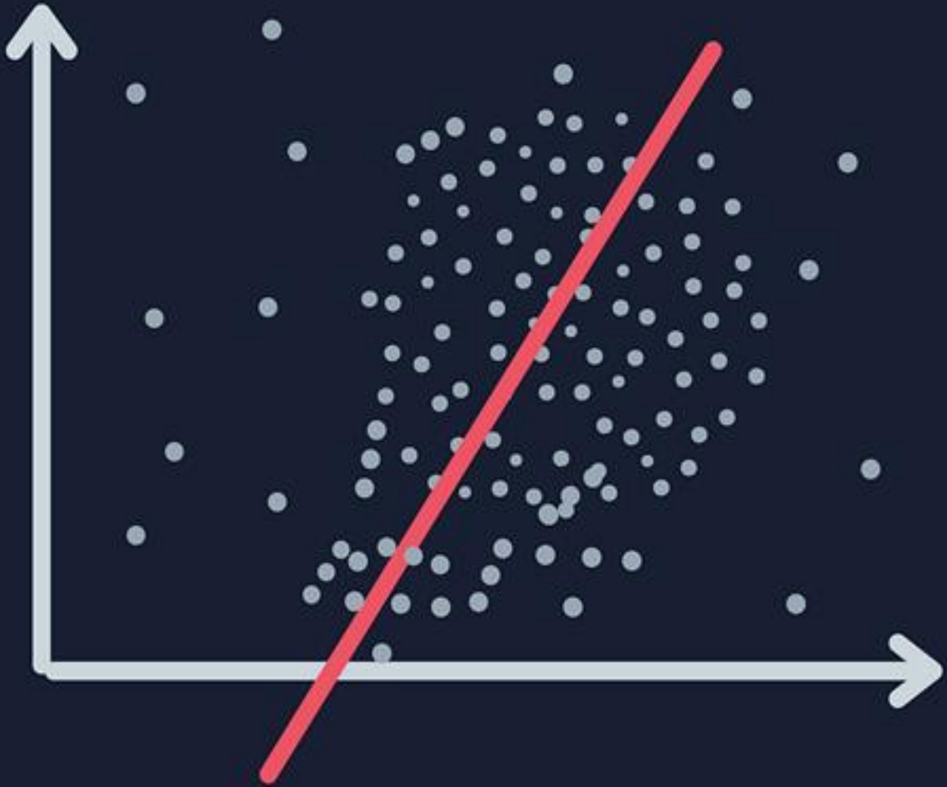
Machine Learning project presentation

Regression - Aleksander Wieliński 420 272

Classification - Jakub Gazda 419 272

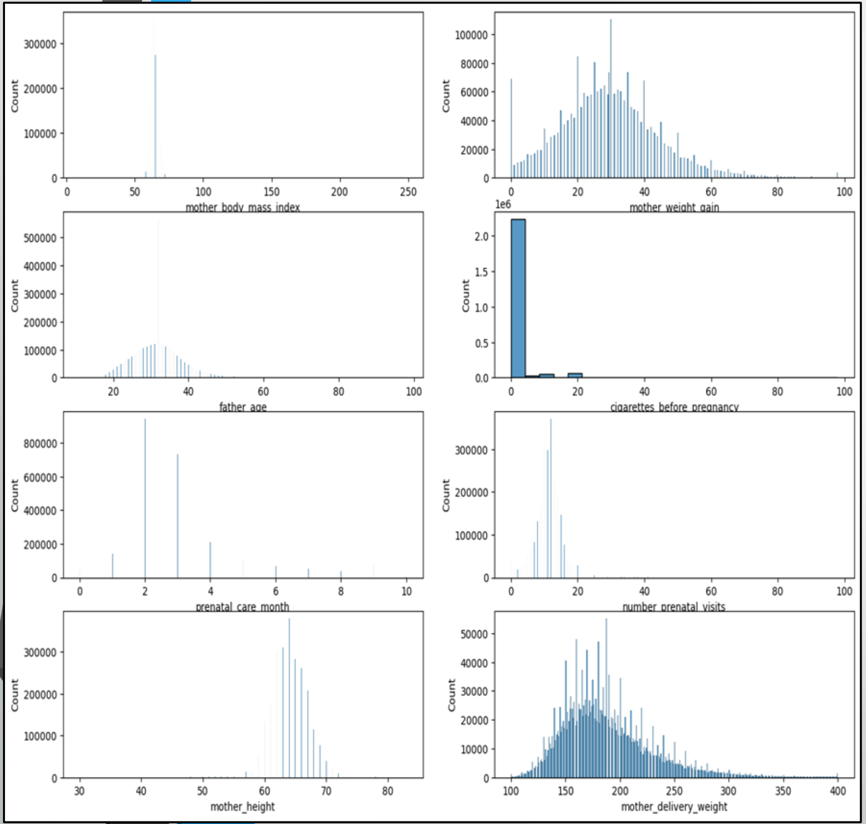
Regression

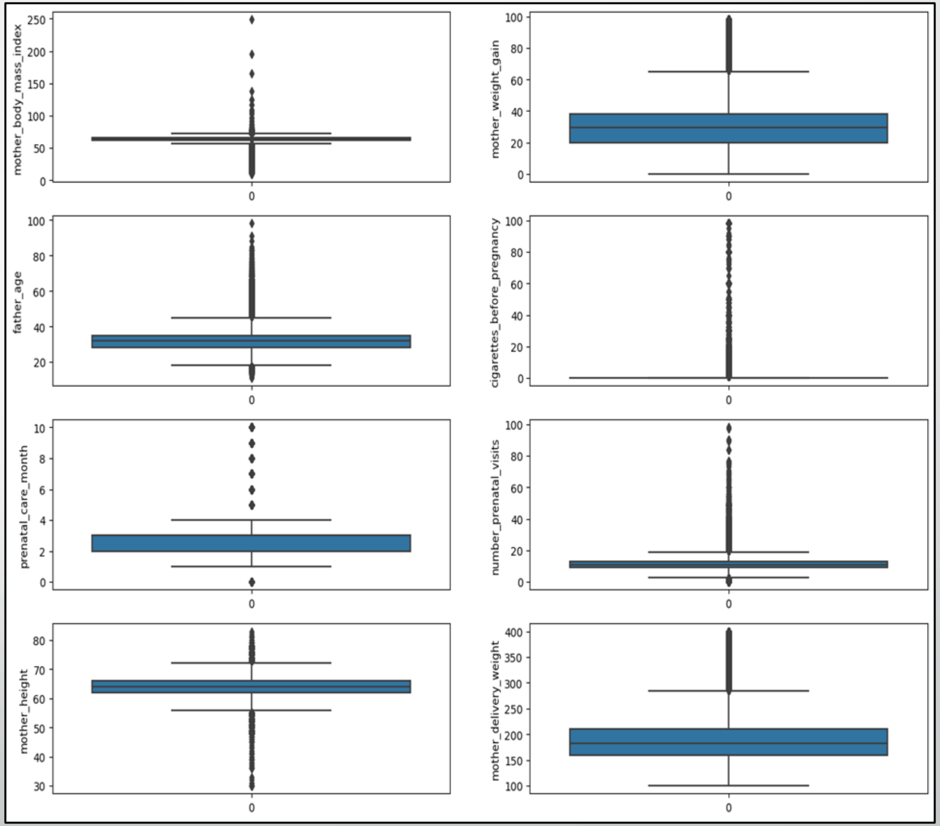


Explaining the data – before cleaning

Variable	Mean	Standard Dev.	Minimum	20%	50%	Maximum
mother_body_mass_index	27.17	6.76	13.00	21.60	25.70	69.80
mother_marital_status	1.40	0.49	1.00	1.00	1.00	2.00
mother_delivery_weight	188.32	41.37	100.00	154.00	181.00	400.00
mother_race	1.52	1.11	1.00	1.00	1.00	6.00
mother_height	64.12	2.84	30.00	62.00	64.00	78.00
mother_weight_gain	29.48	15.15	0.00	17.00	29.00	98.00
father_age	31.80	6.81	11.00	26.00	31.00	98.00
father_education	4.90	2.31	1.00	3.00	4.00	9.00
cigarettes_before_pregnancy	1.10	4.73	0.00	0.00	0.00	98.00
prenatal_care_month	5.30	15.06	0.00	2.00	3.00	99.00
number_prenatal_visits	11.29	4.20	0.00	9.00	12.00	98.00
newborn_weight	3261.84	590.47	227.00	2865.00	3300.00	8165.00
Variable	**Unique**	**Top**	**Frequency**			
previous_cesarean	3	N	2020874			
newborn_gender	2	M	1225891			

Variables distribution – after cleaning

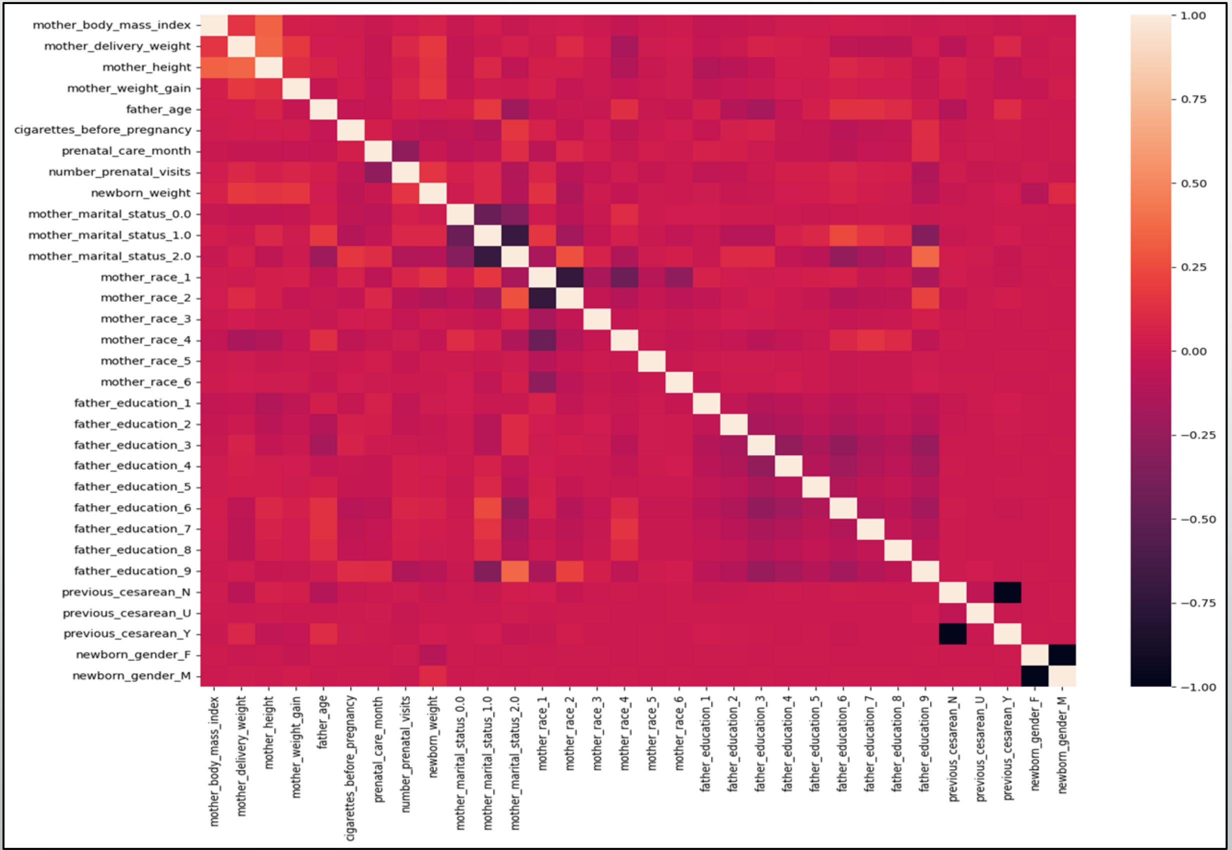




Data cleaning & preparation

1. **Missing values:**
    a. **mother_height, mother_body_mass_index** - calculated using a BMI formula
    b. **mother_delivery_weight, number_prenatal_visits, father_age, mother_weight_gain** - filled with mean values
    c. **mother_marital_status** - new, 0 variable for '*other marital status*' (since the missing % was big)
    d. **cigarettes_before_pregnancy** - missings replaced with 0 (optimistic approach)
    e. **prenatal_care_month** - 99 replaced with 9, with an assumption that a minimum of care was provided at/before birth
    f. No missing values, or outliers were removed, as the model performance was <u>not improved</u>. We believe there is a valid reason for outliers and they seem to be providing better fit.
2. **One-hot encoding:**
    a. mother_marital_status, mother_race, father_education, previous_cesarean, newborn_gender
3. **Feature engineering:**
    a. mother_height * mother_delivery_weight,
    b. mother_weight_gain * mother_body_mass_index,
    c. number_prenatal_visits * prenatal_care_month

Feature engineering - reasons for features added



Model comparison

	Model	MAE
0	XGBoost	403.57
1	Neural network	404.45
2	Ridge	410.33
3	Linear Regression	410.33
4	Elastic Net	410.86
5	Random forest	412.85
6	K-nearest neighbours	445.14
7	Decision tree	592.93

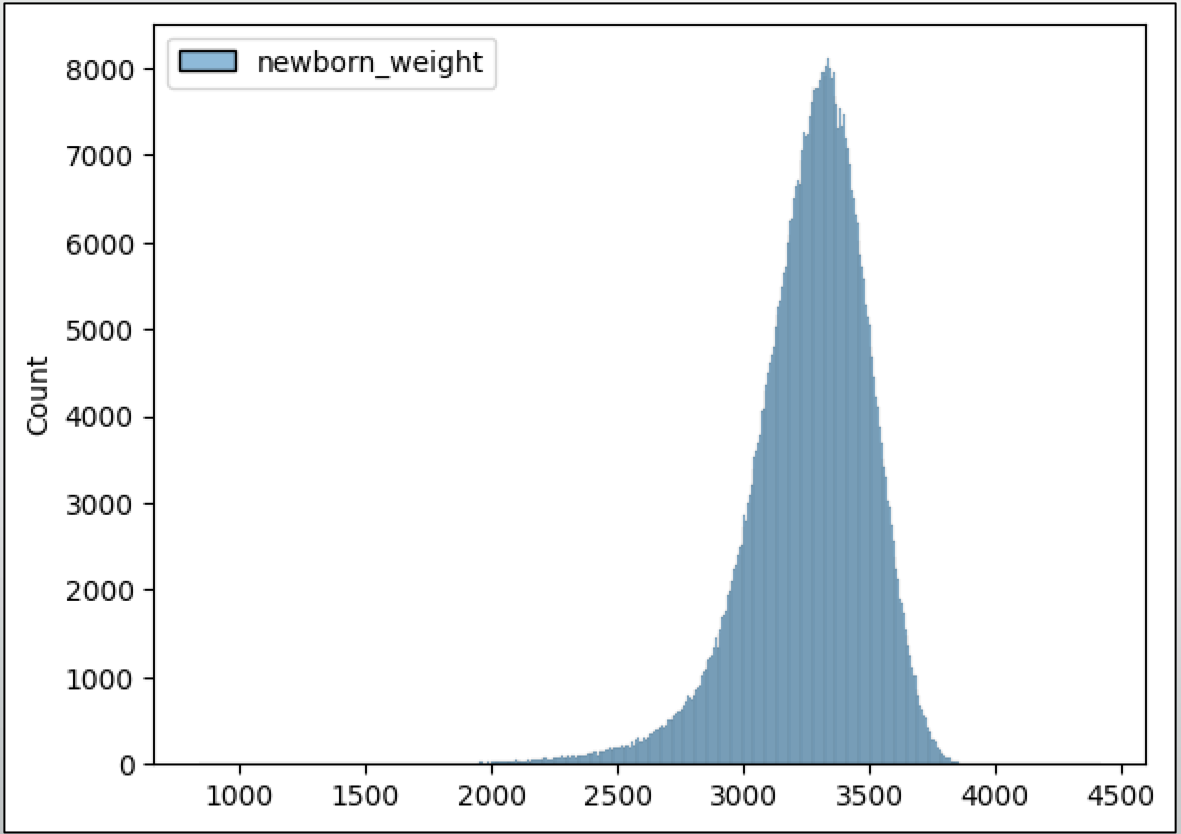
	Model	MAPE
0	XGBoost	15.59
1	Neural network	15.65
2	Random forest	15.66
3	Ridge	16.20
4	LinearRegression	16.20
5	Elastic Net	16.28
6	K-nearest neighbours	17.00
7	Decision tree	21.26

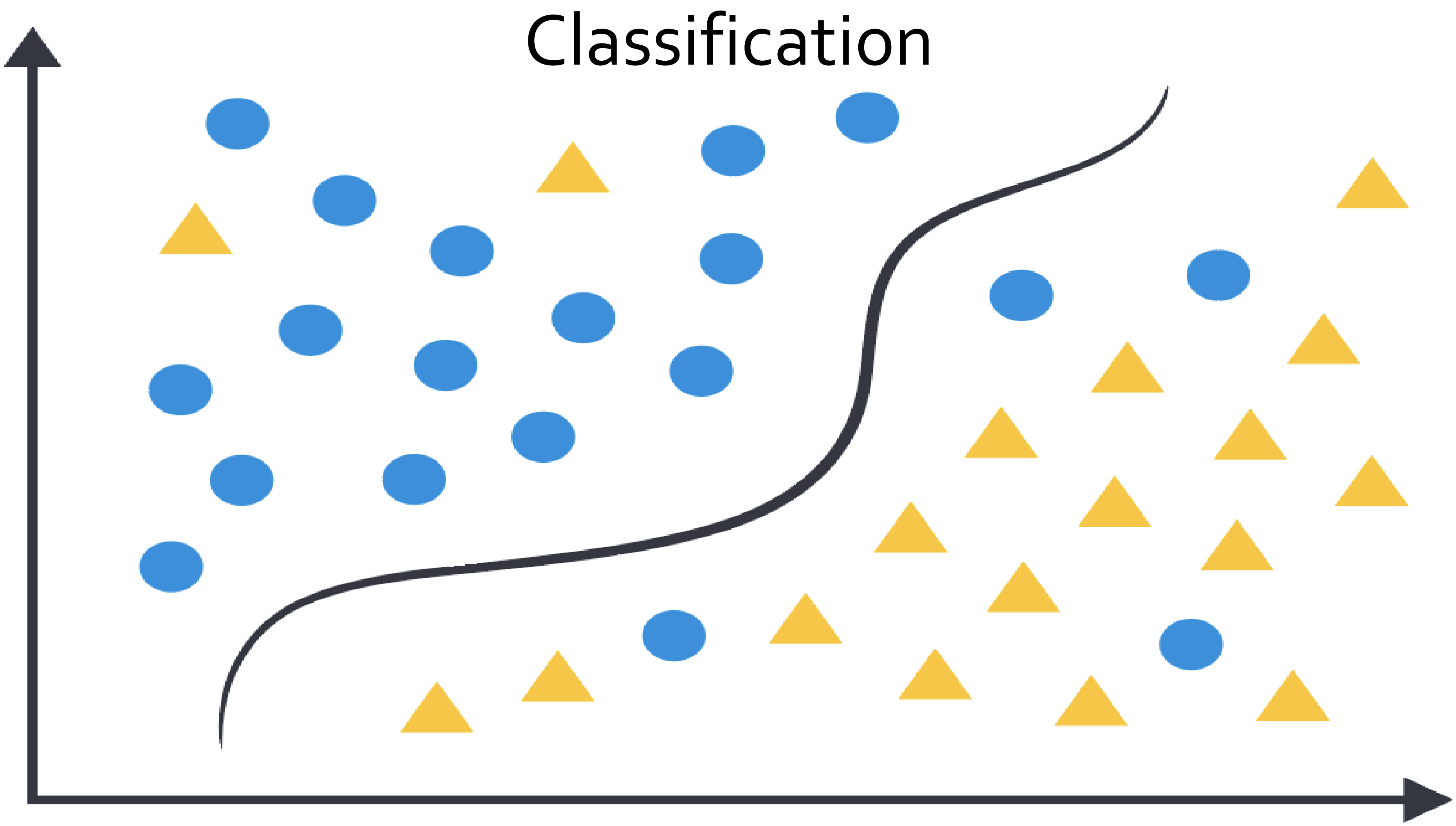
```
XGBRegressor
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=None, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, feature_types=None, gamma=0, gpu_id=None,
             grow_policy=None, importance_type='gain',
             interaction_constraints=None, learning_rate=0.05, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=0,
             max_depth=6, max_leaves=None, min_child_weight=1, missing=None,
             monotone_constraints=None, n_estimators=2000, n_jobs=3,
             nthread=None, num_parallel_tree=None, objective='reg:linear', ...)
```

Overall, **XGBoost** proved to be the most efficient, with a MAPE score of 15.59%.

The other models were dropped, with Decision Tree yielding the worst score of 21.26%.

XGBoost predictions




Classification

Quick glance at the data

	mean	std	min	50%	max
customer_id	550508.99	261237.66	100069.00	552548.00	999911.00
customer_age	46.32	8.00	26.00	46.00	73.00
customer_number_of_dependents	2.35	1.30	0.00	2.00	5.00
customer_relationship_length	35.93	7.99	13.00	36.00	56.00
customer_available_credit_limit	10036.34	17629.71	1438.30	4696.00	310644.00
total_products	4.15	3.18	1.00	4.00	36.00
period_inactive	2.34	1.01	0.00	2.00	6.00
contacts_in_last_year	2.46	1.11	0.00	2.00	6.00
credit_card_debt_balance	1162.81	814.99	0.00	1276.00	2517.00
remaining_credit_limit	7469.14	9090.69	3.00	3474.00	34516.00
transaction_amount_ratio	0.76	0.22	0.00	0.74	3.40
total_transaction_amount	5253.71	7402.26	510.00	3971.00	117159.00
total_transaction_count	64.86	23.47	10.00	67.00	139.00
transaction_count_ratio	0.82	0.62	0.00	0.71	16.25
average_utilization	0.27	0.28	0.00	0.18	1.00

Categorical Variables

	unique	top	freq
customer_sex	2	F	4838
customer_education	7	Graduate	3128
customer_civil_status	4	Married	4687
customer_salary_range	6	below 40K	3327
credit_card_classification	4	Blue	9436
account_status	2	open	8500

!size of train dataset: 10127 observations!

We can notice a presence of large outliers with the MAX and STD statistics of some variables, as well as high imbalance of our target variable.

Data preparation - missing values

1

	Missing Values	Percentage
customer_sex	1018	10.05
customer_salary_range	681	6.72
customer_age	624	6.16
total_transaction_amount	407	4.02

Only 4 variables contain missing values but in high volume, so it was decided to fill the NaNs.

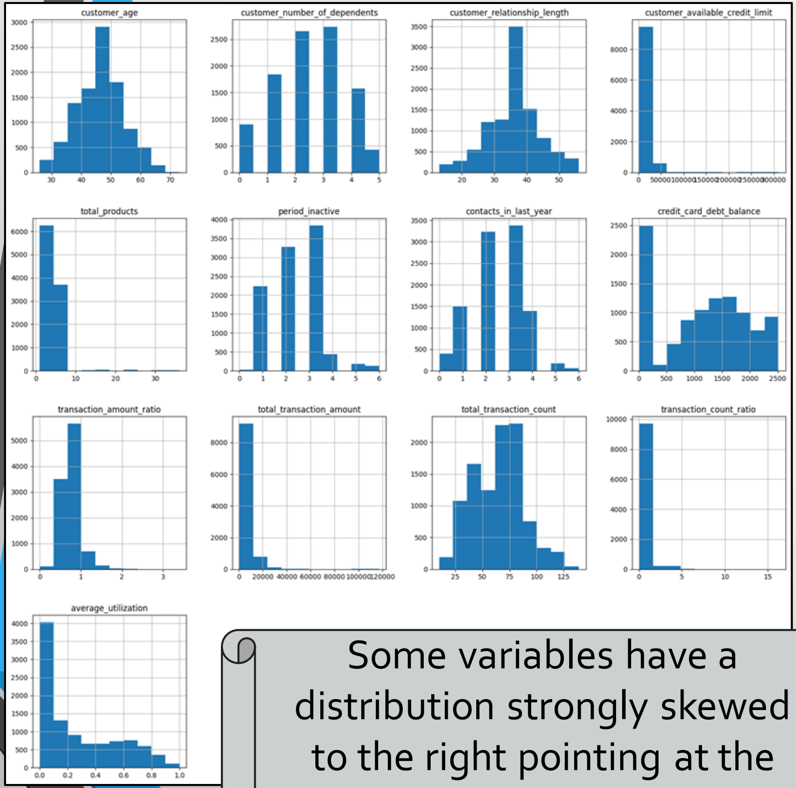
2

```
       total_transaction_amount  total_transaction_count
                            NaN                       63
103                         NaN                       27
107                         NaN                       81
116                         NaN                      121
146                         NaN                       35
...                         ...                      ...
9958                        NaN                       63
9979                        NaN                       32
10031                       NaN                       38
10081                       NaN                       78
10107                       NaN                      114

[407 rows x 2 columns]
```

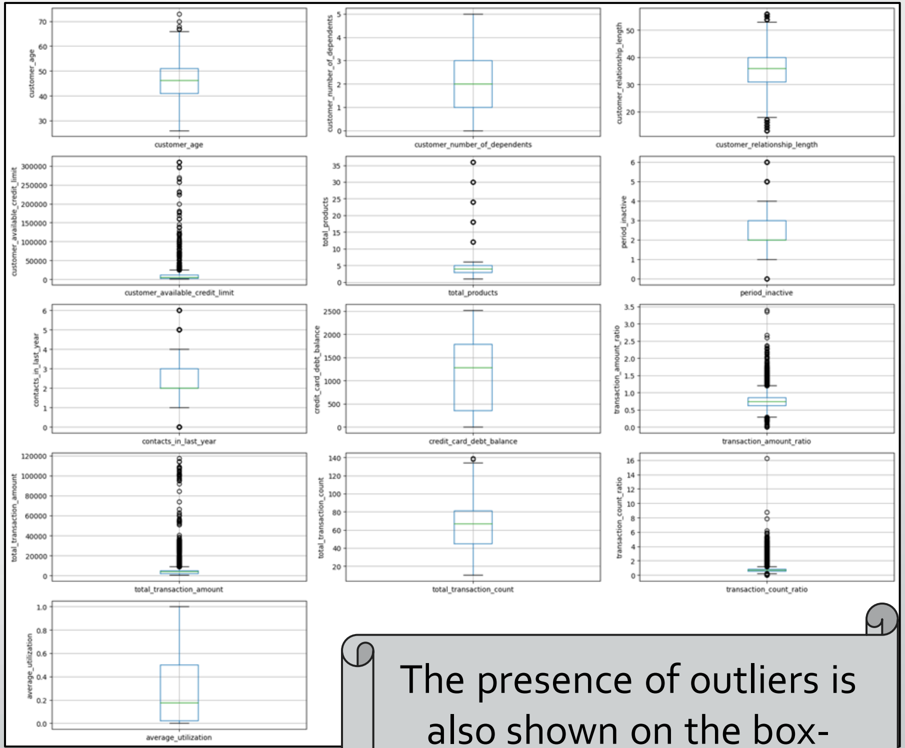
We suspected that some NaNs could be connected to other variables, but unfortunately that wasn't the case so we proceeded with standard way of filling the missing values.

Variable	Customer sex	Customer salary range	Customer age	Total transaction amount
NaNs replaced	New category (“Unknown”)	Mode (“below 40k”)	Mean	Mean

Data preparation - visualization of numerical variables



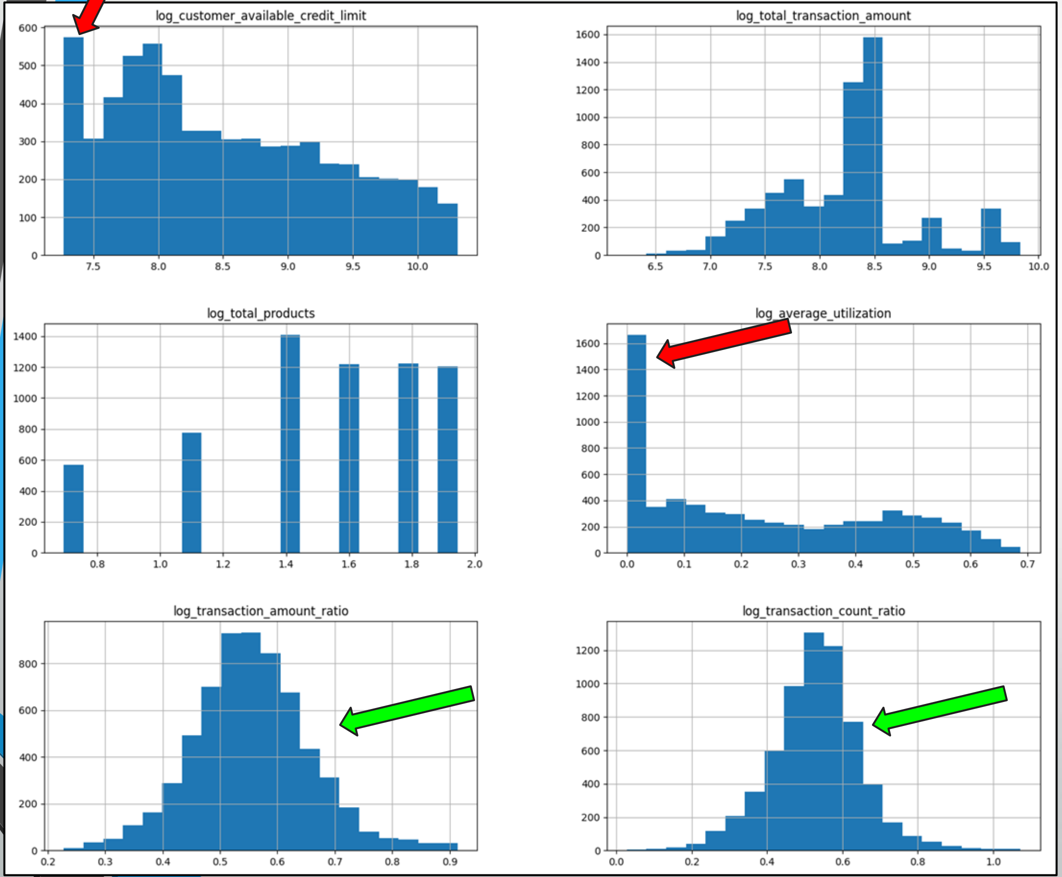
Some variables have a distribution strongly skewed to the right pointing at the outliers.



The presence of outliers is also shown on the box-plots.

Data preparation - visualization of categorical variables

Feature engineering - numerical variables



For some cases deleting the outliers with changing the values to the logarithm created a relatively normal distribution, but for some it couldn't deal with the high volume of values close to o.

Feature engineering - categorical variables

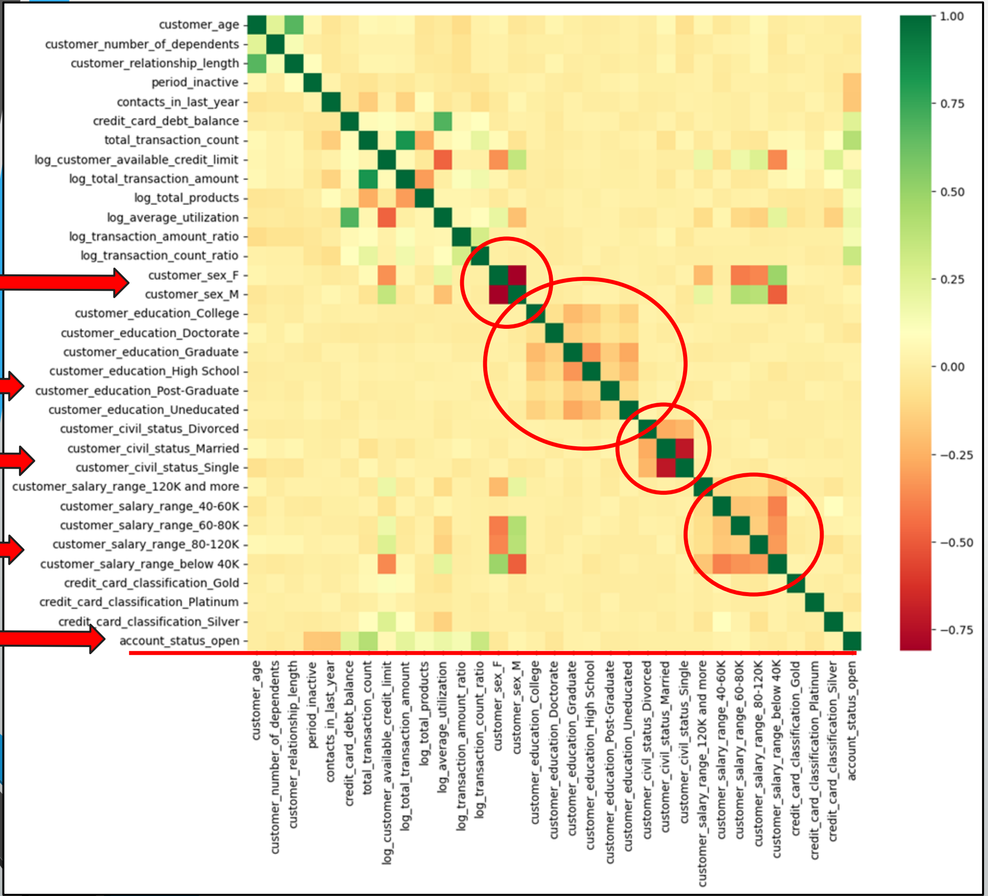
customer_age
customer_number_of_dependents
customer_relationship_length
period_inactive
contacts_in_last_year
credit_card_debt_balance
total_transaction_count
log_customer_available_credit_limit
log_total_transaction_amount
log_total_products
log_average_utilization
log_transaction_amount_ratio
log_transaction_count_ratio

\+

customer_sex_F
customer_sex_M
customer_education_College
customer_education_Doctorate
customer_education_Graduate
customer_education_High School
customer_education_Post-Graduate
customer_education_Uneducated
customer_civil_status_Divorced
customer_civil_status_Married
customer_civil_status_Single
customer_salary_range_120K and more
customer_salary_range_40-60K
customer_salary_range_60-80K
customer_salary_range_80-120K
customer_salary_range_below 40K
credit_card_classification_Gold
credit_card_classification_Platinum
credit_card_classification_Silver
account_status_open

We **One-Hot Encoded** all categorical variables deleting columns representing one category from each variable as we don't need to represent all of them to have a full information.

Correlation matrix and normalization



	mean	std	min	25%	50%	75%	max
customer_age	0.00	1.00	-2.94	-0.57	0.28	0.79	1.47
customer_number_of_dependents	-0.01	1.00	-2.01	-0.41	0.38	0.38	1.98
customer_relationship_length	0.01	1.00	-3.02	-0.52	0.36	0.50	3.29
period_inactive	0.01	1.01	-2.62	-0.28	-0.28	0.88	2.05
contacts_in_last_year	0.00	1.00	-2.31	-0.37	-0.37	0.60	1.57
credit_card_debt_balance	-0.00	1.00	-1.40	-1.40	0.14	0.76	1.69
total_transaction_count	-0.00	1.00	-2.51	-0.86	0.12	0.66	2.90
log_customer_available_credit_limit	0.00	1.00	-1.46	-0.81	-0.19	0.78	2.17
log_total_transaction_amount	-0.01	1.00	-3.23	-0.71	0.15	0.43	2.60
log_total_products	-0.00	1.00	-2.23	-0.35	0.26	0.75	1.17
log_average_utilization	-0.00	1.00	-1.12	-1.12	-0.24	0.91	2.21
log_transaction_amount_ratio	0.00	1.01	-3.20	-0.64	-0.04	0.61	3.50
log_transaction_count_ratio	0.00	1.00	-4.00	-0.55	0.03	0.57	4.52
customer_sex_F	0.01	1.00	-1.01	-1.01	0.99	0.99	0.99
customer_sex_M	-0.00	1.00	-0.80	-0.80	-0.80	1.25	1.25
customer_education_College	-0.00	0.99	-0.33	-0.33	-0.33	-0.33	3.01
customer_education_Doctorate	0.00	1.00	-0.21	-0.21	-0.21	-0.21	4.67
customer_education_Graduate	-0.01	0.99	-0.67	-0.67	-0.67	1.49	1.49
customer_education_High School	0.00	1.00	-0.50	-0.50	-0.50	-0.50	2.02
customer_education_Post-Graduate	0.01	1.02	-0.24	-0.24	-0.24	-0.24	4.21
customer_education_Uneducated	0.00	1.00	-0.41	-0.41	-0.41	-0.41	2.44
customer_civil_status_Divorced	0.01	1.01	-0.30	-0.30	-0.30	-0.30	3.38
customer_civil_status_Married	0.01	1.00	-0.91	-0.91	-0.91	1.10	1.10
customer_civil_status_Single	-0.01	1.00	-0.80	-0.80	-0.80	1.26	1.26
customer_salary_range_120K and more	0.01	1.02	-0.22	-0.22	-0.22	-0.22	4.57
customer_salary_range_40-60K	0.01	1.01	-0.46	-0.46	-0.46	-0.46	2.16
customer_salary_range_60-80K	0.00	1.01	-0.40	-0.40	-0.40	-0.40	2.52
customer_salary_range_80-120K	-0.01	0.99	-0.37	-0.37	-0.37	-0.37	2.68
customer_salary_range_below 40K	-0.01	1.00	-0.85	-0.85	-0.85	1.18	1.18
credit_card_classification_Gold	0.00	1.02	-0.07	-0.07	-0.07	-0.07	13.48
credit_card_classification_Platinum	-0.00	0.97	-0.03	-0.03	-0.03	-0.03	39.98
credit_card_classification_Silver	0.01	1.03	-0.18	-0.18	-0.18	-0.18	5.52

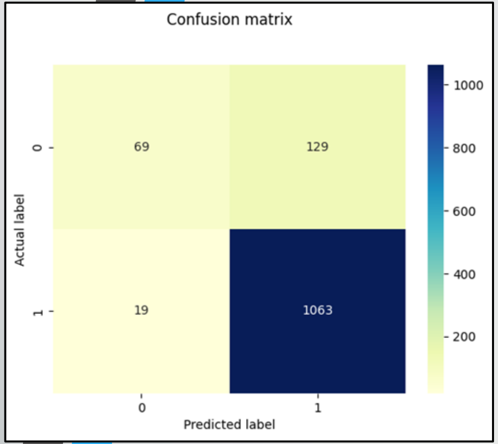
Training the models and finding best parameters with GridSearch

K-Nearest Neighbors

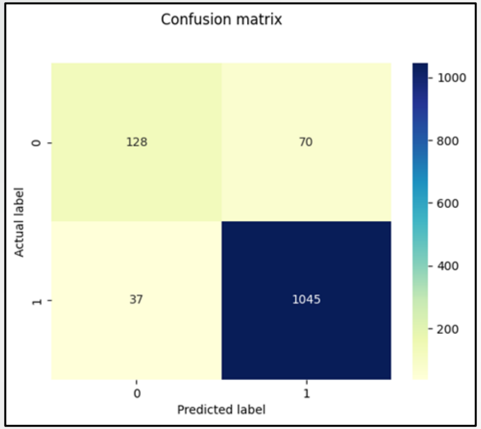
```python
parameter_space = {
    'n_neighbors':np.arange(8,20),
    "weights": ["uniform", "distance"],
    "algorithm": ["ball_tree", "kd_tree", "brute"],
    "leaf_size": [1,2,20,50,200]
}
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
clf = GridSearchCV(KNeighborsClassifier(), parameter_space, cv=cv, scoring="balanced_accuracy", n_jobs=4)

clf.fit(X_train, y_train)
print("Best parameters:")
print(clf.best_params_)

print("Best Score:" + str(clf.best_score_))
print("Best Parameters: " + str(clf.best_params_))
```

```
Best parameters:
{'algorithm': 'ball_tree', 'leaf_size': 1, 'n_neighbors': 8, 'weights': 'uniform'}
Best Score:0.6847075075564258
Best Parameters: {'algorithm': 'ball_tree', 'leaf_size': 1, 'n_neighbors': 8, 'weights': 'uniform'}
```

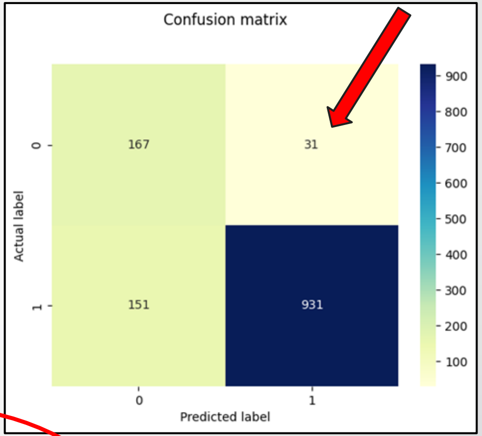
KNN



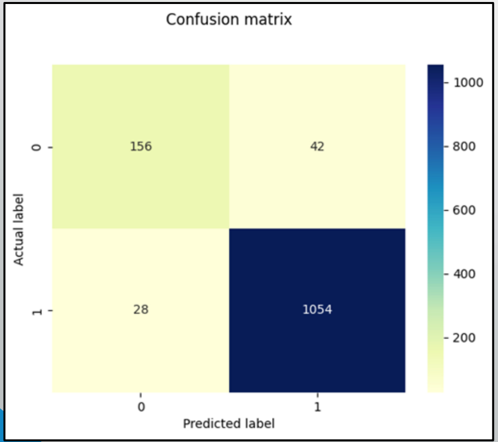
SVC



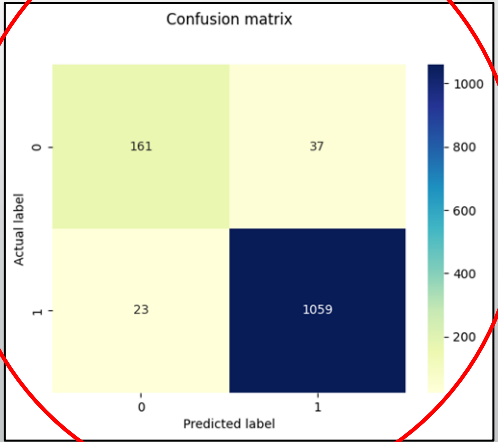
Logistic Regression



Random Forest



XGBoost



Comparing the models

	Balanced Accuracy	F1 Score	ROC AUC	|	Accuracy	Precision	Recall	Specificity
XGBoost	0.90	0.97	0.98	|	0.95	0.97	0.98	0.81
Random Forest	0.88	0.97	0.97	|	0.95	0.96	0.97	0.79
Logistic Regression	0.85	0.91	0.92	|	0.86	0.97	0.86	0.84
SVC	0.81	0.95	NaN	|	0.92	0.94	0.97	0.65
KNN	0.67	0.93	0.85	|	0.88	0.89	0.98	0.35

Accuracy score equal to 1 may suggest model being overfitted, however also high score on validation set only further confirm winning model **XGBoost** performance.

	Training Set	Validation Set	Null Accuracy
XGBoost	1.00	0.95	0.85
Random Forest	1.00	0.95	0.85
SVC	0.99	0.92	0.85
KNN	0.91	0.88	0.85
Logistic Regression	0.86	0.86	0.85

Thank you for your attention