Data Glacier

Your Deep Learning Partner

# Final Report

## Bank Marketing Campaign

**Jan 15, 2023**

Group Name: Evolve Data
Name: Dmitry Sharukhin
Email: sharuhinda@gmail.com
Country: Russia
College/Company: Finval GC
Specialization: Data Science

# Agenda

Problem Description

Summary of EDA Results

Dataset Preprocessing and Encoding

Models and Results

Alternative Approach: Clustering

Conclusions and Recommendations

# Problem Description

**Client:**
ABC Bank

**Description:**
ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution)

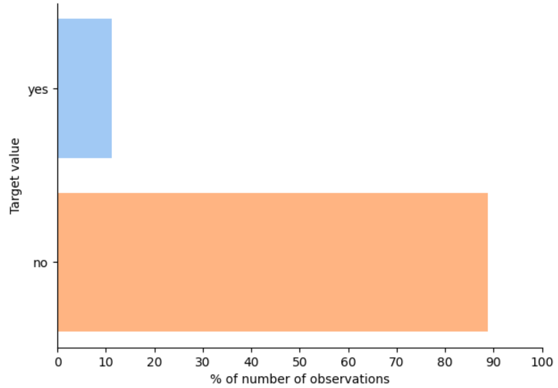
Bank wants to use ML model to shortlist customer whose chances of buying the product are more so that their marketing channel (tele marketing, SMS/email marketing etc.) can focus only to those customers in order to save resources (which is directly involved in the cost (resource billing)) and their time

**Objectives:**

- Create ML model to shortlist customer whose chances of buying the product are more so that their marketing channel (tele marketing, SMS/email marketing etc.) can focus only to those customers

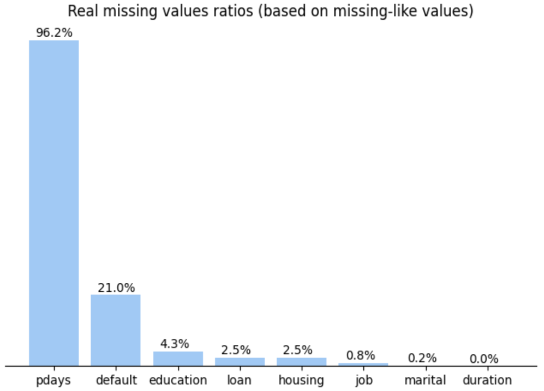
Data Glacier
Your Deep Learning Partner

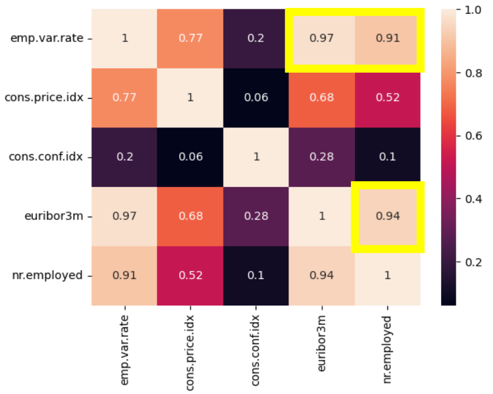
# Summary of EDA Results



We are dealing with **HIGHLY imbalanced** dataset (positive labels << negative labels)

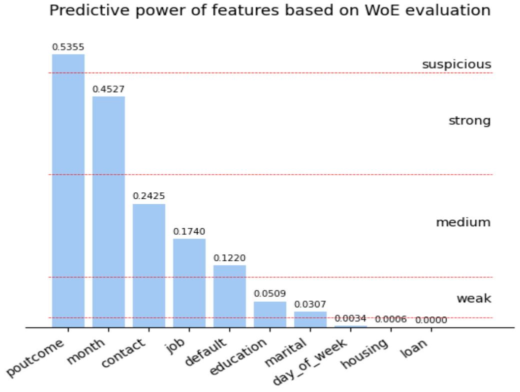
There are **many missing-like values**





There are some correlated features which can help to impute missing values or make troubles when modelling

Weight-of-Evidence was chosen as one of encoding approaches to try

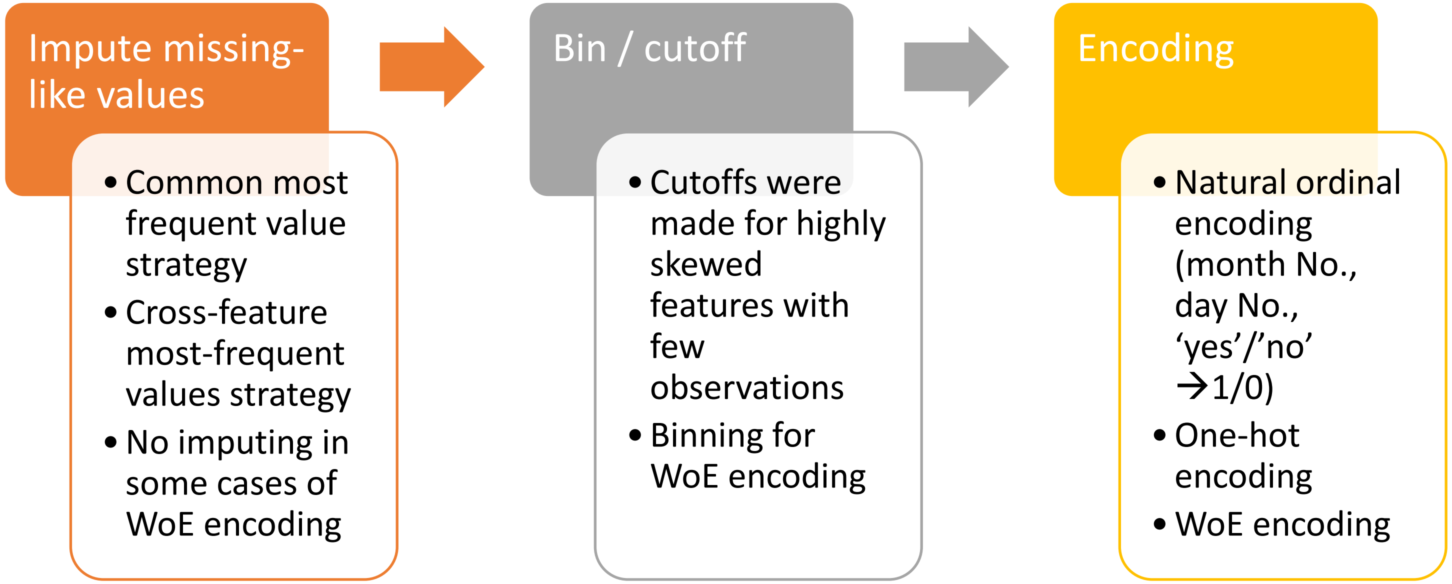


Data Glacier

Your Deep Learning Partner

# Dataset Preprocessing and Encoding

As the dataset is highly imbalanced, all preprocesses and encodings did **not drop any row**
(but redundant columns were dropped if necessary)



### Impute missing-like values

- Common most frequent value strategy
- Cross-feature most-frequent values strategy
- No imputing in some cases of WoE encoding

### Bin / cutoff

- Cutoffs were made for highly skewed features with few observations
- Binning for WoE encoding

### Encoding

- Natural ordinal encoding (month No., day No., 'yes'/'no' →1/0)
- One-hot encoding
- WoE encoding

**11 variants of datasets** were created by applying different techniques to different features and combining them

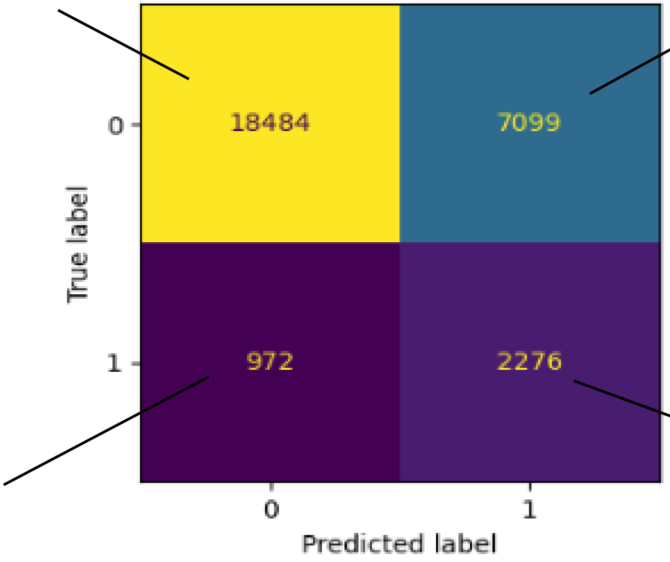
**Pipelines** were created to process initial test dataset according to each option's rules

# Models and Results

## Choosing metric

Clients that **will NOT be called** during campaign **and refuse** to open deposit if they would

Clients that **will be called** during campaign **but refuse** to open deposit **(HIGHER COST OF CAMPAIGN)**



Clients that **will NOT be called** during campaign **but agree** to open deposit if they would **(LOST INCOME)**

Clients that **will be called** during campaign **and agree** to open deposit

Based on assumption that potential income in positive case is significantly higher than cost per call, **the target is to reduce False Negatives.**

**RECALL is the target metric** then

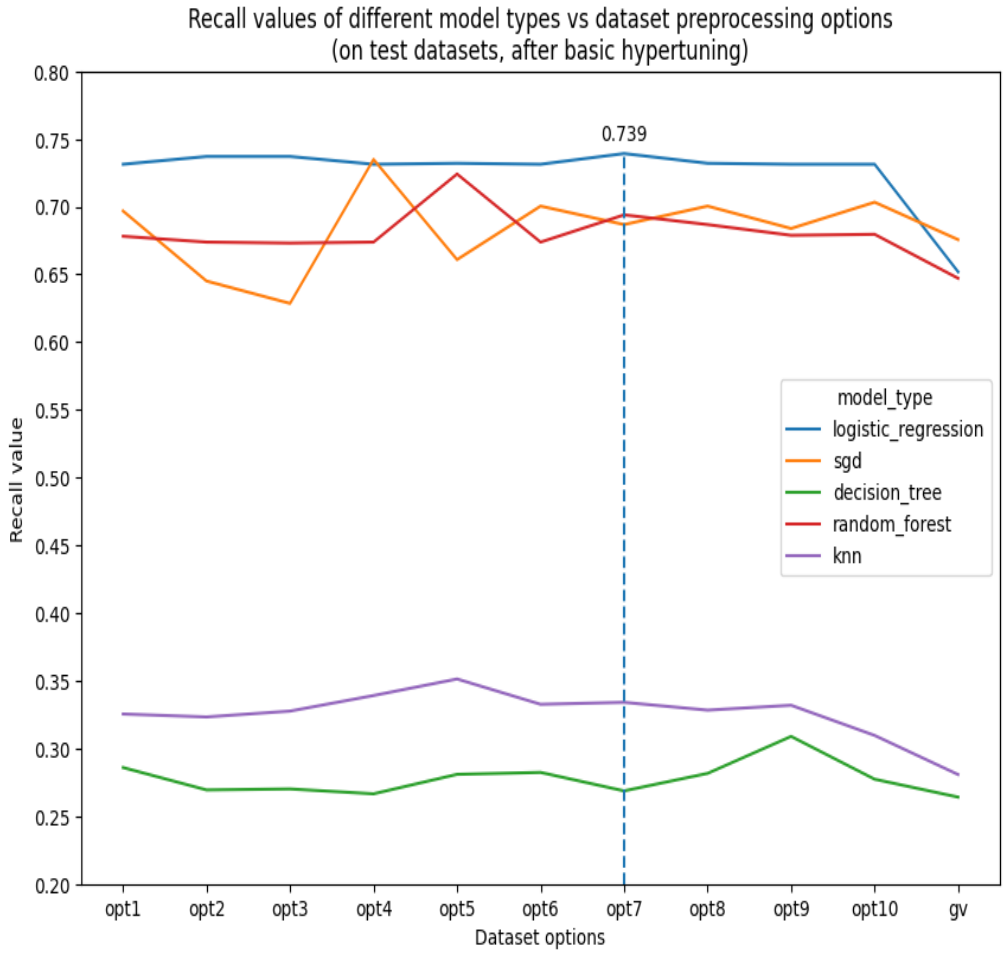
# Models and Results (continue)

5 models were tested on each version of dataset created:

- Logistic Regression
- Stochastic Gradient Descent
- Decision Tree
- Random Forest
- k Nearest Neighbors

Basic hyperparameters tuning was performed on each model using *GridSearchCV* class from *scikit-learn* package with 5-fold stratified cross-validation.

Recall was assigned as optimization metric



Recall values of different model types vs dataset preprocessing options (on test datasets, after basic hypertuning)

Logistic Regression shows best results on almost all dataset variants.

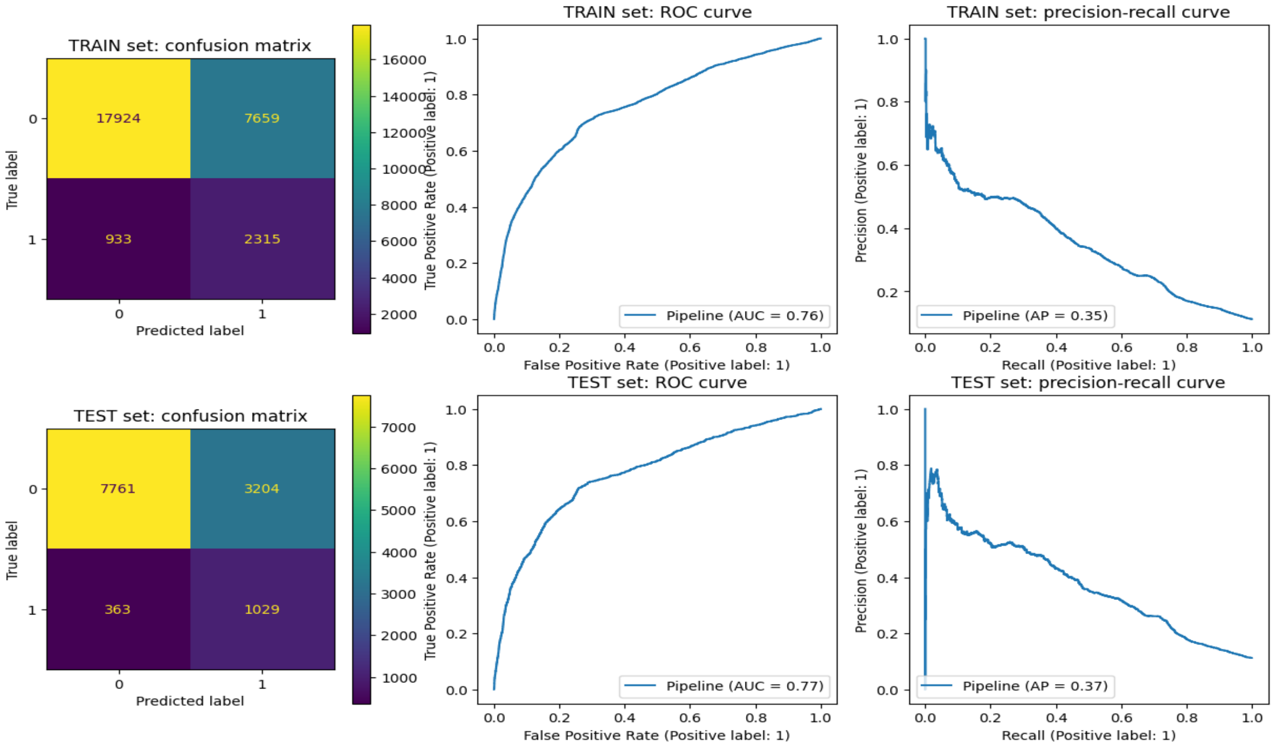
And it’s robust against preprocessing and encoding options

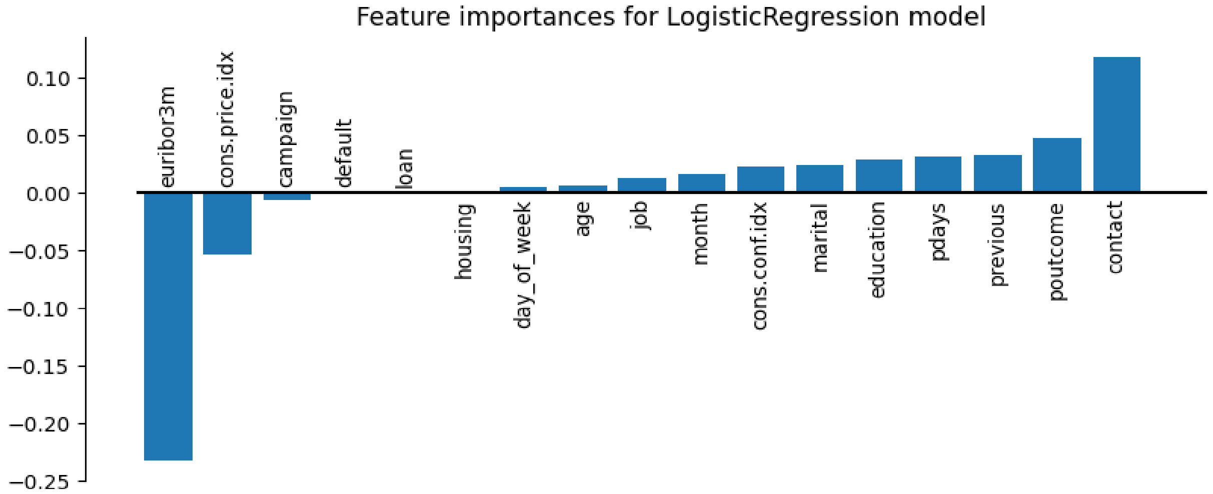
Data Glacier

Your Deep Learning Partner

# Models and Results (continue)

## Best model results





Data Glacier

Your Deep Learning Partner

# Models and Results (continue)

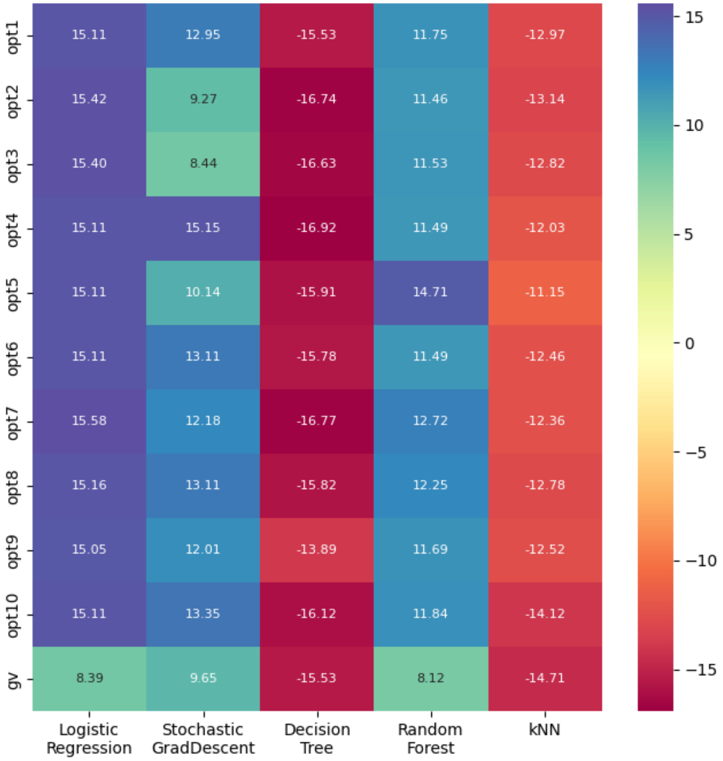
Data Glacier
Your Deep Learning Partner

## Evaluation of financial results

Information to evaluate campaign financial results for each model type was found in Internet

| Indicator | Value |
|---|---|
| Average bank margin | 1%/yr |
| Average deposit sum | $31 600 |
| Mobile call cost | $0.15 /minute |
| Average call duration | 3.4 minutes<br>(mean value from dataset) |
| Average margin from 1 deposit per month (possible duration of the campaign) | 31 600 * 0.01 / 12 ~ $26.3 /mo |
| Average cost of call | 3.4 * 0.15 ~ $0.5 |

Estimated campaign financial result
depending on model type and preprocessing used (in $1000)



|  | Logistic Regression | Stochastic GradDescent | Decision Tree | Random Forest | kNN |
|---|---|---|---|---|---|
| opt1 | 15.11 | 12.95 | -15.53 | 11.75 | -12.97 |
| opt2 | 15.42 | 9.27 | -16.74 | 11.46 | -13.14 |
| opt3 | 15.40 | 8.44 | -16.63 | 11.53 | -12.82 |
| opt4 | 15.11 | 15.15 | -16.92 | 11.49 | -12.03 |
| opt5 | 15.11 | 10.14 | -15.91 | 14.71 | -11.15 |
| opt6 | 15.11 | 13.11 | -15.78 | 11.49 | -12.46 |
| opt7 | 15.58 | 12.18 | -16.77 | 12.72 | -12.36 |
| opt8 | 15.16 | 13.11 | -15.82 | 12.25 | -12.78 |
| opt9 | 15.05 | 12.01 | -13.89 | 11.69 | -12.52 |
| opt10 | 15.11 | 13.35 | -16.12 | 11.84 | -14.12 |
| gv | 8.39 | 9.65 | -15.53 | 8.12 | -14.71 |

Each model’s confusion matrix was built on test sets.
Information from table above was transformed to similar form and estimated financial result was calculated.
It shows that Logistic Regression model is leading not only on recall value but also on financial results

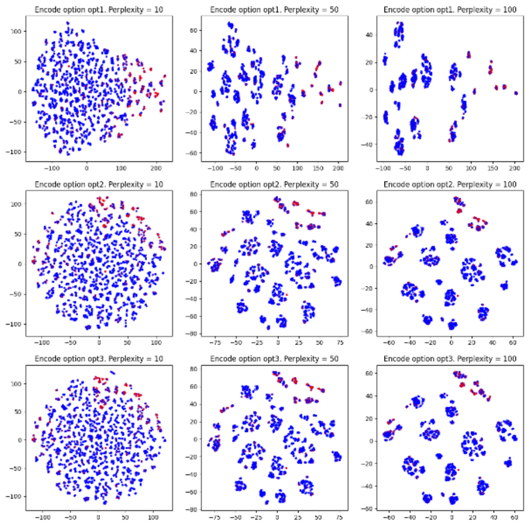
Financial result of each outcome type (per 1 call)

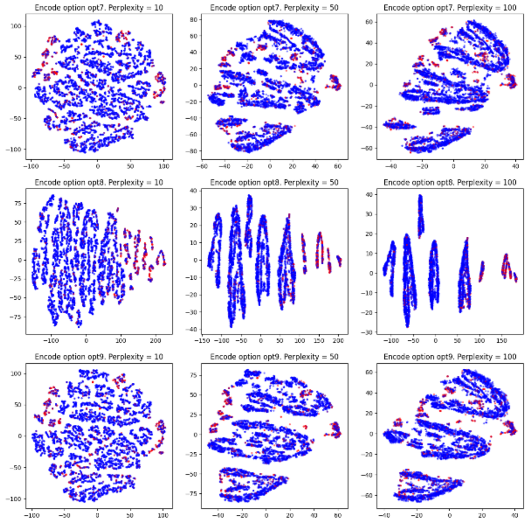
|  |  |
|---|---|
| TN = $ 0.00 | FP = $ -0.50 |
| FN = $ -25.8<br>(lost income) | TP = $ 25.8<br>(26.3 - 0.5) |

# Alternative Approach: Clustering

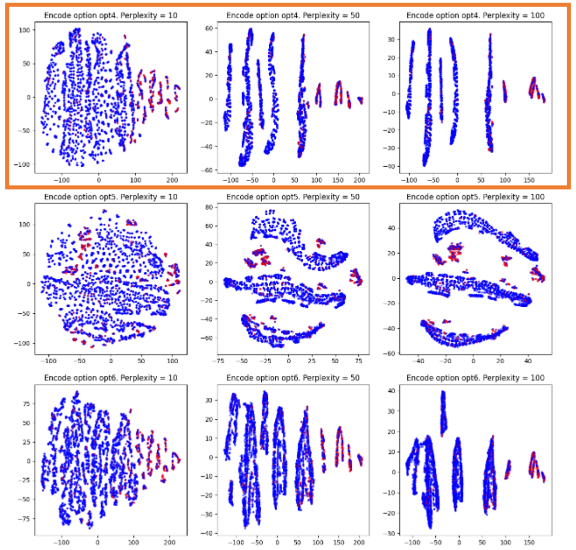
Based on the task description we assumed that clustering technique may help to determine groups of customers with high or low probability of opening deposit

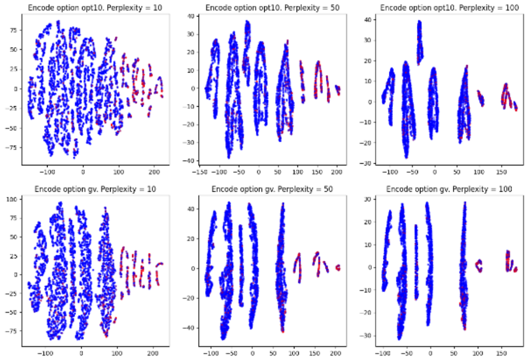
We used t-SNE method to visualize preprocessed datasets. But took only about 8000 random observation from each using stratified sampling



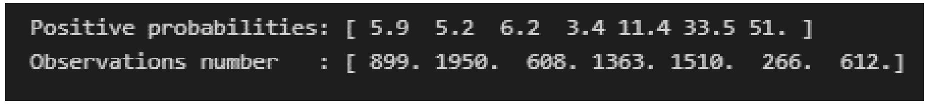


Some preprocessing methods results in **well-shaped clusters**. These clusters have visually different density of positives (red dots).
We chose opt4 preprocessing option to experiment with it

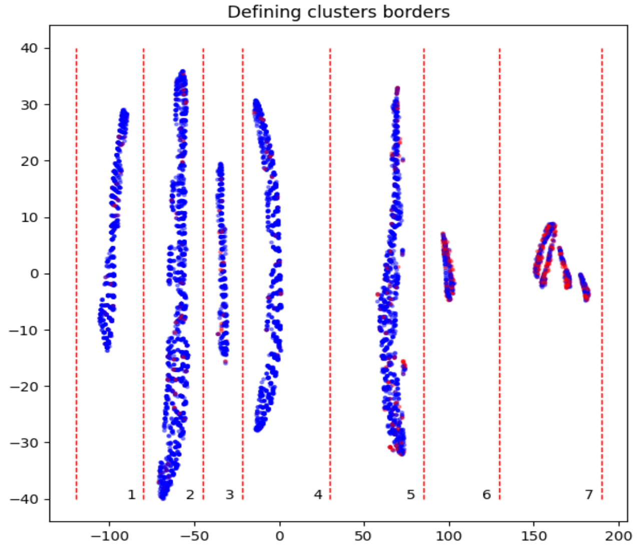




# Alternative Approach: Clustering (continue)



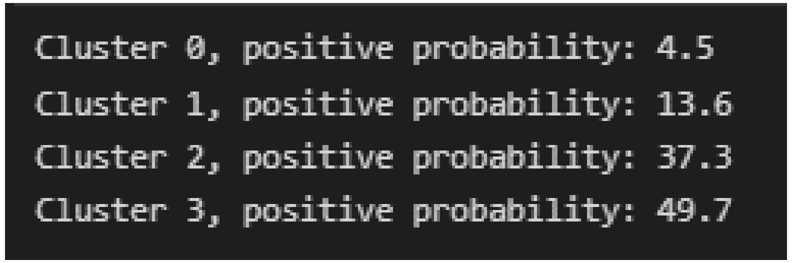
```
Positive probabilities: [ 5.9  5.2  6.2  3.4 11.4 33.5 51. ]
Observations number    : [ 899. 1950.  608. 1363. 1510.  266.  612.]
```



Defining clusters borders

Despite there are 7 clusters on the chart, 4 of them (1 to 4) have near the same low probability of positives (5-6%)

We trained KMeans clustering model from scikit-learn to make 4 clusters on train set. The results show that clustering was performed successfully as we got probabilities we expected



```
Cluster 0, positive probability: 4.5
Cluster 1, positive probability: 13.6
Cluster 2, positive probability: 37.3
Cluster 3, positive probability: 49.7
```

Applying model to test dataset was successful too – probabilities in each of clusters defined by model stay with respect to ones in train dataset

Considering success of clustering technique, we suggest to use it to define groups of customers. Then bank can prioritize calls according to probabilities of positive income
Recall value in this case will be equal to 1.0 as all customers from cluster are supposed to get a call

# Conclusions and Recommendations

**Conclusions:**

- Initial dataset was transformed using 11 different combinations of imputing and encoding
- 5 types of models were trained and evaluated on created datasets
- Logistic Regression model showed high robustness to encoding method and highest recall values on almost all dataset options
- Unsupervised clustering method was alternatively developed and tested on dataset

**Final recommendations:**

1. Use Logistic regression as main predictive model to determine if the customer will open a deposit
2. Try to fine tune model hyperparameters further to improve model efficiency
3. Use simple (ordinal and binary) encoding and imputing based on “most frequent” strategy (preprocessing option 7)
4. Analyze sensitivity of models’ financial results to the level of costs relative to income value

# THANK YOU

Data Glacier

Your Deep Learning Partner