Amount of Children Prediction

Using A Credit Card approval Dataset for Machine Learning

Team Members:
Dvir Iluz
Shulamit Elgrabli
Yitzhak Vider

Dr. Chen Hagag

Time line

Project Goal:

Project Motivation

Methodology

Data Method

Data Preprocessing:

Classification

Clustering:

Project Goal:

Using machine learning to predict the number of .children a bank customer

Project Motivation

Helping the bank customize their marketing and promotions for credit cards that fit the needs of specific groups of customers.

It can also help the bank predict which customers may have trouble paying their credit card bills, so the bank can set credit limits appropriately.

In addition, we will try to look for groups with similar characteristics, to match them with relevant business proposals

Methodology

The data was taken form Kaggle website:

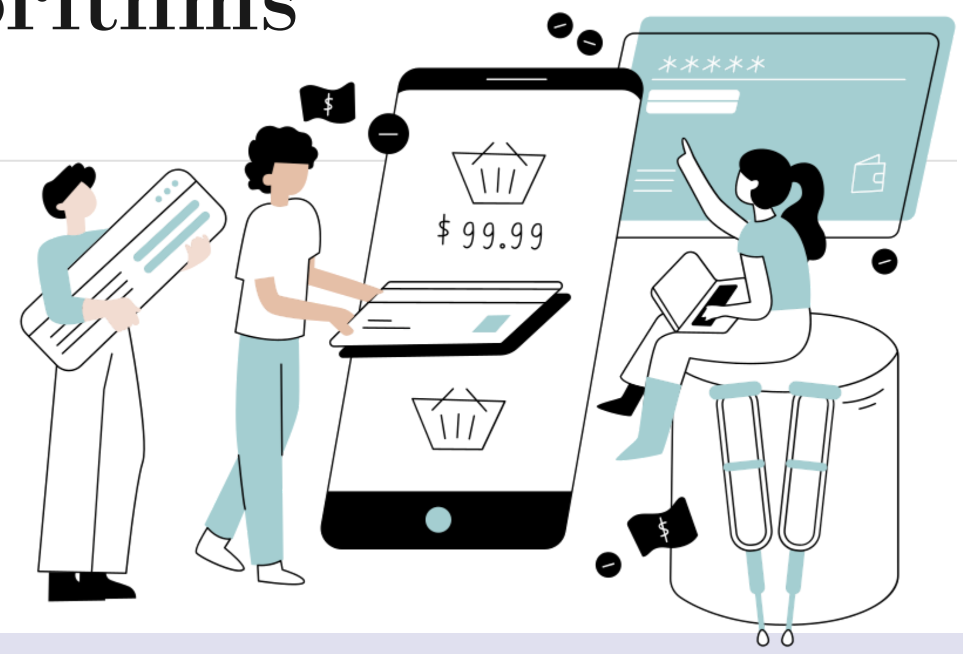
https://www.kaggle.com/datasets/rikdifos/credit-card-approval-

.(prediction and has 438557 rows (examples

Method

The project is split into two parts.
The first part focuses on classification models to determine the number of children in a family
We used the following Machine Learning classification Algorithms

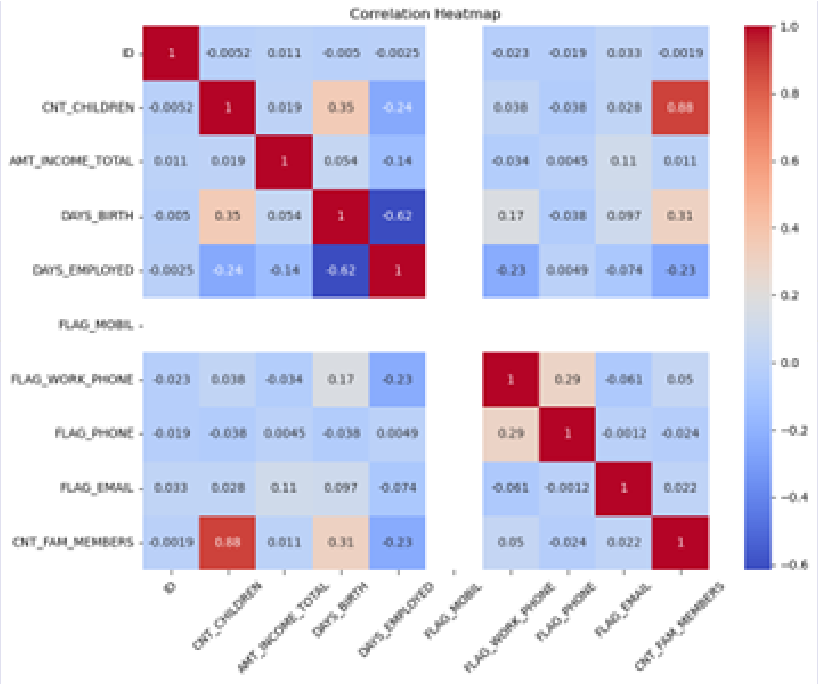
- KNN
- Logistic Regression
- Random Forest
- Gradient Boosting



The second part uses the K-MEAN model to classify people into different groups and understand their unique characteristics. This analysis helps the bank make informed decisions.

Data Preprocessing:

- Check the correlation between two columns that describe the family size
- Converted categorical features into binary values.
- Remove all rows that had Null values in important features
- Divided the data into groups based on the number of children in a family
- Removed irrelevant columns



```
application_rec = pd.read_csv('application_record.csv')
application_rec.loc[(application_rec['CNT_CHILDREN']== 0)  , 'y'] = '0'
application_rec.loc[(application_rec['CNT_CHILDREN']> 1) & (application_rec['CNT_CHILDREN']<= 3) , 'y'] = '1-3'
application_rec.loc[(application_rec['CNT_CHILDREN']> 3) & (application_rec['CNT_CHILDREN']<= 6) , 'y'] = '4-6'
application_rec.loc[(application_rec['CNT_CHILDREN']> 6) & (application_rec['CNT_CHILDREN']<= 10), 'y'] = '7-10'
application_rec.loc[(application_rec['CNT_CHILDREN'] > 10) , 'y'] = 'up10'
application_rec
application_rec.loc[(application_rec['y']== 'up10')]
```

```
application_rec = pd.read_csv('application_record.csv')
application_rec.loc[(application_rec['CNT_CHILDREN']== 0)  , 'y'] = '0'
application_rec.loc[(application_rec['CNT_CHILDREN']== 1)  , 'y'] = '1'
application_rec.loc[(application_rec['CNT_CHILDREN']== 2)  , 'y'] = '2'
application_rec.loc[(application_rec['CNT_CHILDREN']== 3)  , 'y'] = '3'
application_rec.loc[(application_rec['CNT_CHILDREN'] >= 4) , 'y'] = '4up'
application_rec
```

classification

1. Used the Stratified K-Fold cross-validation technique with 10 folds to evaluate model performance.
2. Used one-hot encoder on categorical features and standardized numerical features.
3. We used the following Machine Learning classification Algorithms for the predation:
    - KNN
    - Logistic Regression
    - Random Forest
    - Gradient Boosting

```
summeries['0']
```

	Mean TP	sd TP	Mean FP	sd FP	Mean TN	sd TN	Mean FN	sd FN	Accuracy	Precision	Recall	F1_score	AUC
KNN	16781.6	216.712805	2791.4	100.479053	1045.4	100.400398	2503.7	216.820686	0.770994	0.857385	0.870176	0.863733	0.571321
Random forest	18783.7	59.299325	3562.6	23.139576	274.2	23.241342	501.6	59.267529	0.824229	0.840573	0.973991	0.902377	0.522728
Logistic regression	19285.3	0.458258	3836.8	0.400000	0.0	0.000000	0.0	0.000000	0.834064	0.834064	1.000000	0.909525	0.500000
Gradient Boosting	19267.2	11.461239	3824.7	7.785242	12.1	7.942921	18.1	11.344161	0.833804	0.834370	0.999061	0.909319	0.501108

```
summeries['1-3']
```

	Mean TP	sd TP	Mean FP	sd FP	Mean TN	sd TN	Mean FN	sd FN	Accuracy	Precision	Recall	F1_score	AUC
KNN	1012.9	95.671783	2483.5	197.999621	16856.3	197.863109	2769.4	95.958533	0.772819	0.289698	0.267800	0.278319	0.569693
Random forest	269.1	22.237131	498.2	59.121570	18841.6	59.218578	3513.2	22.306053	0.826512	0.350710	0.071147	0.118296	0.522693
Logistic regression	0.0	0.000000	0.0	0.000000	19339.8	0.600000	3782.3	0.458258	0.836421	NaN	0.000000	NaN	0.500000
Gradient Boosting	12.1	7.942921	18.1	11.344161	19321.7	11.358257	3770.2	7.997500	0.836161	0.400662	0.003199	0.006348	0.501132

4. Calculated accuracy, recall, precision and F1-score for each model using the test dataset and saved them separately.

Classification Conclusions:

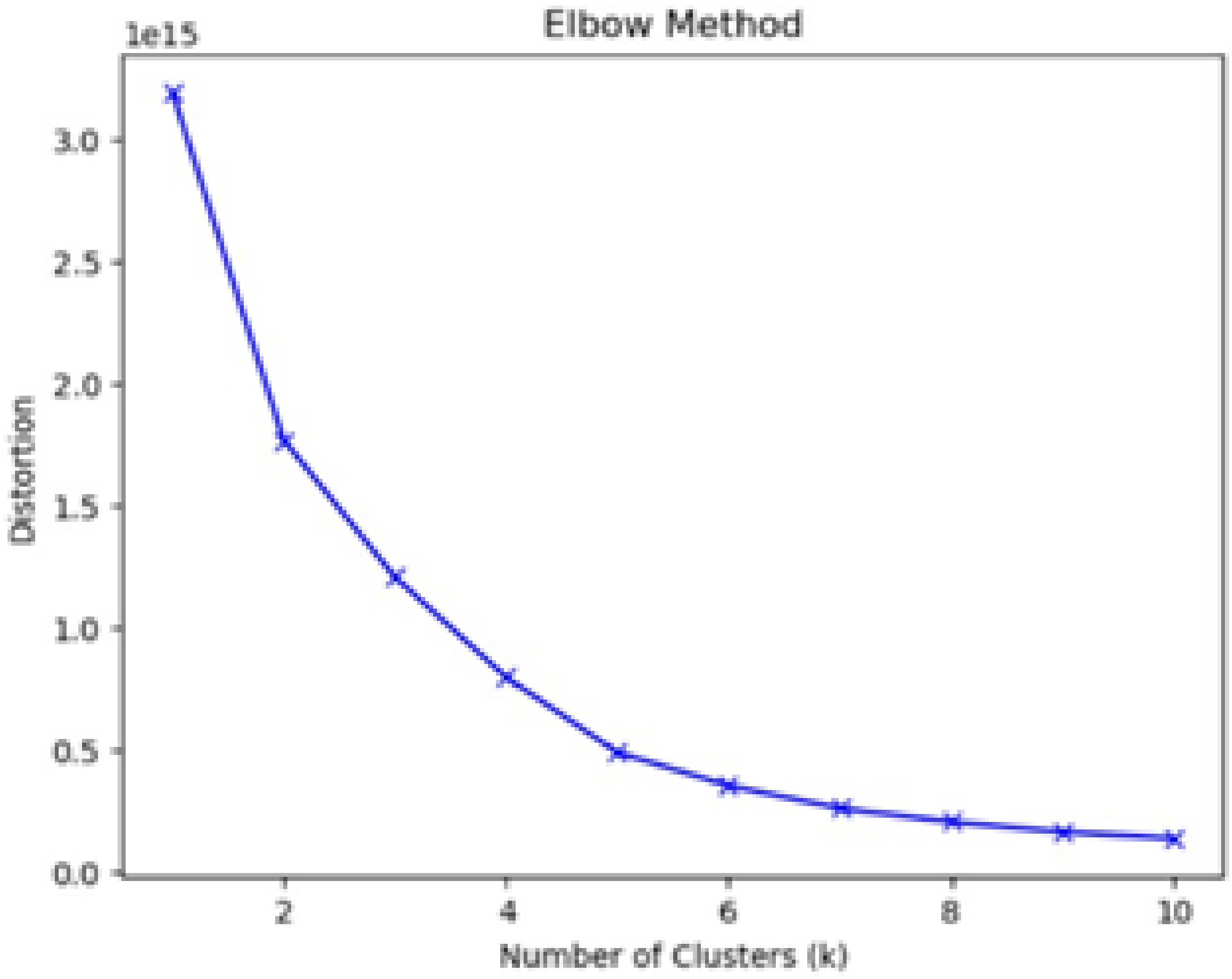
Table 1 demonstrated better performances compared to Tables 2, 3, 4, and 5, with higher precision, recall, and F1-score values. The models in Table 1 had moderate accuracy and showed relatively stronger capabilities in identifying positive instances. However, all tables displayed room for improvement, as the models struggled with low recall and inconsistent AUC values. Further optimization is needed to enhance the models' accuracy and overall predictive power. To improve the model's performance, we aimed to balance the data division based on the variable Y, representing the number of children in the family. Unfortunately, no improvement was found in the indices, and it even seems that there was a deterioration in performance.

```
summeries['0']
```

	Mean TP	sd TP	Mean FP	sd FP	Mean TN	sd TN	Mean FN	sd FN	Accuracy	Precision	Recall	F1_score	AUC
KNN	16781.6	216.712805	2791.4	100.479053	1045.4	100.400398	2503.7	216.820686	0.770994	0.857385	0.870176	0.863733	0.571321
Random forest	18783.7	59.299325	3562.6	23.139576	274.2	23.241342	501.6	59.267529	0.824229	0.840573	0.973991	0.902377	0.522728
Logistic regression	19285.3	0.458258	3836.8	0.400000	0.0	0.000000	0.0	0.000000	0.834064	0.834064	1.000000	0.909525	0.500000
Gradient Boosting	19267.2	11.461239	3824.7	7.785242	12.1	7.942921	18.1	11.344161	0.833804	0.834370	0.999061	0.909319	0.501108

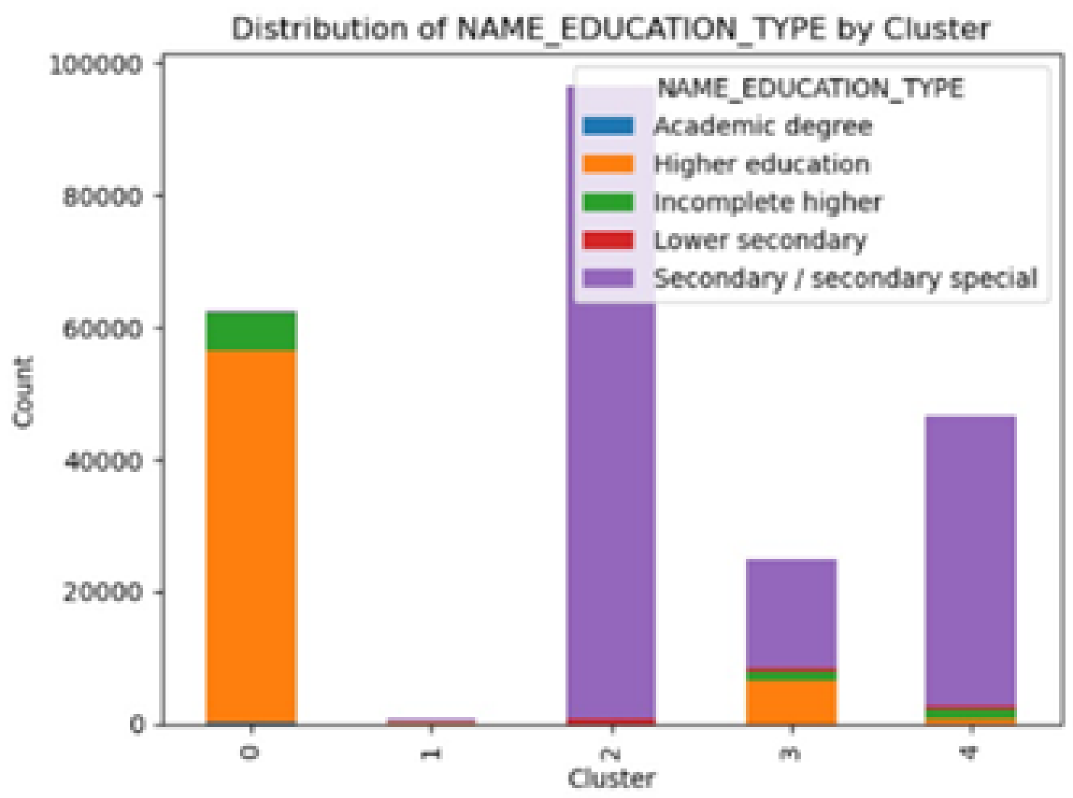
Clustering:

Visualize the elbow curve, which helps determine the optimal number of clusters for k-means clustering. The point indicates the number of clusters that capture the most significant amount of information without excessive complexity.

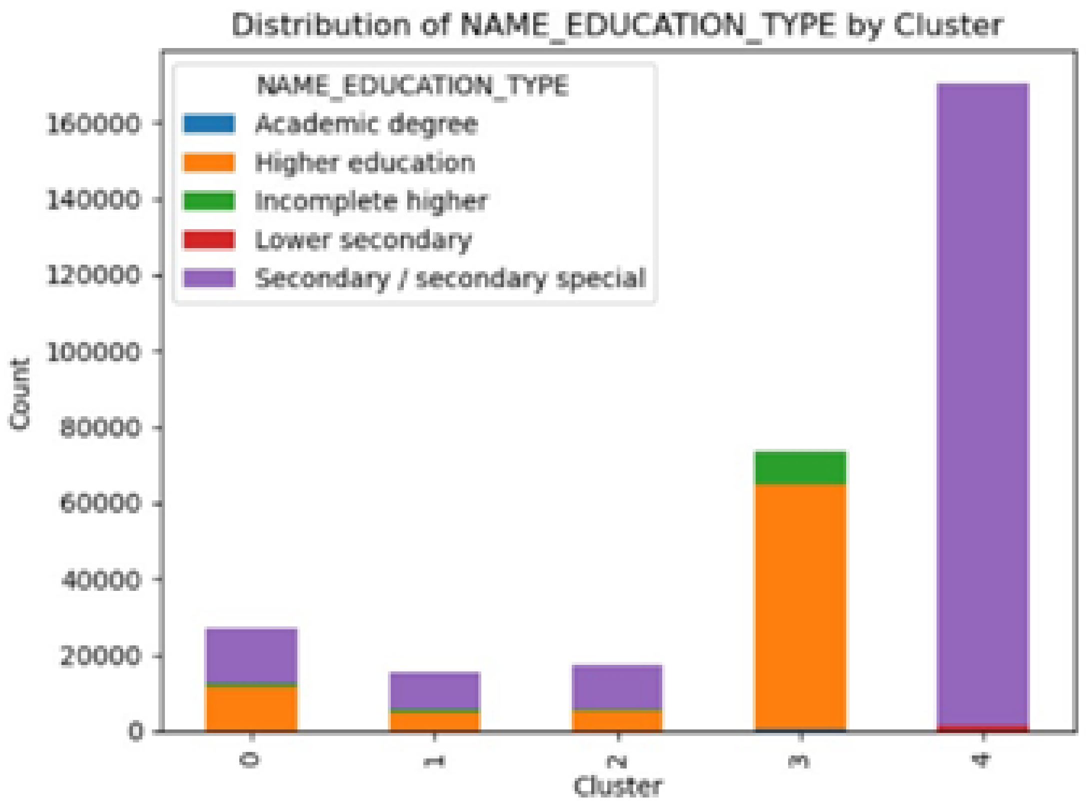


1. We loaded our data without the target column, which is the CNT_CHILDREN column.
2. We preprocess the data by one-hot encoding categorical features, scales the numerical features, and then fits the KMeans model
3. When the amount of children is classified according to the 2 groups we discussed earlier

Clustering Visualization:



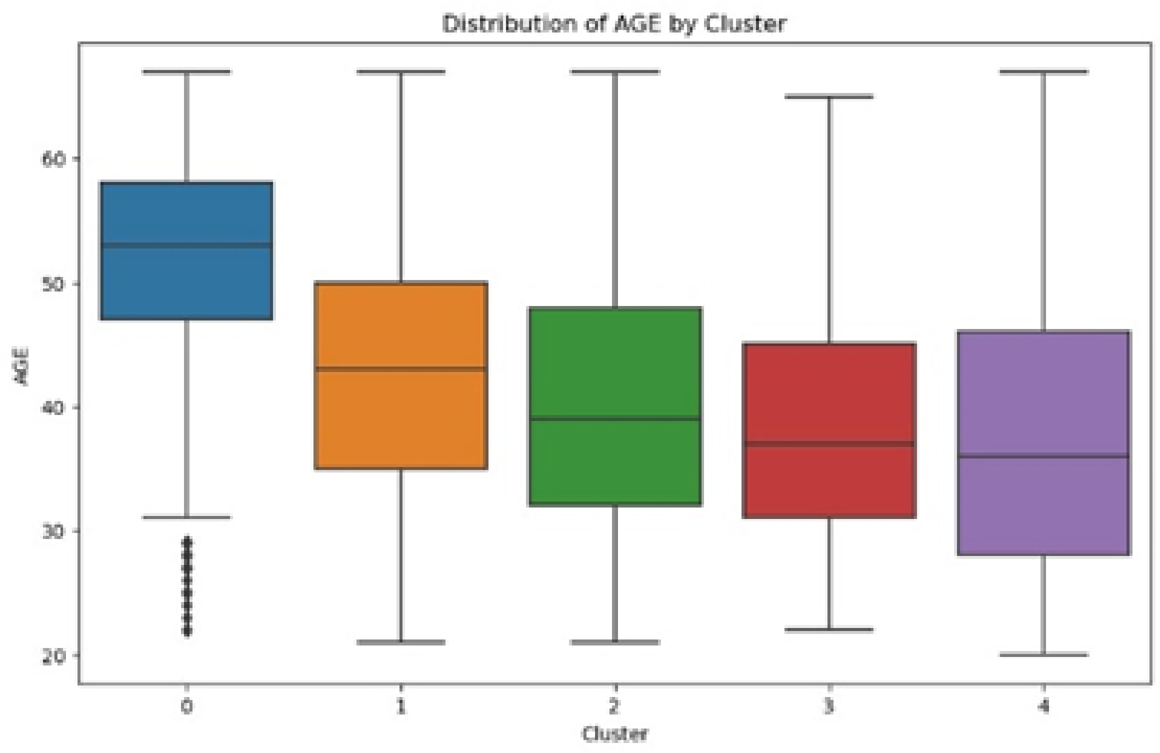
Unequal division y parmter into groups



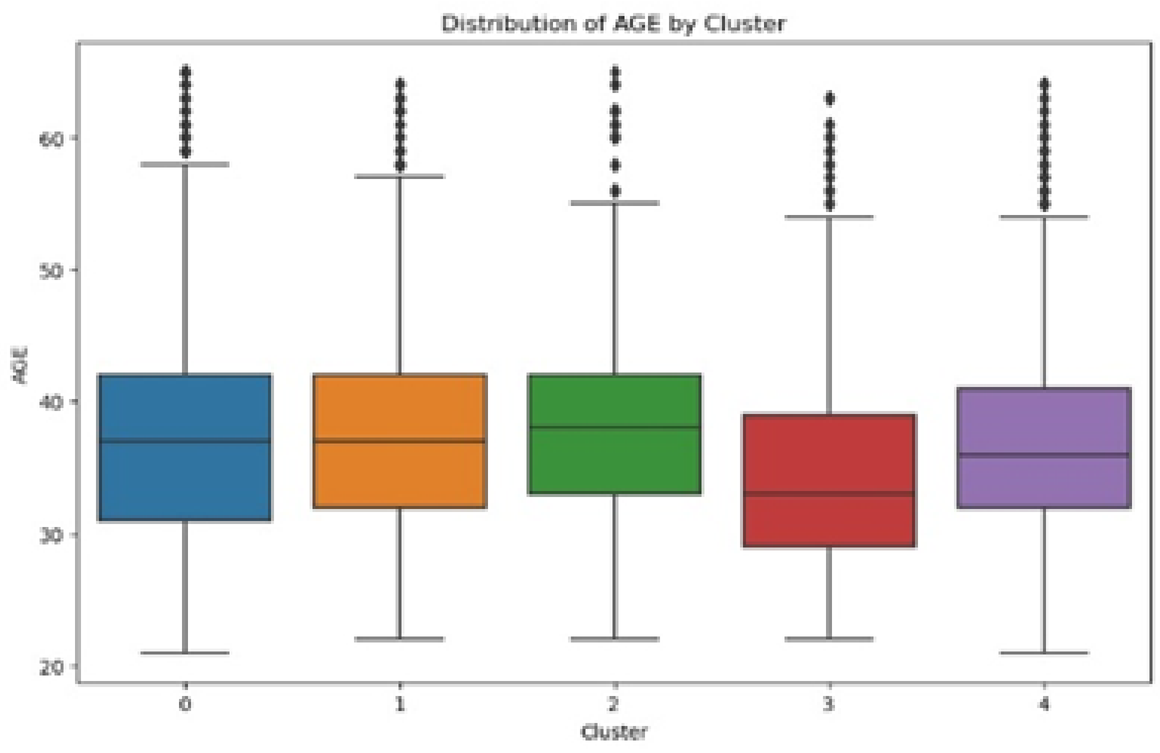
Equal division y pamter into groups

In the graph where the groups are not equal, a person with a higher education is associated with groups 2, 3, 4, while in the graph where the division into groups is equal, a person with a higher education is associated with the 4th group.

Clustering Visualization:



Unequal division y parmter into groups

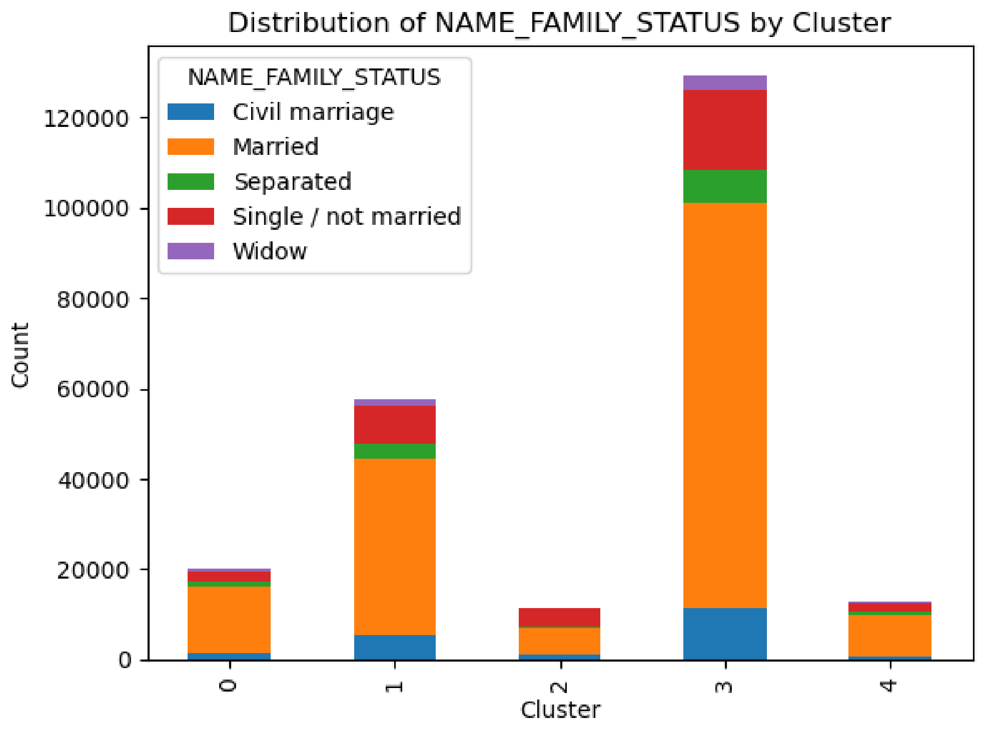


Equal division y pamter into groups

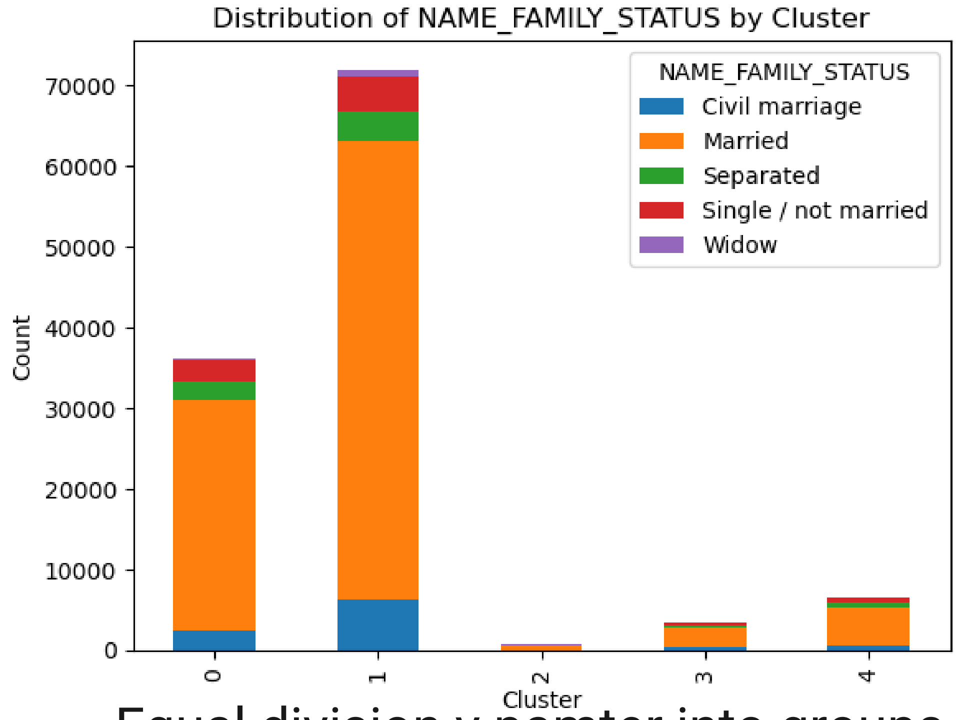
We can clearly see when the Y is divided in the form of categories that there is a difference between the ages. People in group 0 are more likely to be adults, around their 50’s.

On the other hand, in the second division there is no real difference between the ages in each group.

Clustering Visualization:



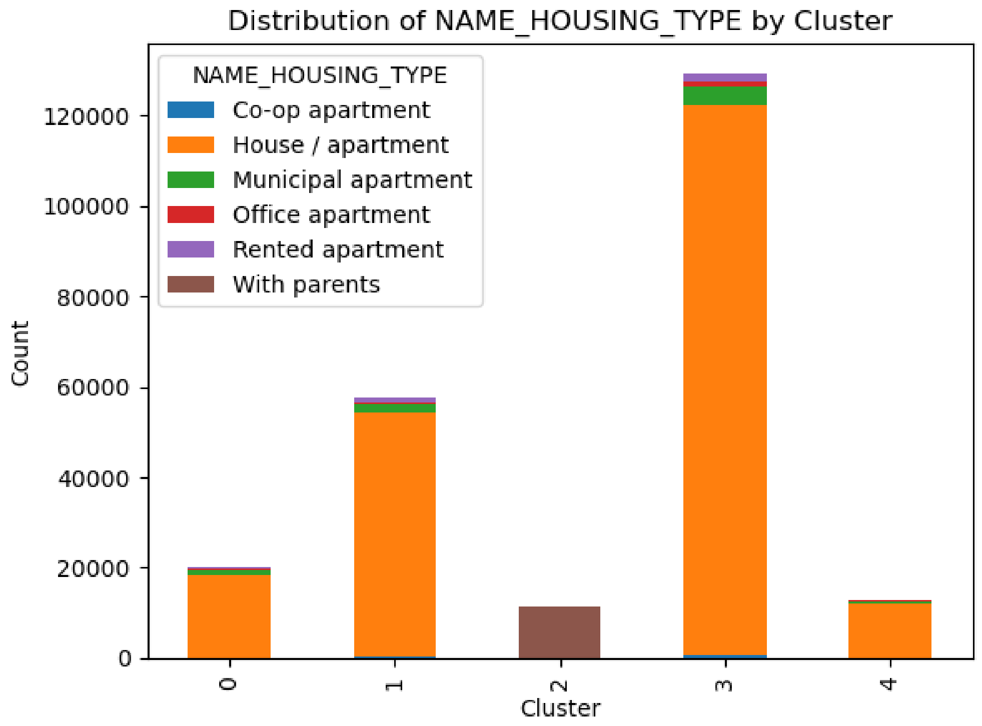
Unequal division y parmter into groups



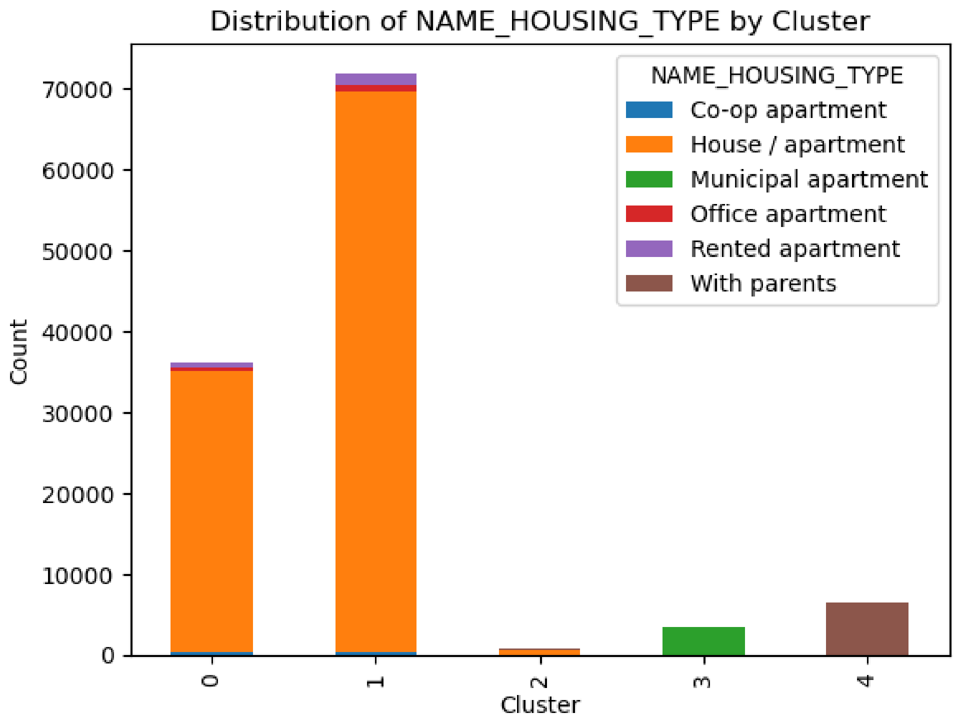
Equal division y pamter into groups

It does not appear that the groups are divided according to the marital status of the people

Clustering Visualization:



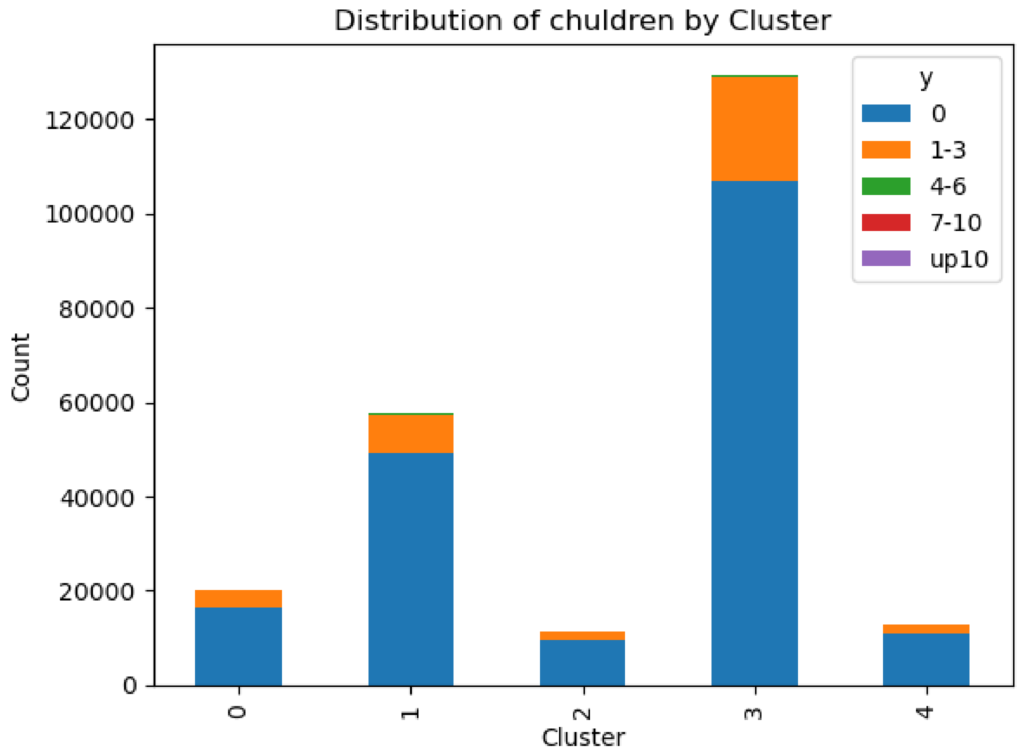
Unequal division y parmter into groups



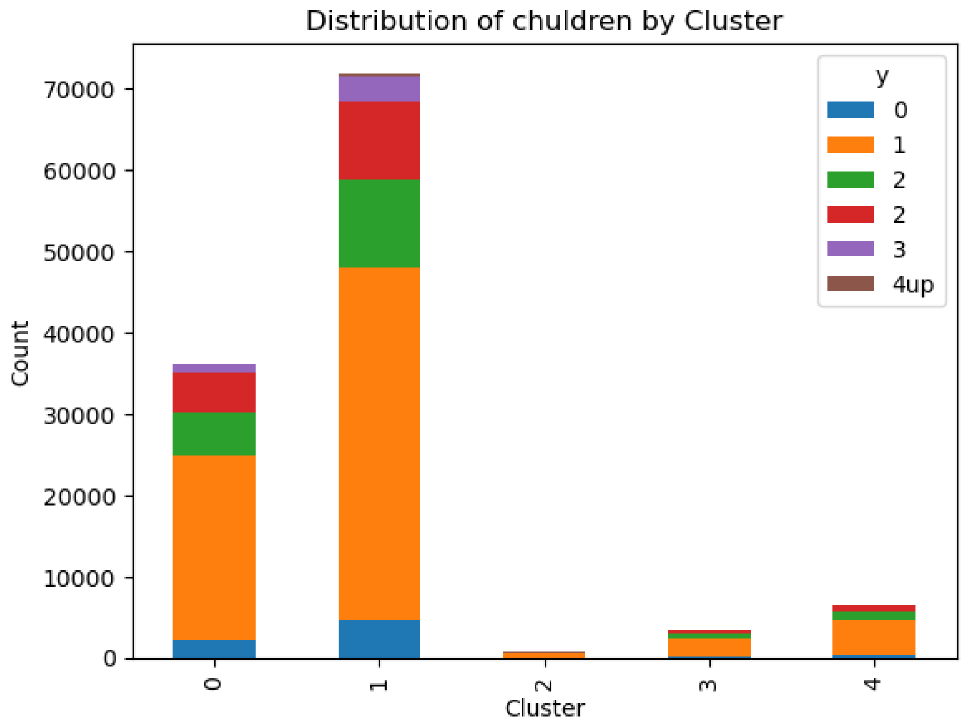
Equal division y pamter into groups

The model classifies well the group of people who live with the parents in the 2 cases
In the case where the division into the number of children is more balanced, the model also classifies people who live in munificial housing in a good way

Clustering Visualization:



Unequal division y parmter into groups



Equal division y pamter into groups

It seems that there is no connection between the number of children and the classification into groups

.Clustering Conclusions:

.There are no clear answers in the clustering model

We could not find interesting or relevant results in our models, even after trying

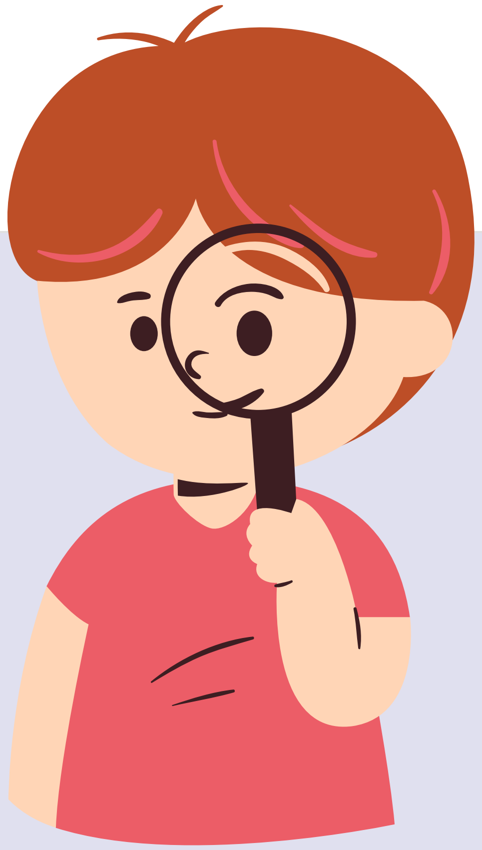
.in several ways of divisions

Distribution by group of number of children

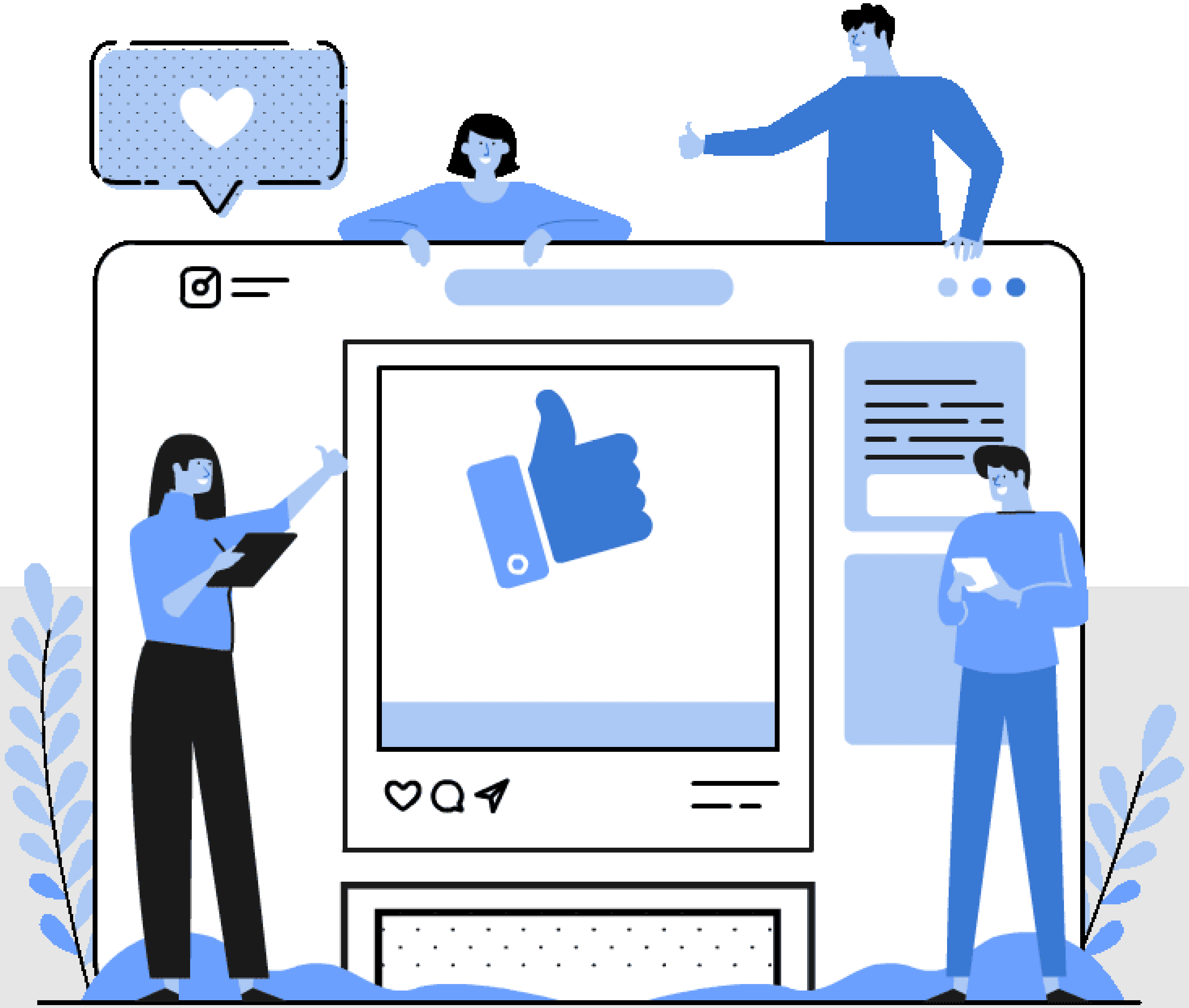
A. 0, 1-3,4-6,7-10, 10 and above

B. 0,1,2,3,4 and above

C. 0,1,2 and above



We didn't talk about it because there were no good results



Thank you for your time