755  33.960461
XGBClassifier 4                     0.648         0.125  0.730  30.536391

                                 encoders      cimputers      nimputers  \
model
LogisticRegression 3          OneHotEncoder  SimpleImputer  SimpleImputer
LogisticRegression 7             WOEEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 2  OneHotEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 5         WOEEncoder  SimpleImputer  SimpleImputer
RandomForestClassifier 1      OneHotEncoder  SimpleImputer  SimpleImputer
GradientBoostingClassifier 6     WOEEncoder  SimpleImputer  SimpleImputer
XGBClassifier 0               OneHotEncoder  SimpleImputer  SimpleImputer
XGBClassifier 4                  WOEEncoder  SimpleImputer  SimpleImputer

                                    scalers        num_features  \
model
LogisticRegression 3          StandardScaler  numeric_features_6
LogisticRegression 7          StandardScaler  numeric_features_6
GradientBoostingClassifier 2  StandardScaler  numeric_features_6
RandomForestClassifier 5      StandardScaler  numeric_features_6
RandomForestClassifier 1      StandardScaler  numeric_features_6
GradientBoostingClassifier 6  StandardScaler  numeric_features_6
XGBClassifier 0               StandardScaler  numeric_features_6
XGBClassifier 4               StandardScaler  numeric_features_6

                                          cat_features       bin_features  \
model
LogisticRegression 3          categorical_features_6  binary_features_6
LogisticRegression 7          categorical_features_6  binary_features_6
GradientBoostingClassifier 2  categorical_features_6  binary_features_6
RandomForestClassifier 5      categorical_features_6  binary_features_6
RandomForestClassifier 1      categorical_features_6  binary_features_6
GradientBoostingClassifier 6  categorical_features_6  binary_features_6
XGBClassifier 0               categorical_features_6  binary_features_6
XGBClassifier 4               categorical_features_6  binary_features_6

                                other_features
model
```

```
LogisticRegression 3          other_features_6
LogisticRegression 7          other_features_6
GradientBoostingClassifier 2  other_features_6
RandomForestClassifier 5      other_features_6
RandomForestClassifier 1      other_features_6
GradientBoostingClassifier 6  other_features_6
XGBClassifier 0               other_features_6
XGBClassifier 4               other_features_6
```



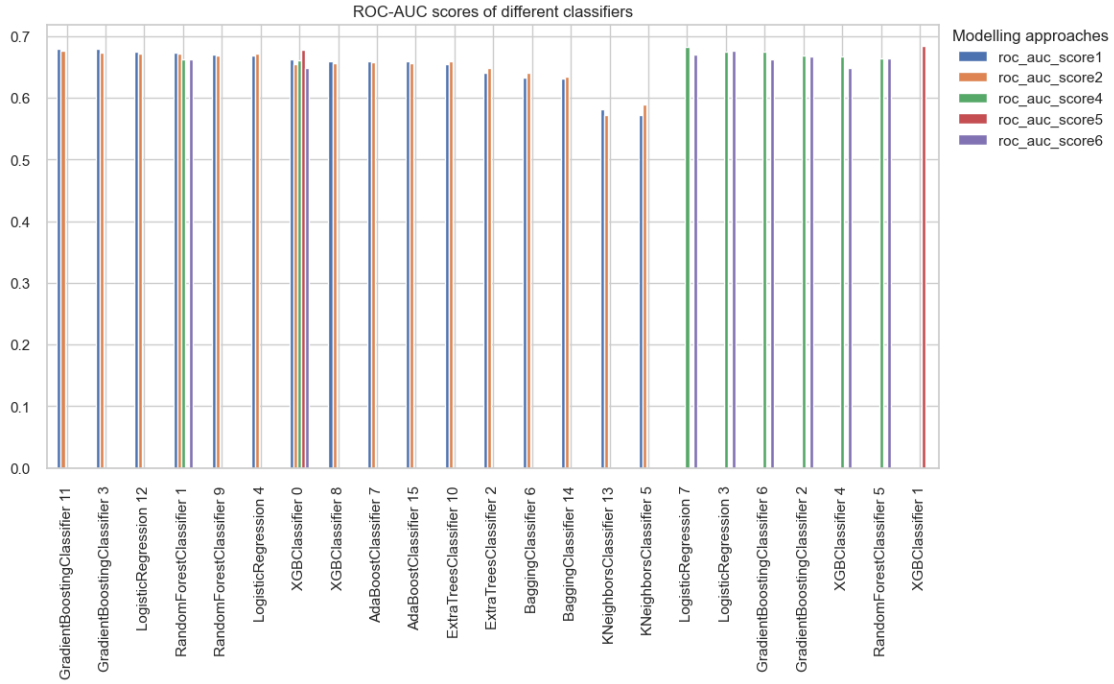
It can be observed that this approach did not generate better roc-auc scores than other feature selection approaches. the logistic regression classifier preprocessed with one-hot encoder gave the best scores compared to other models.

Comparing different machine learning models Scores of the models trained with different parameters, feature combinations and transformers were compared between each other (scores presented in the data table and the plot).

[730]:

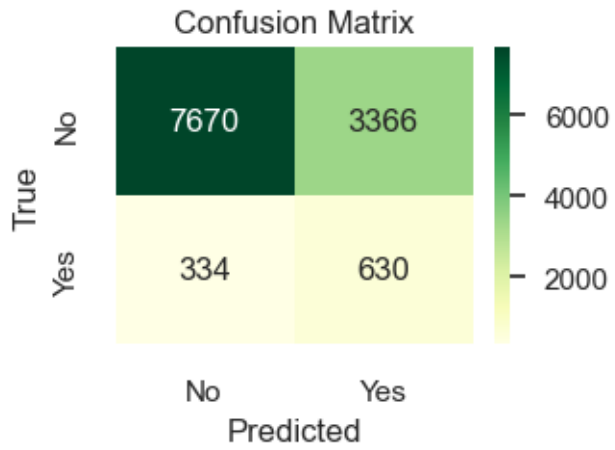
model	roc_auc_score1	roc_auc_score2	roc_auc_score4	roc_auc_score5	roc_auc_score6
GradientBoostingClassifier 11	0.680	0.676	NaN	NaN	NaN
GradientBoostingClassifier 3	0.679	0.673	NaN	NaN	NaN
LogisticRegression 12	0.675	0.672	NaN	NaN	NaN
RandomForestClassifier 1	0.674	0.672	0.663	NaN	0.663
RandomForestClassifier 9	0.670	0.668	NaN	NaN	NaN
LogisticRegression 4	0.669	0.672	NaN	NaN	NaN
XGBClassifier 0	0.662	0.655	0.661	0.678	0.648
XGBClassifier 8	0.660	0.656	NaN	NaN	NaN
AdaBoostClassifier 7	0.659	0.658	NaN	NaN	NaN
AdaBoostClassifier 15	0.659	0.657	NaN	NaN	NaN
ExtraTreesClassifier 10	0.654	0.659	NaN	NaN	NaN
ExtraTreesClassifier 2	0.640	0.649	NaN	NaN	NaN
BaggingClassifier 6	0.633	0.641	NaN	NaN	NaN
BaggingClassifier 14	0.632	0.634	NaN	NaN	NaN
KNeighborsClassifier 13	0.582	0.573	NaN	NaN	NaN
KNeighborsClassifier 5	0.573	0.590	NaN	NaN	NaN
LogisticRegression 7	NaN	NaN	0.682	NaN	0.670
LogisticRegression 3	NaN	NaN	0.675	NaN	0.677
GradientBoostingClassifier 6	NaN	NaN	0.675	NaN	0.662
GradientBoostingClassifier 2	NaN	NaN	0.668		
XGBClassifier 4	NaN	NaN	0.667		
RandomForestClassifier 5	NaN	NaN	0.664		
XGBClassifier 1	NaN	NaN	NaN		

GradientBoostingClassifier 2	NaN	0.667
XGBClassifier 4	NaN	0.648
RandomForestClassifier 5	NaN	0.664
XGBClassifier 1	0.685	NaN

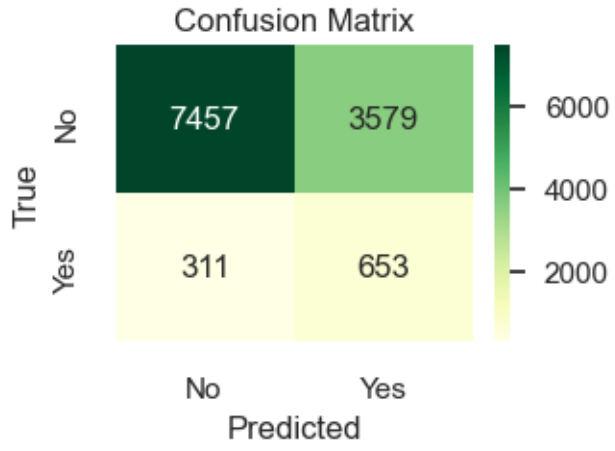


Predicting clients with payment difficulties on the best performing models and the data from the test dataset The best performing models (the Logistic regression classifier with data preprocessed with the WOEencoder and etc. and the combination of 132 features; the Gradient Boosting classifier trained with the WOE encoder and etc. and all features; and the XGBoost classifier with tuned parameters, the WOE encoder and all features) were selected for predicting clients with payment difficulties on the test dataset (separated from the random sample for machine learning).

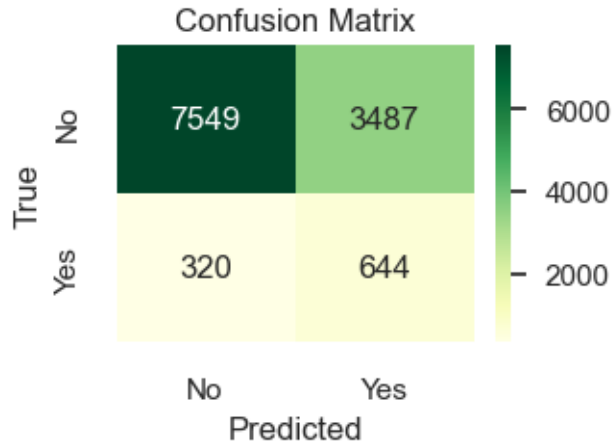
`LogisticRegression`




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.69      0.81     11036
         Yes       0.16      0.65      0.25       964

    accuracy                           0.69     12000
   macro avg       0.56      0.67      0.53     12000
weighted avg       0.89      0.69      0.76     12000

Accuracy score: 0.692
F1 score: 0.53
ROC-AUC score: 0.674
PR-AUC score: 0.131
Log-loss: 0.612
GradientBoostingClassifier
```



```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.79     11036
         Yes       0.15      0.68      0.25       964

    accuracy                           0.68     12000
   macro avg       0.56      0.68      0.52     12000
weighted avg       0.90      0.68      0.75     12000

Accuracy score: 0.676
F1 score: 0.522
ROC-AUC score: 0.677
PR-AUC score: 0.13
Log-loss: 0.607
XGBClassifier
```




```
Classification Report:
              precision    recall  f1-score   support

          No       0.96      0.68      0.80     11036
         Yes       0.16      0.67      0.25       964

    accuracy                           0.68     12000
   macro avg       0.56      0.68      0.53     12000
weighted avg       0.89      0.68      0.75     12000

Accuracy score: 0.683
F1 score: 0.526
ROC-AUC score: 0.676
```

```
PR-AUC score: 0.131
Log-loss: 0.702
```

The roc-auc scores (and other scores) are quite similar as the scores generated on the validation dataset. The XGBoost classifier generated the best scores, thus, it was chosen for the use in the API and deployment.

[727]: `['XGBoost.joblib']`

BUILDING AND RUNNING DEEP LEARNING MODELS Also, it was decided to train a deep learning model to predict clients with loan payment difficulties. For that purpose, the tensorflow library adn keras modules were used. The model was trainen on the full dataset of 3057511 cases and 315 features.

Spliting the data into training and test datasets The target variable was separated, data were split into training (70 percent) and test (30 percent) datasets.

Preprocessing the data Data were preprocessed with transformers in in the lists which were used for sklearn machine learning (WOE encoder was chosen). Also, random undersampling was applied for the data as it was highly unbalanced.

Running the model

```
Epoch 1/20
  1/865 […] - ETA: 5:50 - loss: 0.0951 - roc_auc:
0.4479

2023-08-30 10:33:11.970129: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

865/865 [==============================] - ETA: 0s - loss: 0.0490 - roc_auc:
0.6877

2023-08-30 10:33:17.595886: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

865/865 [==============================] - 7s 8ms/step - loss: 0.0490 - roc_auc:
0.6877 - val_loss: 0.0634 - val_roc_auc: 0.0000e+00
Epoch 2/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0460 - roc_auc:
0.7395 - val_loss: 0.0685 - val_roc_auc: 0.0000e+00
Epoch 3/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0451 - roc_auc:
0.7532 - val_loss: 0.0613 - val_roc_auc: 0.0000e+00
Epoch 4/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0447 - roc_auc:
0.7589 - val_loss: 0.0649 - val_roc_auc: 0.0000e+00
Epoch 5/20
```

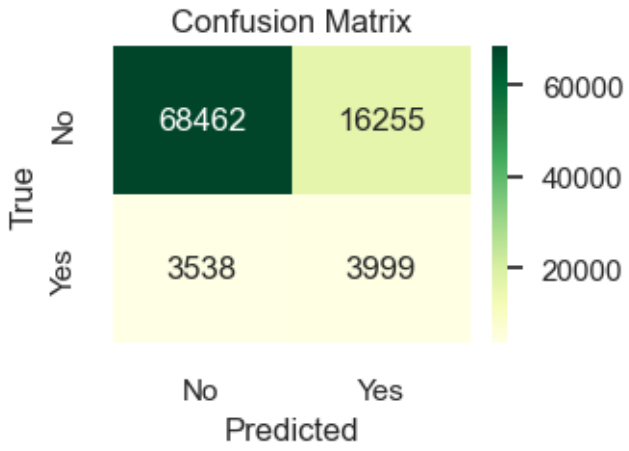
```
865/865 [==============================] - 6s 7ms/step - loss: 0.0445 - roc_auc:
0.7618 - val_loss: 0.0578 - val_roc_auc: 0.0000e+00
Epoch 6/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0442 - roc_auc:
0.7651 - val_loss: 0.0685 - val_roc_auc: 0.0000e+00
Epoch 7/20
865/865 [==============================] - 7s 8ms/step - loss: 0.0441 - roc_auc:
0.7670 - val_loss: 0.0658 - val_roc_auc: 0.0000e+00
Epoch 8/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0439 - roc_auc:
0.7702 - val_loss: 0.0750 - val_roc_auc: 0.0000e+00
Epoch 9/20
865/865 [==============================] - 7s 8ms/step - loss: 0.0437 - roc_auc:
0.7733 - val_loss: 0.0624 - val_roc_auc: 0.0000e+00
Epoch 10/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0436 - roc_auc:
0.7746 - val_loss: 0.0619 - val_roc_auc: 0.0000e+00
Epoch 11/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0434 - roc_auc:
0.7770 - val_loss: 0.0616 - val_roc_auc: 0.0000e+00
Epoch 12/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0432 - roc_auc:
0.7784 - val_loss: 0.0670 - val_roc_auc: 0.0000e+00
Epoch 13/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0431 - roc_auc:
0.7804 - val_loss: 0.0628 - val_roc_auc: 0.0000e+00
Epoch 14/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0430 - roc_auc:
0.7819 - val_loss: 0.0686 - val_roc_auc: 0.0000e+00
Epoch 15/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0428 - roc_auc:
0.7836 - val_loss: 0.0582 - val_roc_auc: 0.0000e+00
Epoch 16/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0427 - roc_auc:
0.7847 - val_loss: 0.0576 - val_roc_auc: 0.0000e+00
Epoch 17/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0426 - roc_auc:
0.7870 - val_loss: 0.0605 - val_roc_auc: 0.0000e+00
Epoch 18/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0424 - roc_auc:
0.7889 - val_loss: 0.0734 - val_roc_auc: 0.0000e+00
Epoch 19/20
865/865 [==============================] - 6s 7ms/step - loss: 0.0424 - roc_auc:
0.7887 - val_loss: 0.0661 - val_roc_auc: 0.0000e+00
Epoch 20/20
865/865 [==============================] - 7s 8ms/step - loss: 0.0422 - roc_auc:
0.7912 - val_loss: 0.0638 - val_roc_auc: 0.0000e+00
```

```
Execution time: 127.31609606742859
```

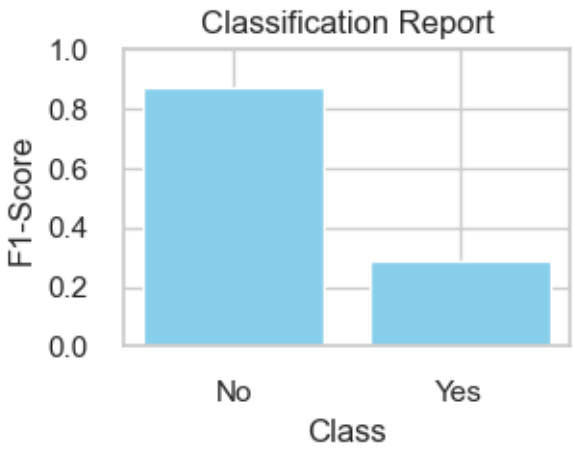
Predicting on the test data and evaluating the model

```
 128/2883 [>…] - ETA: 3s

2023-08-30 10:35:25.349925: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

2883/2883 [==============================] - 4s 1ms/step

2883/2883 [==============================] - 14s 5ms/step - loss: 0.0390 -
roc_auc: 0.7511
Test loss: 0.0390, Test ROC-AUC: 0.7511
```

The confusion matrix and classification report for this model predictions are presented bellow.



	precision	recall	f1-score	support
No	0.95	0.81	0.87	84717
Yes	0.20	0.53	0.29	7537
accuracy			0.79	92254
macro avg	0.57	0.67	0.58	92254
weighted avg	0.89	0.79	0.83	92254



It can be seen that the deep learning model (the roc-auc score - 0.751; f1 score for 'Yes' - 0.29; f1 score for "No" - 0.87) generates better scores than any one of machine learning models trained with the algorithms in the sklearn library.

The model is saved to be used for the API and deployment.

Using the selected model for prediction The XG Boost model as well as the deep learning model will be used for predicting probabilities of clients' having difficulties to pay loans. The dataset which will be used for prediction is the fulldata_test which was preprocessed from the 'application_test.csv' file (see the exploratory analysis part). The code bellow is used to randomy select client ids (the number of ids could be chosen arbitrarily) from this dataset an generate probabilites for a client to have loan payment difficulties by the pretained models.

In order to provide predictions from the deep learning model, the data has to be preprocessed. The preporcessor is taken from the XG Boost model pipeline. However, the function 'fit_transform' requires the target variable as an argument which is not present in the fulldata_test dataset (but not uses it for generating predictions). In order to solve this issue, a random pandas Series with binary values 0 and 1 were generated and included as an argument for the fit_transform function.

Prediction of random cases from the dataset:

```
1/1 [==============================] - 0s 36ms/step

 Probability that the client whose id is 437532 will experience paying
difficulties
(prediction by a XGBoost classifier):  0.557

 Probability that the client whose id is 437532 will experience paying
difficulties
(prediction by a deep learning model):  0.213
```

```
 Probability that the client whose id is 196748 will experience paying
difficulties
(prediction by a XGBoost classifier):  0.084

 Probability that the client whose id is 196748 will experience paying
difficulties
(prediction by a deep learning model):  0.461



 Probability that the client whose id is 280941 will experience paying
difficulties
(prediction by a XGBoost classifier):  0.845

 Probability that the client whose id is 280941 will experience paying
difficulties
(prediction by a deep learning model):  0.522



 Probability that the client whose id is 220516 will experience paying
difficulties
(prediction by a XGBoost classifier):  0.907

 Probability that the client whose id is 220516 will experience paying
difficulties
(prediction by a deep learning model):  0.401



 Probability that the client whose id is 237624 will experience paying
difficulties
(prediction by a XGBoost classifier):  0.684

 Probability that the client whose id is 237624 will experience paying
difficulties
(prediction by a deep learning model):  0.336
```

Prediction of cases by a client id input: Also, the ids could be provided by the user input and probabilities generated on them.

```
What is the id of a client whose riskiness to experience paying difficulties you
would like to predict?  353167

1/1 [==============================] - 0s 57ms/step
```

```
 Probability that the client whose id is 353167 will experience paying
difficulties
(prediction by a HGBoost classifier): 0.857

 Probability that the client whose id is 353167 will experience paying
difficulties
(prediction by a deep learning model): 0.648
```

1.1 Conclusions

These final conclusion can be made;

1. From the exploratory analysis it can be observed that it is more likely that the clients will experience loan payment difficulties if they (some hypotheses were confirmed others not):

- are of older age;
- have been living longer in the same area;
- have not changed their id document for a longer time;
- live in a region with a rating of higher number (rather the region 3 than the region 1);
- live in a region with a rating of higher number (rather the region 3 than the region 1) taking city into account;
- the living conditions of the factor 2 of the clients have higher scores (e.g. have older houses);
- take cash loans;
- own real estate;
- are on maternity leave or unemployed;
- provided home and work phone numbers;
- their permanent adress does not match contact or work adresses in region or city levels;
- work in agriculture, business entity (type 3), industry (type 1,11, 13, 3, 4, 8), construction, cleaning, mobile, postal, realtor, restaurant, security, trade (type 1, 3, 7), transport (type 3, 4);
- are self_employed;
- are in civil mariage or single/ not married;
- were unaccompanied or accompanied by a group of people when applying for a loan;
- live in rented apartment or with parents;
- work as low-skill laborers, laborers, drivers, security staff, waiters/ barmen staff, cooking staff (percentages higher than 10 percent in the “No” group);
- live in specific housing, walls are wooden (percentages higher than 9 percent in the “Yes” group);
- are men.

2. It is more likely that the clients will not experience loan payment difficulties if they:

- have higher income;
- they took credits of higher amount;
- live in more populated regions;
- have better education;
- the living conditions of the factor 2 of the clients have higher scores (e.g have houses of with longer periods ofexploitation);
- numbers of the Credit Bureau enquiries about the person of teh factor 2 are higher (i.e., more enquiries during the last quartier).

- take revolving loans;
- own real estate;
- provided their mobile phone number;
- work for a bank, the government, industry (type 12, 9), kindergarden, medicine, military, police, school, security ministries, trade (type 6), university;
- are married or widows;
- work as core staff, accountants, medicine staff, managers, private service staff, high skill tech staff, hr staff (percentages less than 7 percent in the "Yes" group;
- live in monolithic housing (percentages lower than 5 in the "Yes" group);
- are women.

3. the XGBoost classifier:

- with these parameters (n_estimators: 1000, max_depth: 9, learning_rate: 0.024246395212299744, subsample: 0.8) suggested by the Bayesian optimization,
- trained on all 315 features selected after initial exploratory analysis (which included reduction dimension, cleaning data, transformation of data joined from tables with time dimension, recoding of some features, etc.)
- train with such transformers as WOE encoder, standardscaler is able to generate the highest roc-auc WOEEncoder, SimpleImputer(strategy=median) for numerical features, SimpleImputer(fill_value=0, strategy=constant) for categorical features, standard scaler for numeric features

is able to generathe the highest roc-auc scores (0.68 for validation data and 0.676 for test data).

4. A deep learning model trained with the tensorflow model generates event higher roc-auc scores comparing to XGboos classifier - 0.75.

5. The API is created which randomly selects client ids from the test dataset and, based on two models' - XG Boost classifier and the dee learning model - predictions, provides probabilities that the clients will experience difficulties while paying loans. Also, predictions could be generated by by the user input of client ids. Such API could be valuable to the Home Credit Club company which, taking into account these prediction, could make decisions either to accept applications for credits or not.

Limitations and suggestions for improvement:

1. Scores of classifiers which were tested are mediocre (though they are better than random guessing). Maybe scores could be improved by more advanced feature engeneering (creation of new features) and hyperparameter tuning, arso by trying different datatransformers.

1.2 References

1. Dhruv Narayanan. Home Credit Default Risk (Part 1) : Business Understanding, Data Cleaning and EDA, Medium, https://medium.com/analytics-vidhya/home-credit-default-risk-part-1-business-understanding-data-cleaning-and-eda-1203913e979c
2. Dhruv Narayanan.Home Credit Default Risk (Part 2): Feature Engineering and Modelling-I, Medium, https://medium.com/@dhruvnarayanan20/home-credit-default-risk-part-2-feature-engineering-and-modelling-i-be9385ad77fd
3. Home Credit Default Risk, https://www.kaggle.com/competitions/home-credit-default-risk/data