**Term Deposit Prediction based on Banking Dataset of a Portuguese Banking Institution**

**Description of the problem**

Objective: To develop a predictive model that identifies potential customers who are likely to subscribe to a term deposit based on historical telephonic marketing campaign data. This will help the bank efficiently allocate resources by targeting the most promising customers, optimizing the effectiveness of their telephonic marketing campaigns and reducing costs.

**Context** (terminology that is not obvious):

Term Deposits - Fixed-term investments where customers deposit cash for a predetermined period in exchange for an agreed-upon rate of interest.

Telephonic Marketing Campaigns - Despite their effectiveness, these campaigns require substantial investments, including the hiring of large call centers. Efficient targeting of customers is crucial to ensure a higher conversion rate and cost-effectiveness.

**Key steps:**

* Data Preprocessing: Ensuring the dataset is clean and well-prepared for analysis.
* Exploratory Data Analysis (EDA): Gaining insights into the data and understanding the relationships between features and the target variable.
* Feature Engineering and Selection: Identifying and creating relevant features that enhance model performance.
* Model Building and Evaluation: Testing various classification algorithms and rigorously evaluating their performance using appropriate metrics.
* Optimization and Deployment: Fine-tuning the model for optimal performance and deploying it for real-time predictions.

**Data**

**Data source:** <https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets>

Data was provided by a Kaggle user, Prakhar Rathi.

**Exploratory Data Analysis**

The Data contains two sets: train.csv and test.csv with the ratio being 100% to 10%.

Train set contains 45,211 rows, while test set contains 4521. The test set was randomly extracted from the train set.

### Customer Attributes

1. Age: Age of the customer (numeric).
2. Job: Type of job that a customer currently does (categorical):
   * "admin."
   * "unknown"
   * "unemployed"
   * "management"
   * "housemaid"
   * "entrepreneur"
   * "student"
   * "blue-collar"
   * "self-employed"
   * "retired"
   * "technician"
   * "services"

"Unknown" value means that we do not know what kind of job a customer has.

1. Marital: Marital status of the client (categorical):
   * "married"
   * "divorced" (may also mean "widowed")
   * "single"
2. Education: Level of education of the customer (categorical):
   * "unknown"
   * "secondary"
   * "primary"
   * "tertiary"
3. Default: Has credit in default? (binary):
   * "yes"
   * "no"
4. Balance: Average yearly balance in euros (numeric).
5. Housing: Has housing loan? (binary):
   * "yes"
   * "no"
6. Loan: Has personal loan? (binary):
   * "yes"
   * "no"

### Related with the Last Contact of the Current Campaign

1. Contact: Contact communication type (categorical):
   * "unknown"
   * "telephone"
   * "cellular"
2. Day: Last contact day of the month (numeric).
3. Month: Last contact month of year (categorical):

* "jan"
* "feb"
* "mar"
* …
* "nov"
* "dec"

1. Duration: Last contact duration, in seconds (numeric).

### Other Attributes

1. Campaign: Number of contacts performed during this campaign and for this client (numeric, includes last contact).
2. Pdays: Number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted).
3. Previous: Number of contacts performed before this campaign and for this client (numeric).
4. Poutcome: Outcome of the previous marketing campaign (categorical):

* "unknown"
* "other"
* "failure"
* "success"

### Target Variable

* y: Whether the client subscribed to a term deposit (binary):
  + "yes"
  + "no"

Description of the provided dataset:

“Age” column contains values ranging from 18 to 95 years.

“Job” contains all the values described above.

“Marital” – all values.

“Education” – all values.

“Default” – both values.

“Balance” – ranges from -8019 to 102,127.

“Housing” – both.

“Loan” – both.

“Contact” – all values.

“Day” and “Month” – all values.

“Duration” – ranges from 0 to 4918.

“Campaign” – from 1 to 63.

“Pdays” – -1 to 871.

“Previous” – 0 to 275.

“Poutcome” – all values.

“Y” – both values in every row.

In conclusion, every column of every row has a value in the train set.

**Handling “unknown” value**

Since “unknown” obviously means that we do not know that information about a client, we could treat as a missing value, hence there is a need to handle it somehow. We have decided that the best way to do that for the following columns would be:

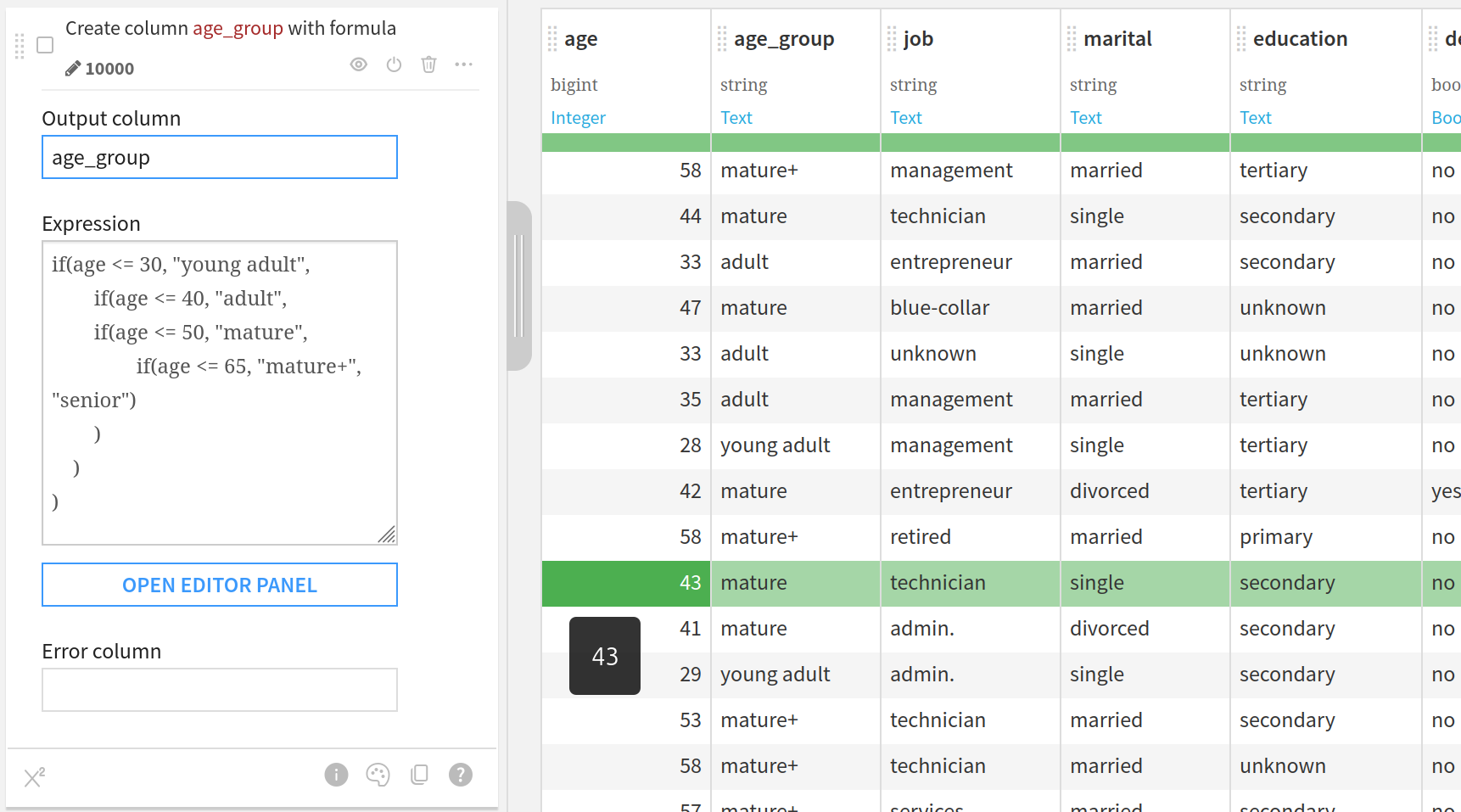
“Contact” – fill “unknown” with “cellular” since it is the most common value (64.8%).

“Poutcome” – since most values are “unknown” (81.7%), we could drop that column entirely.

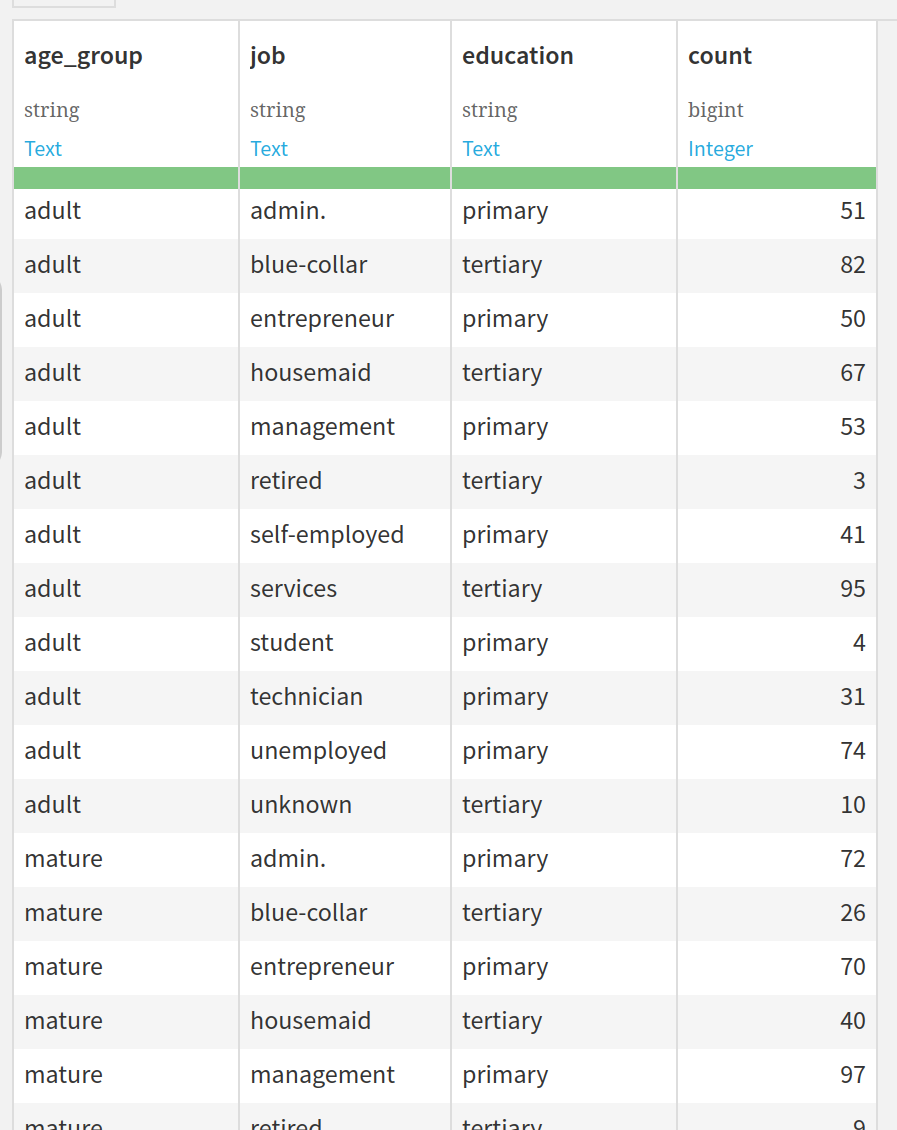
“Job” – “unknown” value could mean a lot of things, there is no obvious way to determine what is an average job of a person. It could still be important to not drop rows that contain it and also to not change based on some rule, so we have decided to leave it as it is.

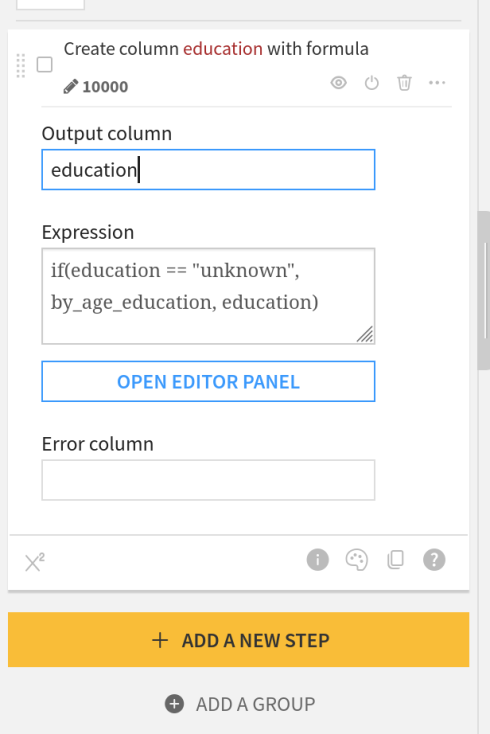
“Education” – “unknown” value is present in 4.1% of all the rows. That could be significant, so we have decided to interpolate values based on “age” and “job”.

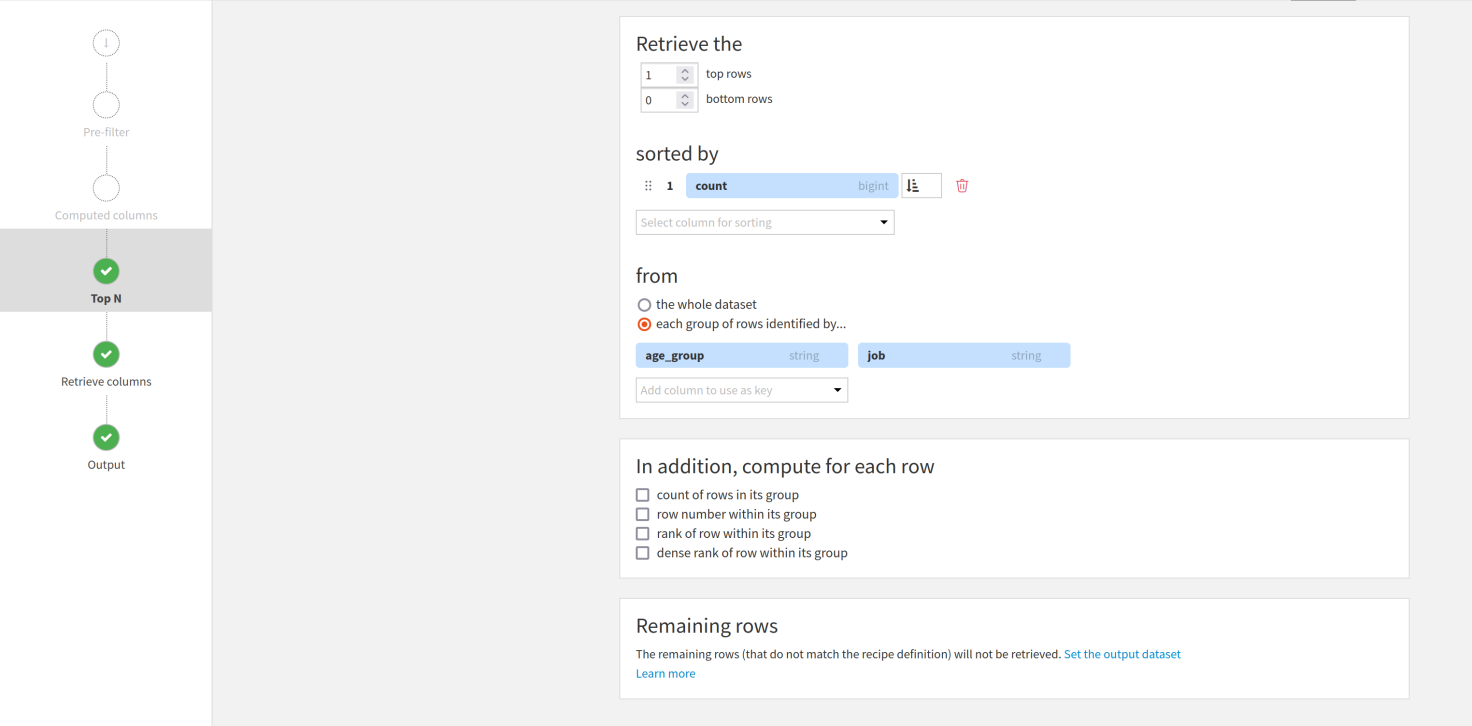
These are the steps that we did in order to achieve what is described above:

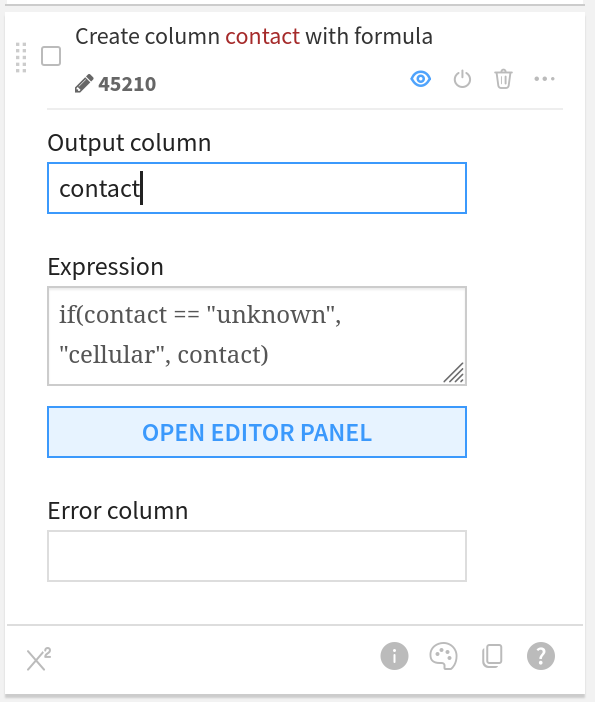
1) Divide the rows into age groups (*age of retirement in Portugal is 66*): 18 – 30 (young adult), 31 – 40 (adult), 41 – 50 (mature), 50 – 65 (mature plus), 66 – 95 (senior):

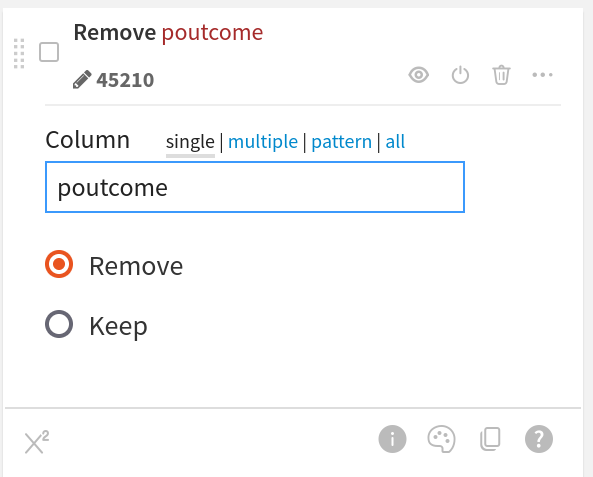
2) Filter “unknown” and group data by age group, education and job:



3) Assign top education values for specific age and job groups to “unknown” education:



4) Change “unknown” contact to “cellular”.

5) Drop “Poutcome”

The resulting dataset does not contain “unknown” values in “education” and the “poutcome” column.

**Handling outliers**

Among all attributes we have to spot outliers in numerical ones. In order to do that we calculate the interquartile range. Generally, a data point is an outlier if it is over 1.5 times the IQR below the first quartile or 1.5 times the IQR above the third quartile.

For balance the outliers are values below -627 and above 13163.64; for duration – values above 1269; for campaign – above 13; for pdays – values above 370; for previous – values above 9.

Spotted outliers are then assigned with border values.

**Feature Engineering**

“Age Group” column was added: 18 – 30 (young adult), 31 – 40 (adult), 41 – 50 (mature), 50 – 65 (mature plus), 66 – 95 (senior).

**A way to solve the problem**

**Selected algorithms**

The most common methods for binary classification are Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machine, Naive Bayes, or more sophisticated methods, such as Random Forest or Neural Network. k-Nearest Neighbors does not work well with big datasets and with many dimensions.

We would like to try the next two algorithms: Logistic Regression and Random Forest.

***Random Forest***

*A Random Forest is made of many decision trees. Each tree in the forest predicts a record, and each tree "votes" for the final answer of the forest.  
The forest chooses the class having the most votes.*

*A decision tree is a simple algorithm which builds a decision tree. Each node of the decision tree includes a condition on one of the input features.*

*When "growing" (ie, training) the forest:*

* *for each tree, a random sample of the training set is used;*
* *for each decision point in the tree, a random subset of the input features is considered.*

*Random Forests generally provide good results, at the expense of "explainability" of the model.*

***Logistic Regression***

*Despite its name, Logistic Regression is a classification algorithm, using a linear model (i.e., it computes the target feature as a linear combination of input features).*

*Logistic Regression minimizes a specific cost function (called* logit *or* sigmoid *function), which makes it appropriate for classification.   
A simple Logistic regression algorithm is prone to overfitting and sensitive to errors in the input dataset. To address these issues, it is possible to use a* penalty *(or* regularization term*) to the weights.   
This implementation can use either 'L1' or 'L2' regularization terms.*

**Preparation For Training**

Test.csv turned out to be included in train.csv, so we only used train.csv for modelling and evaluation, because Dataiku actually does the required split of data automatically. We just had to provide the ratio of that split.

We divided dataset 80% to 20%, train to test respectively.

In our case we do not want a lot of false positive predictions, because the goal is to minimize costs of telephonic marketing campaigns, hence we do not the bank making unnecessary calls. However, we also do not want a lot of false negatives, because that way the bank would lose potential profit. Considering both these facts we have to find a balance between precision and recall. A good metrics to aim for would be **accuracy and F1-score**.

***Accuracy*** *- a metric that measures how often a machine learning model correctly predicts the outcome. You can calculate accuracy by dividing the number of correct predictions by the total number of predictions.\*

***F1-score*** *- an alternative evaluation metric in machine learning is the F1 score, which assesses the predictive ability of a model by examining its performance on each class individually rather than considering overall performance like accuracy does. The F1 score combines two competing metrics, precision and recall.*

***Precision*** *- a metric that measures how often a machine learning model correctly predicts the positive class. You can calculate precision by dividing the number of correct positive predictions (true positives) by the total number of instances the model predicted as positive (both true and false positives).*

***Recall*** *- a metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset. You can calculate recall by dividing the number of true positives by the number of positive instances. The latter includes true positives (successfully identified cases) and false negative results (missed cases).*

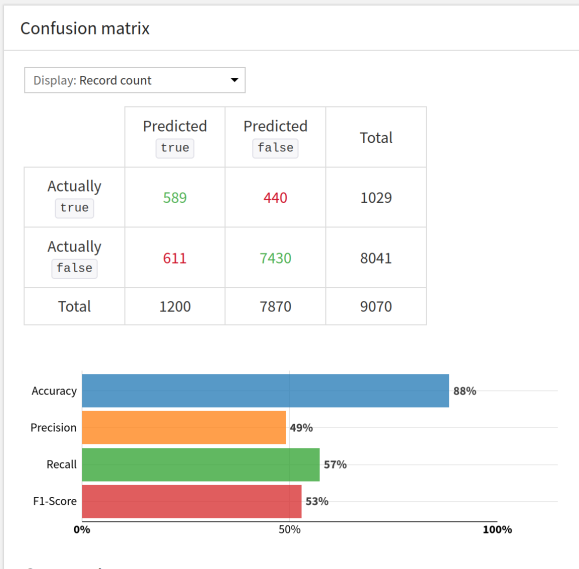
In Dataiku we chose options to optimize hyperparameters for accuracy and optimize threshold for F1-score.

We used min-max rescaling on numerical attributes.

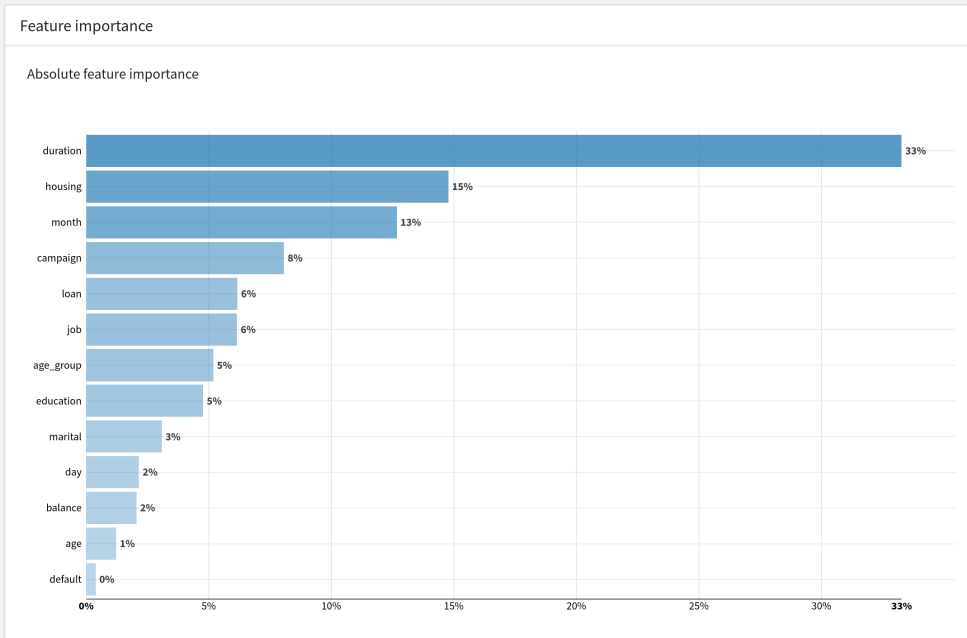
Choosing features

Before starting the first session, we rejected attributes pdays, previous and contact, because they have too many equal values.

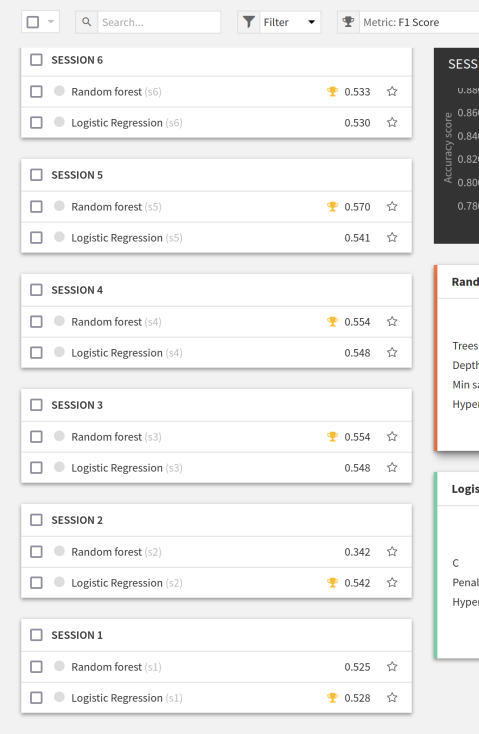
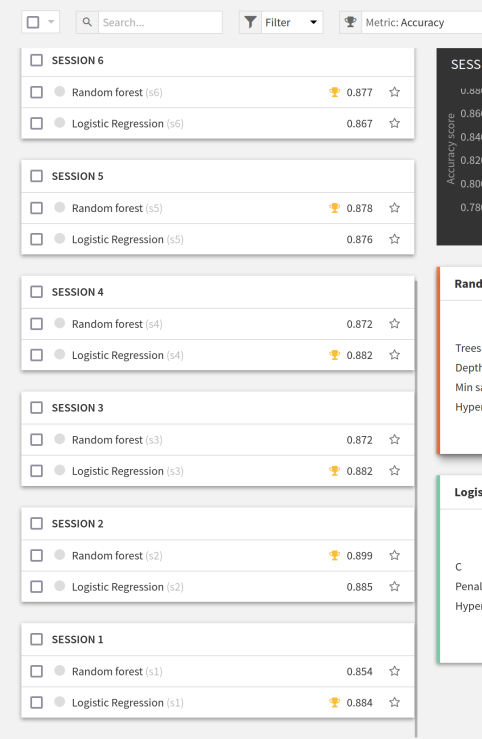
After the first session Logistic Regression achieved higher accuracy and F1-score (0.884 and 0.528).



We decided to drop features like age and default, because their importance turned out to be very low.

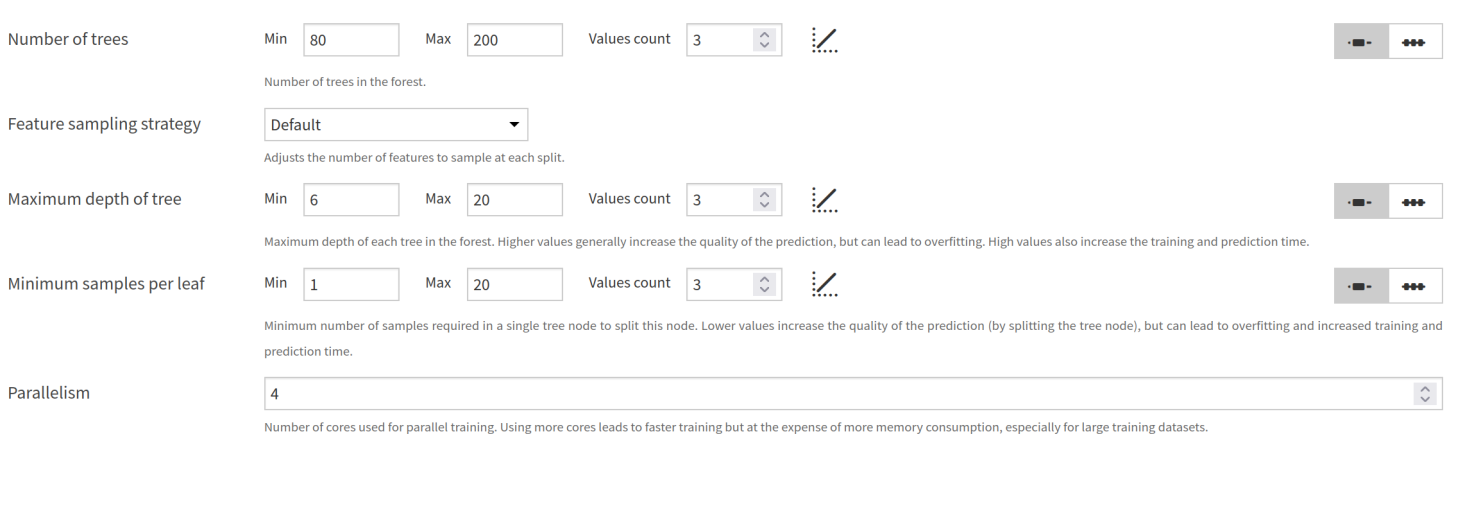


Also after several more session a decision to drop age group was made, because it helped us to achieve better results.

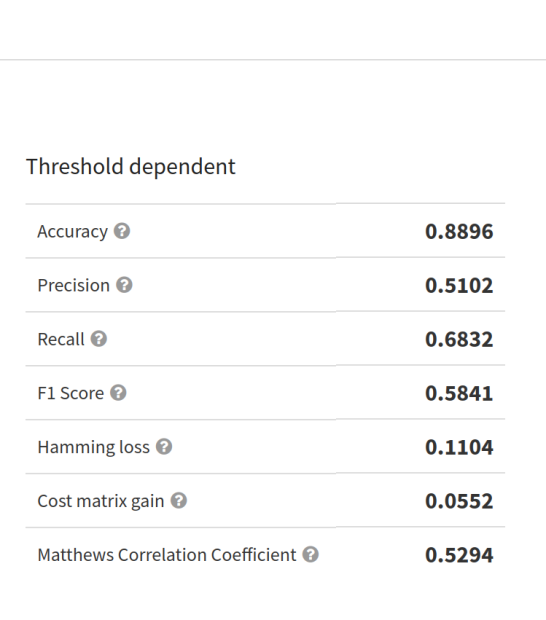


As you can see, **session number 5** showed us that random forest with dropped attributes provided us with a better result of F1-score. Accuracy in this case is also good enough.

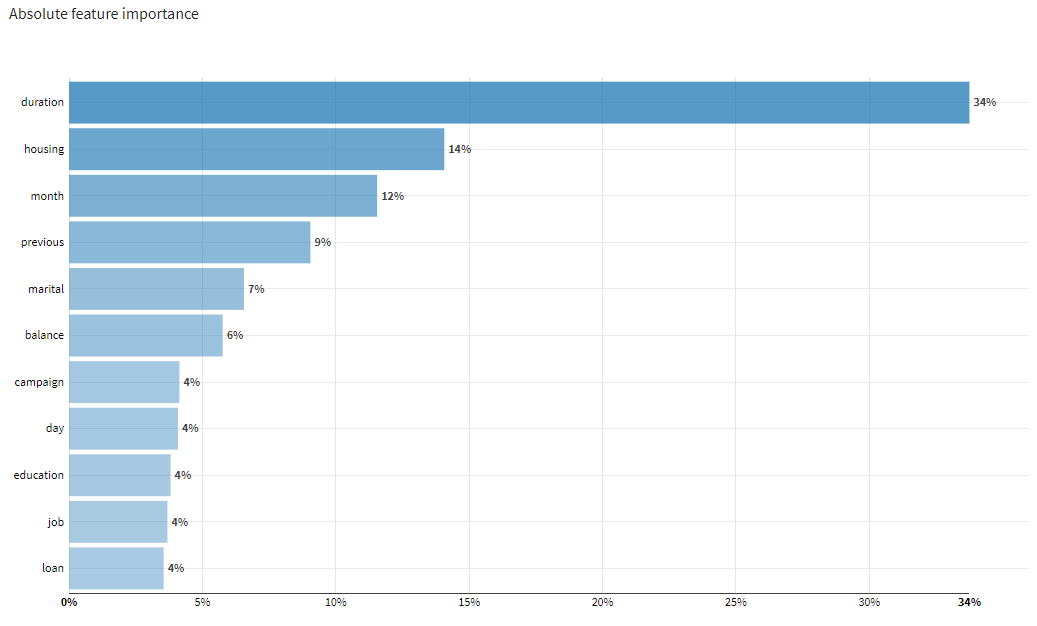
After that we chose an interval in which dataiku would search optimal hyper-parameters.

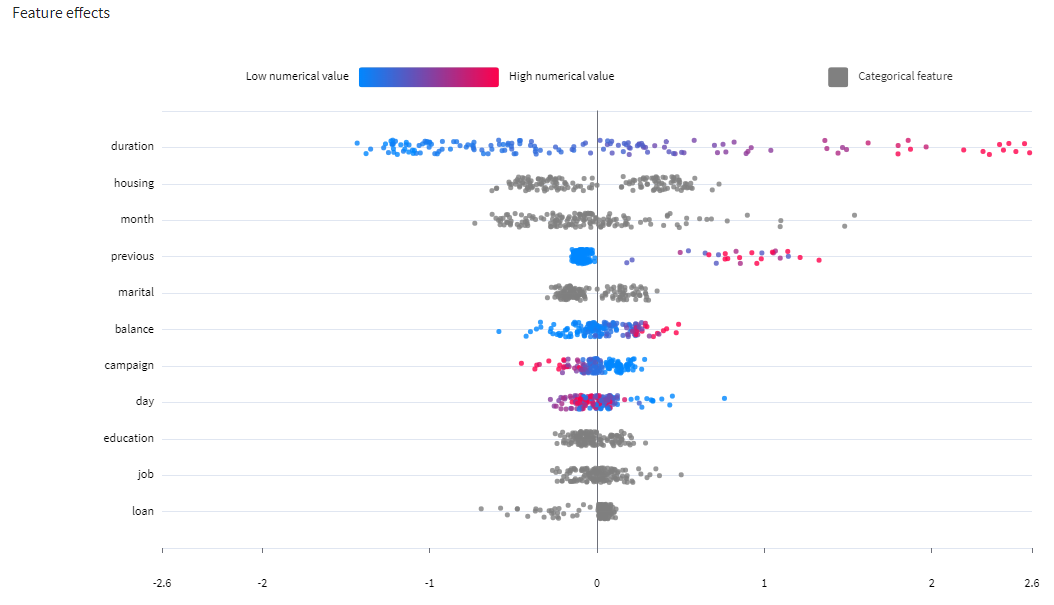
Values count set to 3 in each of them means that it would undergo 27 searches (3x3x3).

The best results were achieved with number of trees set to 200, max trees depth set to 20, min samples per leaf set to 10. In this case **accuracy is 0.8896 and F1-score is 0.5841**, which is the best result among all of them.

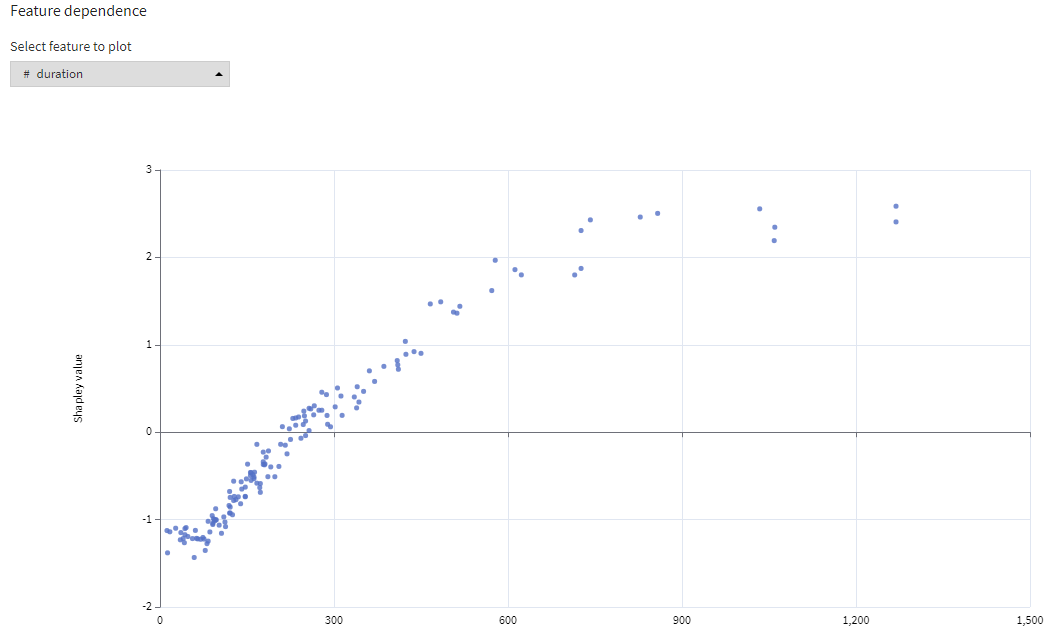
We treat this model as good enough for our purposes.

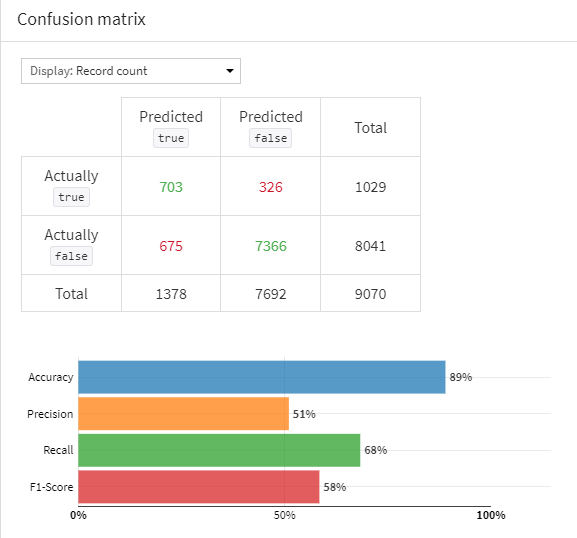
**Discussion of the results and evaluation of the model**

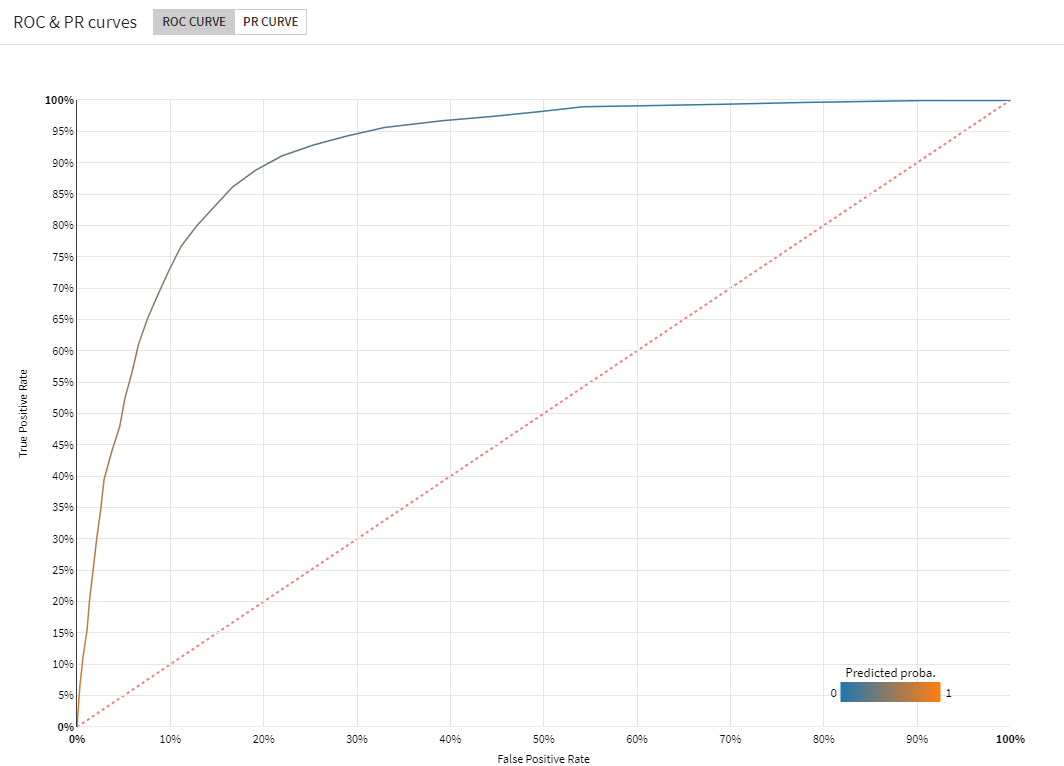
In the selected (most effective for our purposes) model the feature importance is following:

Feature effects:

Among all of the features “Duration”, “Housing”, “Month” and “Previous” should be included in future models.

Dependence of the most important feature:

Confusion matrix and basic metrics:

ROC curve:

Overall, **threshold independent** metrics seem good:

Lift at 40% - 2.39;

Log loss – 0.35;

ROC-AUC score – 0.914;

Average Precision score – 0.5452;

Calibration loss – 0.1759

**Threshold dependent (cutoff at 0.65):**

Accuracy – 0.8896;

Precision – 0.5102;

Recall – 0.68.32;

F1-score – 0.5841;

Hamming loss – 0.1104;

Cost matrix gain – 0.0552;

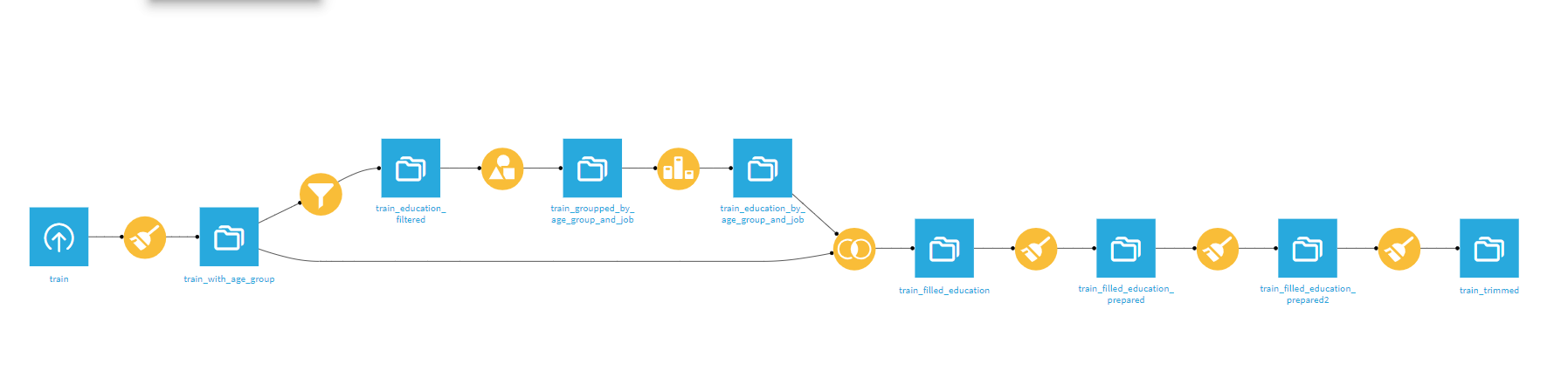
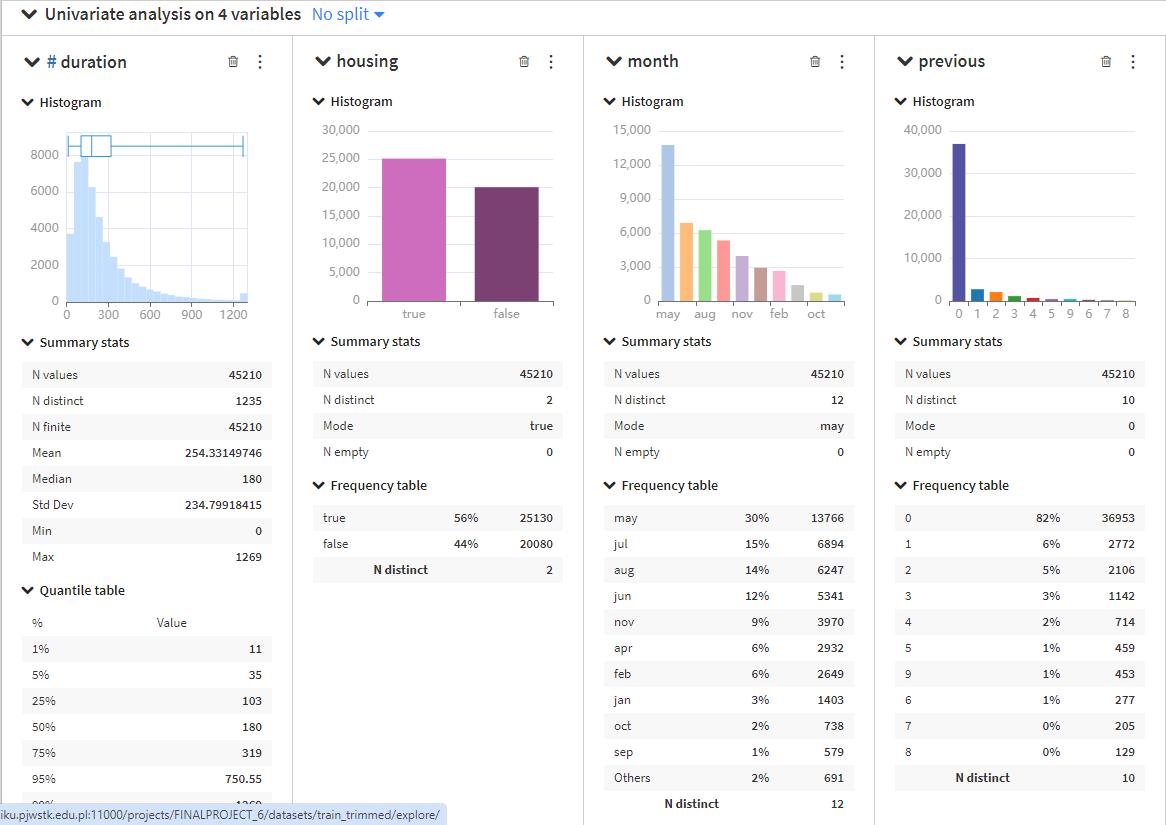
Matthews correlation coefficient – 0.5294.

**Summary**

The results of appear to be quite satisfactory. The provided data was already very clean, however some preparation was done nonetheless (although, some prepared columns were dropped due to containing a lot of equal values).

The problem was quite interesting to solve, but at the same time we think that solving problems of modern banking would require collecting more data about every single customer.

On a more subjective note, working with already clean data, having atomic values andno missing data, felt more rewarding and easy.

**Attachments:**

