Clustering project

Abstract— The main task of the analysis is segmentation of the customers of the bank given a dataset containing information about various credit card operations of the customers. As we had no correct labels for the dataset we decided to take unsupervised learning approach, particularly to run K-Means algorithm on our dataset to be able to segment the customers. Before creating the model we did data preprocessing` replaced missing values, removed columns that didn't have numerical values and removed the variables which we estimated to be unnecessary for our model looking at the variation of them and the correlation with different variables. After it we performed Principal Component Analysis to do dimensionality reduction, which would help to visualize the data and give K-Means a simple data so that it would be able to perform better and cluster the data points in an accurate way. After all of the steps, which are explained in details in the following paper, we were able to reach our goal and assign each customer to the particular group.

1 Problem Statement

Customer segmentation is very important in every business, as it gives deep knowledge about the structure of the market and defines hints on how the policy of the company can change to target more customers and gain higher profits. Banks are not exception and the management of a bank also needs to segment customers in an efficient way.

The main purpose of our study will be to do customer segmentation for the bank, having in our hands information about the behavior of almost 9000 credit card holders of the bank. There is one question to answer` to which group each customer belongs. It can help to define the groups of the customers that don't bring high profit to the bank and try to solve the issues that lead to it. Also we can note what are the characteristics of the groups that bring high profit, which again can be helpful in marketing decisions.
We will use "Credit Card Dataset" from Kaggle [1], which has 8950 rows and 18 different features. We will apply unsupervised learning, particularly K-Means clustering, to our dataset and 18 features should be more than enough to create a good model which will give reliable results. We will even need to pick the best features or do dimensionality reduction to create a good model.

2 State of the Art

Unsupervised learning and particularly K-Means clustering is very common nowadays, because if we have unlabeled data, it can be very helpful on creating the necessary categories and doing the segmentation of the data points. Also K-Means is pretty easy to understand and doesn't require sophisticated techniques to build a good model. Its easiness also leads to some drawbacks, as small changes in the data can lead to high variance in the outcome.

We have selected 3 articles from the internet which discuss problems common with the one we are going to discuss.

The first article [2] gives steps on how to apply K-Means clustering in Python. It generates a random data and tries to create clusters on the generated data points. The technology used in the article is Python, which will also be used in our study. Likewise the article we are going to use Pandas, Numpy and Scikit-learn libraries for the analysis, and Matplotlib library for creating visualizations.

The second [3] and the third [4] articles are even closer to the study we are going to take, a they also have different datasets with more than 2 features. The purpose of the articles is again to do clustering of the data and create appropriate groups of the data points. They apply dimensionality reduction with help of Principal Component Analysis, and that is exactly what we are also going to do. There are 2 main reasons for dimensionality reduction, the first one is that K-Means can make more accurate clustering if the number of features is not high (a drawback here is that if we don't do the reduction in the right way, we might lose important information, and even good clustering on the reduced data may not provide reliable information for the general data). The second reason is that with many features we can't visualize data and show the clustered data points, but with the reduced data we will be available to perform that task.

We will use the same approaches mentioned in the article + some additional steps. It won't cause any problems, because for the K-Means algorithm the content of the data doesn't matter, it will do its work if it is given well-processed data and the steps mentioned in the articles will just help us to do good preprocessing and create the best possible model.

3 Properties of the Data

The data was directly downloaded from Kaggle [1]. It has information about 8950 credit card holders and has 18 different features.

All the columns of the dataset contain numerical values, except the first one, which is the column representing information about the customer's ID. The dataset contains information about the balance on the credit cards, and different purchases and payments with their frequency.

One problem with the dataset was that it was containing missing values in two columns` "CREDIT_LIMIT" and "MINIMUM_PAYMENTS, so later we will need to use common technique in machine learning and replace all the missing values with the mean value of the particular column.

Another problem in the dataset might be the difference between the values of different columns, for example the values in "PAYMENTS" columns are pretty big numbers, while the values for example in "BALANCE_FREQUENCY" are floating point numbers in the range of 0 and 1. To solve that problem we will normalize the dataset when doing the analysis, so that all the values will be in range of 0 and 1.
After doing normalization we can also check the variances of distinct columns and remove the columns that have low variance, because they won't give any help when determining the clusters later.
We don't have problems with outliers in the dataset, as our goal is to cluster all the customers and we can't remove any data points.

The last step of the analysis will be to investigate the correlation between the pairs of the variables. For this we will create the correlation matrix of the variables`

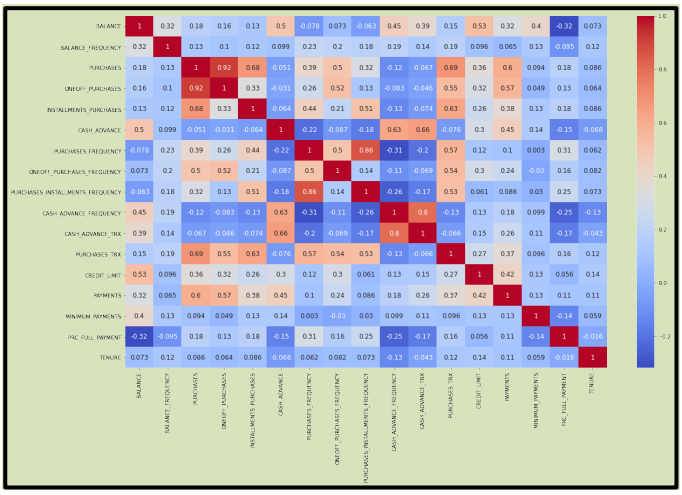

Fig. 1. Correlation matrix of the variables (in this matrix all the variables are included, but later the number of variables might be less, if we remove any of the columns in the steps mentioned above, all the graphs and tables will be provided in the given Jupyter Notebook)

In the analysis process we will define a threshold, for example 0.5, and if the absolute value of the correlation between two variables is higher than that threshold, we can remove one of those variables from the dataset, as the other one will already provide the necessary information and there is no use in keeping both. From the matrix provided we can see that there are many pairs that have high correlation, so if in the final matrix there are still such variables, we will do the necessary removals.

Of course all the steps we perform might not be enough to remove all the necessary variables, so we will also perform Principal Component Analysis (PCA) to be able to plot the data points and create a good clustering model.

4 Analysis

4.1 Approach

Now we will define the steps that will be carried out in the analysis process`

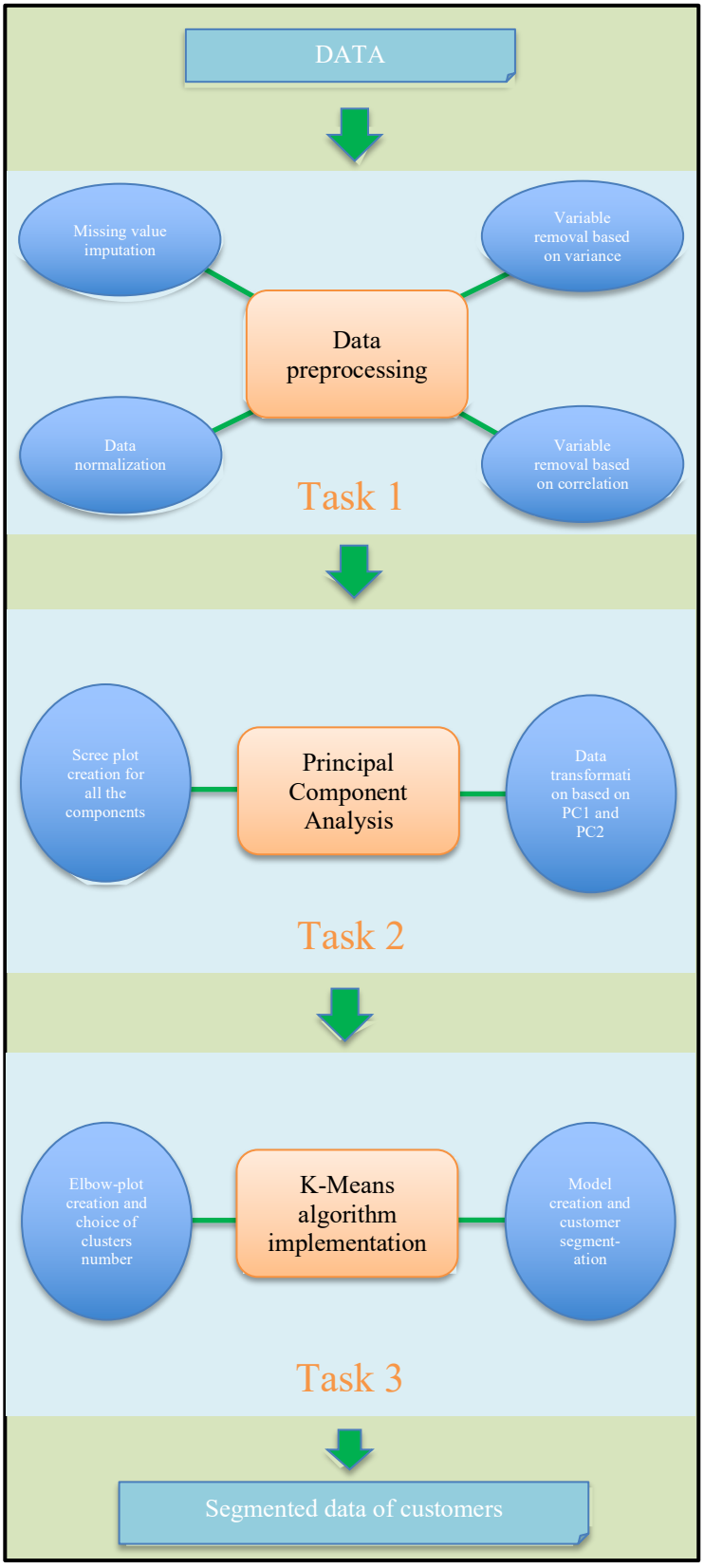

Fig. 2. Approach to the analysis

In the first task we will perform the steps described in the previous section. First we will substitute the missing values with the means of the particular columns, then we will normalize the data and filter the variables based on their variance and correlation with other variables. For doing the last two steps we will have to manually define thresholds (as it is absolutely our choice what number to define, computer can't count it), so that variables out of the defined thresholds will be removed from the dataset. We will gather the necessary numbers from the computer, look at the given values and remove the variables we don't need.

In the second task we will apply Principal Component Analysis to our dataset to do dimensionality reduction. Surely the ideal situation will be if we are able to keep only two components that together will explain the majority of the variation in the dataset. We will need to check this by creating a scree plot, which will show what part of variation each principal component explains. If we succeed with leaving only two components, we can transform the dataset based on those two components and move on to creating the clustering model.

The last task of the analysis will be to create a K-Means clustering model which will group the customers. Here we will need to manually select the number of clusters, as we don't have any predefined information on how many clusters we should have. We will create test models with different numbers of clusters and will generate an elbow-plot based on the distance estimations we get. After selecting the best number of clusters, we will create our final model and assign a cluster number to each of the customers.

By final step we will create a new dataframe, which will contain only two columns` the ID of the customer and the cluster number the customer belongs to, thus we will have a fully segmented list of the customers.

4.2 Process

The first step in the analysis was to fill in the missing values of the dataset (all the manipulations with the data are done with Python's Pandas library). We substituted missing values with the mean of the column the value belongs to, so now there are now empty cells in the dataset.

Next we created a copy of the dataset to keep the original one, and removed the "CUST_ID" column from our dataset, which is the only column that contains text and not numbers.

After it we normalized our dataset, so each value is now in the range from 0 to 1.

After it we counted the variances for each columns and defined a threshold of variances 0,01. So the columns that have variance less then 0,01, meaning the values generally differ by less than 1%, will be removed from the dataset, as having not much varied values they won't be able to help in defining the clusters. We removed the particular columns from the dataset` "PURCHASES", "ONEOFF_PURCHASES", "INSTALLMENTS_PURCHASES", "CASH_ADVANCE", "CASH_ADVANCE_TRX", "PURCHASES_TRX", "PAYMENTS", "MINIMUM_PAYMENTS". So after this step we have only 9 columns left in the dataset, meaning we have kept only 9 variables.

The final step of the data preprocessing was plotting the correlation matrix of the remaining variables`

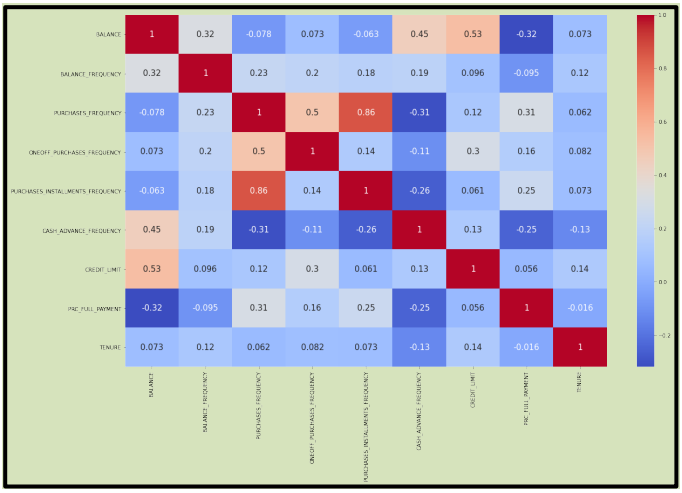

Fig. 3. Correlation matrix of the remaining variables

Here too we define the threshold as 0.5 and will remove one variable from the pairs that have higher correlation greater than that. We do this because if the two variables are highly correlated there is no use in keeping both because they will define the same behavior. We will remove "CREDIT_LIMIT" for the pair "BALANCE" - "CREDIT_LIMIT", and "PURCHASES_INSTALLMENTS_FREQUENCY" from the pair "PURCHASES_FREQUENCY" - "PURCHASES_INSTALLMENTS_FREQUENCY". So now there are 7 columns left in the dataset.

Now we are ready to start applying PCA to our dataset, which will help to reduce the dimension even more. First we will do the transformation with the number of principal components equal to the number of variables to check whether PC1 and PC2 will explain enough portion of the variance of the data points and can be reliable for creating the clusters later. We got those results`

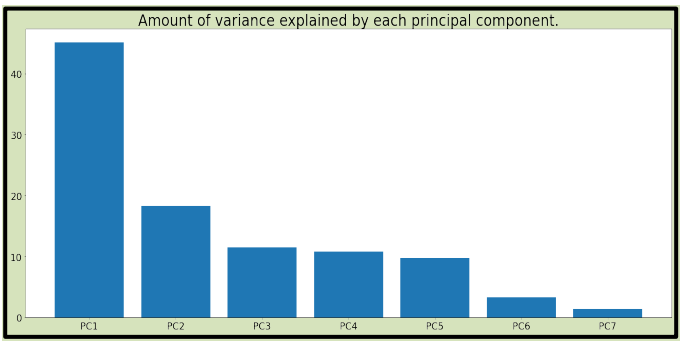

Fig. 4. Scree plot of principal components

From the scree plot we can detect that PC1 and PC2 together explain more than 60% of the variance, so we can easily use them to reduce the dimension of the data, plot it and create the K-Means algorithm. So we will apply PCA again, but will specify that we want to use only two principal components and will transform the data based only on that two.

Now we are finally ready to create the model. Here we have a manual step to conduct. To create the K-Means model we will need to specify the number of the clusters, but as we don't have any pre-given information about the dataset, we will have to do additional analysis to determine what will be the best number of the clusters. For this, we will create different models with the number of cluster ranging from 1 to 10, will count the values of the squared distances and will draw a plot of it to see what sum of distances we have for each number of cluster`

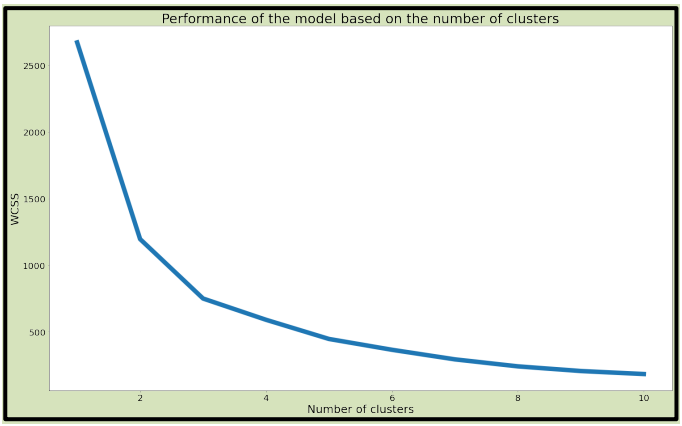

Fig. 5. Plot of the sum of squared distances for different number of clusters

From the elbow-graph created above we can see that the values of distances decrease with big steps when we add the number of clusters until we get to 5, after it the sum of distances decreases in a slow manner, which means that 5 is the optimal number of clusters for our model.

Now when we have defined the optimal number of clusters, we can move on to the final model creation, which will be a K-Means algorithm that will group all the rows in our dataset into 5 clusters. But before doing it, we will run a simple model with only one cluster to plot the data points initially, when they have not been divided to groups yet`

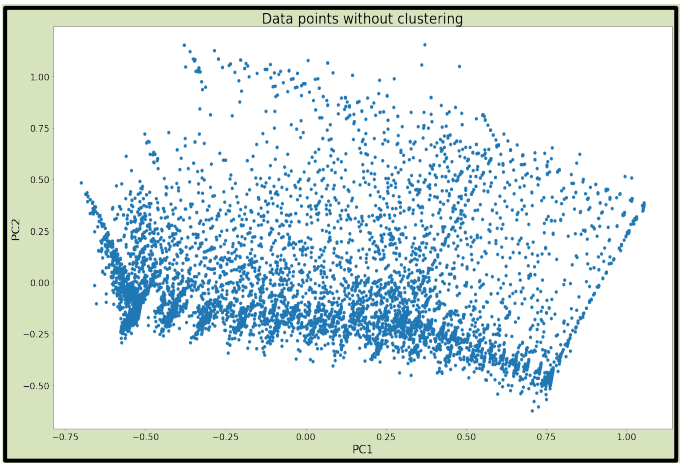

Fig. 6 The distribution of the data points before applying clustering algorithm

From the graph we can see how two principal components helped us to visualize the data points and when we apply the algorithm in the next step, we will clearly be able to see how the groups are formed. Now we will create the same scatter plot, but will set the number of clusters to 5 and will also plot the centroids for each cluster, to visually emphasize how the data points are grouped around the particular centroid`

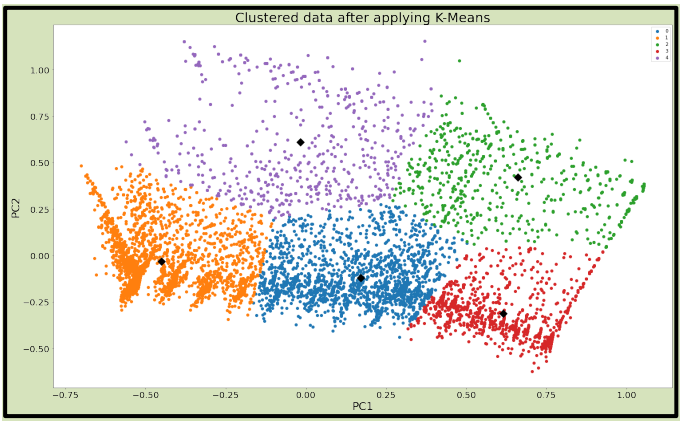

Fig. 7 The distribution of the data points after applying clustering algorithm

Now from this graph we can clearly see how the data points that represent customers are grouped into 5 clusters. We can also detect centroids` the rhombuses in black.
For finishing the task we will create a new dataset, that will contain only two columns` the ID of the particular customer and the number of the cluster they belong to, so we will have a fully segmented list of all the customers available in the dataset.

4.3 Results

To sum up, we were able to achieve the goal we initially set` we assigned each customer to one of the 5 groups.

The initial dataset was very rich in features and it would be very hard to classify people looking at the different numbers from that dataset, as we would need to find different criterions and start grouping based on them.

The approach we took was much more efficient, because first we removed the unnecessary variables that wouldn't give any help to us and run a clustering algorithm on the remained variables to define the groups of people.

Here is the showcase of the work we have done, some rows of the final dataset we created, where for every customer we have a cluster number to which that customer belongs to`

	Customer_ID	Cluster
0	C10001	1
1	C10002	1
2	C10003	3
3	C10004	1
4	C10005	1
...	...	...
8945	C19186	2
8946	C19187	0
8947	C19188	4
8948	C19189	1
8949	C19190	0

Fig. 8 A portion of the final dataset

Now when the fully segmented list of customers is available, the bank can produce any analysis that requires segmentation of customers, so we can clearly say that the model can have practical use for every bank.

5 Critical reflection

Nothing is perfect in the world of Machine Learning and our case is not an exception. The biggest problem we have is that we used unsupervised learning for classifying the customers and we don't have any initial dataset with correct labels to which we can compare our results to. Surely, this was the main task, to create that dataset with label and it is clear that any model we would create, even the most complicated and fancy one, there would still be a risk of making mistakes as we don't have anything to compare our final results to.

Besides that problem mentioned above, everything else has been done in a very detailed and accurate way, and we can almost surely state that all the steps performed in the analyzing process are correct one, because in the analysis we have substantiated every step we take. There is certainly a chance that by defining different thresholds for variation or correlation, or for picking different number of clusters, we would be able to get slightly better results, but it wouldn't make much change, and might even harm, because if we didn't simplify our model enough, K-Means wouldn't be able to categorize everything in a good way and we wouldn't be able to visually prove the effectiveness of the results and show the groups that have been formed for the customers. Also if we kept big amount of the variables, if the algorithm was to be run on a bigger dataset it would take too much time to get the necessary results.

Manually decision making was one of the key points in the whole analysis process and we have taken decisions ourselves a couple of times. Our decisions are surely reliable as every time we have created the necessary visualizations or have gathered necessary information from the dataset to prove every choice we make.
So we can state the analysis process reached the goal and we were able to create a high quality model.

Table of word counts

Problem statement	217/250
State of the art	389/500
Properties of the data	391/500
Analysis: Approach	344/500
Analysis: Process	800/1500
Analysis: Results	168/200
Critical reflection	330/500

References

[1] https://www.kaggle.com/arjunbhasin2013/ccdata?select=CC+GENERAL.csv
Credit Card dataset for clustering

[2] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
Understanding K-means Clustering in Machine Learning, Dr. Michael J. Garbade - Sep 13, 2018

[3] https://medium.com/@dmitriy.kavyazin/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2
Principal Component Analysis and k-means Clustering to Visualize a High Dimensional Dataset, Dmitriy - Feb 21, 2019

[4] https://365datascience.com/tutorials/python-tutorials/pca-k-means/
How to Combine PCA and K-means Clustering in Python?, 365DataScience