Customer Segmentation based on Behavioural Data in E-marketplace

Andrew Aziz
Abstract

Customer Segmentation based on Behavioural Data in E-marketplace

Andrew Aziz

In the past years, research in the fields of big data analysis, machine learning and data mining techniques is getting more frequent. This thesis describes a customer segmentation approach in a second hand vintage clothing E-marketplace Plick. These customer groups are based on user interactions with items in the marketplace such as views and "likes". A major goal of this thesis was to construct a personal feed for each user where the items are derived from the user groups. The customer segmentation method discussed in this paper is based on the clustering algorithm K-means using cosine similarity as the similarity measure. The input matrix used by the K-means algorithm is a User-Brand ratings matrix where each brand is given a rating by each user. A visualization tool was also constructed in order to get a better picture of the data and the resulting clusters. In order to visualize the highly dimensional User-Brand matrix, Principal Component Analysis is used as a dimensionality reduction algorithm.
Table of Contents

1 Introduction 1
  1.1 Background .................................. 1
  1.2 Project Description ............................. 1
  1.3 Questions .................................. 2
  1.4 Report Outline ................................ 3

2 Related Work 4
  2.1 Plick Recommender System .......................... 4
  2.2 Customer Segmentation using K-means .................. 4
  2.3 Dimensionality reduction using Principal Components Analysis 5

3 Cluster Analysis 6
  3.1 Customer Segmentation ............................... 6
  3.2 Data ........................................ 6
  3.3 Clustering Algorithms ............................... 7
    3.3.1 Similarity Measures ............................ 7
    3.3.2 Principal Component Analysis .................... 8
    3.3.3 K-means .................................. 11
  3.4 Collaborative Filtering ............................. 12

4 Plick System Information 14
  4.1 System Description ................................ 14
  4.2 Features ..................................... 16
  4.3 Data in Plick .................................. 17

5 Evaluation of Clustering Analysis 18
  5.1 Data Preprocessing ................................ 18
    5.1.1 Principal Component Analysis ................. 19
    5.1.2 Data analysis ................................ 20
  5.2 Cluster Analysis ................................ 22

6 Implementation 30
  6.1 Cluster Analysis ................................ 30
  6.2 Data Visualization ................................ 30
  6.3 Final system .................................. 32

7 Conclusion 33

References 35
1 Introduction

1.1 Background

The aim of this project was to get more insight about the users of the vintage clothing E-marketplace application Plick and also to improve their user experience. The application has seen an increase of popularity, boasting a user-base in the tens of thousands. This large user-base has lead to an increasing stream of uploaded items available for the interested buyers. However, many of these items are not viewed by most buyers due to the amount of items being created continuously. Older items are lost in the seemingly never-ending list of articles. This problem is often referred to as information overload [1] and can be solved by providing specialized article streams for each user depending on his/her preferences. This report will cover an analysis of the Plick user-base, ideally, the analysis should uncover some patterns in the data that can be used to combine users into smaller groups. Two users in the Plick marketplace are considered similar if they have similar preferences in clothes. The gathered data for the analysis and the similarity algorithms will be thoroughly described in this report.

Clustering users into small groups, also called customer segmentation [2] [3], that can be targeted with specialized content is a strategic approach from a business standpoint to improve each user’s experience which in turn aids in customer retention. Furthermore, pleased customers are more likely to recommend the marketplace to other people, leading to an increase in value for the company. This type of strategic approach is part of the company’s business intelligence (BI) which is a set of technical processes, e.g. data mining and analytics, used to identify and create new business opportunities connected to each customer’s needs.

Customer segmentation is regularly used for splitting users into groups based on gender, age, geographic location and spending patterns to name a few. However, in this report a more uncommon customer clustering approach based on user behaviour in the Plick E-marketplace will be evaluated. This process is not based on any pre-existing relations or rules. Rather, the data itself reveals possible similarities between users.

1.2 Project Description

The main goal of this project were to implement and deliver a component that can be used in the Plick system. Given behavioural data of the users from the Plick database, the component should produce a set of user groups
based on user similarity. Using these groups, another desired aspect was to recommend popular items to the users.

This project was practical in its nature but it was based on a comprehensive research study on recommender systems, cluster analysis algorithms and business intelligence. The project was split into three phases:

- Study of the research field to identify common algorithms and methods used to solve this kind of problem.
- Implementation and comparison of the chosen algorithms integrated with the Plick data.
- Evaluation of the implemented system and integration with the Plick system for customer segmentation.

The component accesses and manipulates the Plick PostgreSQL database and the software itself is implemented in Python and the Python libraries Numpy and SciPy. When ready for launch, the component is to be hosted in a separate server in the Amazon Web Services (AWS) Cloud.

1.3 Questions

This kind of problem poses unique challenges that needed to be answered prior to the project. Firstly, the data in an E-marketplace is always changing since items in the system are removed whenever they are sold. Therefore, grouping people using popular items within the group as a similarity measure is not possible. Rather, the way a user behaves in the application needs to be the focus of the clustering algorithm. The main questions that arise from this are:

- How can the Plick data be preprocessed in order to correctly segment users into groups of similar people?
- What is a reasonable cluster size?
- How can the created clusters be used in the application to target users with specialized content e.g. item recommendations?

These questions will be answered thoroughly in this report.
1.4 Report Outline

The report consists of seven sections:

1. Section 1 introduces the project’s research area and the project’s goals.
2. Section 2 presents some research papers that are related to this project.
3. Section 3 gives a thorough background about cluster analysis and customer segmentation. In this section, a set of chosen methods are chosen and presented.
4. Section 4 describes the Plick system more in depth.
5. Section 5 gives an evaluation of the chosen method.
6. In Section 6, the system implementation is described.
7. Finally, in Section 7, the results are discussed and some future work is presented.
2 Related Work

In the field of cluster analysis it is important to choose an algorithm well suited for the available dataset. In order to make this choice, projects that tackle similar problems and use similar techniques were analysed during the first step of the project. In this section, three systems using different algorithms will be presented.

2.1 Plick Recommender System

In 2015, Elvander, a master’s student from Uppsala University created a recommender system for Plick [4] using user-based collaborative filtering [1]. This method is largely used in the recommender system field used in renowned systems such as Youtube [5] and Amazon [6]. Elvander states that due to the unstable nature of the item data in the system, it is more beneficial to use users to create recommendations rather than using items.

This type of recommender system is similar to the cluster analysis problem that this project is centred around. Recommending items to a user with user-based collaborative filtering is a two step process where each user is first given a neighbourhood of similar users. Secondly, the data in the computed neighbourhood is used to find a set of popular items within the neighbourhood based on the behaviour of the users in the neighbourhood.

In order to compare the performance of the clustering made by a collaborative filtering algorithm with other cluster analysis specific solutions, the first part of the user-based collaborative filtering algorithm provided by Elvander can be used.

2.2 Customer Segmentation using K-means

Pranata and Skinner evaluate the use of the clustering algorithm K-means to segment and target users of a wholesale distributor [7]. The segmentation is based on the annual spending of the customers in the system.

Pranata and Skinner evaluate the use of K-means in this type of environment and come to the conclusion that given the correct parameter tuning, K-means display an exceptional performance with large datasets. K-means is an algorithm designed to group a set of items into $K$ subgroup. The algorithm is dependent on a manually set value for $K$. The authors determine $K$ using three evaluation validation measures: Elbow Method [8], Davies-Bouldin Index [9] and Silhouette Width [10]. A study of K-means will follow in Section 3.3.3 and more about the validation measures in Section 5.2. The authors explain that these groups of similar customers can give the business
more insight about spending patterns in the different groups and connect these with the type of product that these customer groups prefer. Furthermore, this type of technology opens up the possibilities to analyse customers using different aspects as well.

2.3 Dimensionality reduction using Principal Components Analysis

In [11], Han presents an approach to customer segmentation using an artificial neural network architecture using Principal Components Analysis (PCA) as a method for dimensionality reduction. PCA is a widely used method for dimensionality reduction that is explained and evaluated in this report. Han uses PCA in his approach to reduce the dimensionality of the feature space. The customer data is divided into three categories:

1. Consumer information such as gender, age, marital status and education level are grouped in the category Consumer Individual Characteristics.

2. Characteristics such as consumption, payment methods, frequency of consumption are grouped in the category Consumption Characteristics.

3. Consumer attitude factors such as consumer satisfaction, consumer loyalty and consumer confidence are grouped in the category Consumer attitude.

In the study, 80 customers were selected. Through PCA the high dimensional data described above is reduced to six main components accounting for approximately 85% of the contribution of all principal components. These new feature vectors are used in the neural network architecture for customer segmentation. By reducing the data into eight dimensions, the computational time required to run the neural network is decreased.
3 Cluster Analysis

In this section, the research study that was conducted during this project is presented providing a basic knowledge of cluster analysis and customer segmentation.

3.1 Customer Segmentation

An important marketing strategy that is widely used by businesses is customer segmentation [2] [3]. As previously stated, the point of customer segmentation is to split the user-base into smaller groups that can be targeted with specialized content and offers.

The produced customer groups are drawn from user behaviour data which gives the business a deeper understanding of the types of users that exists in the system. The benefit of customer segmentation is twofold. Firstly, a better knowledge about the types of users in a system can lead to better business and marketing strategies. Secondly, a user is likely to use an application more often if he/she always receives relevant content. Another essential point is that if a customer is pleased, he/she is more likely to recommend the application to other people which helps in the expansion of a company.

This type of marketing technique is a subset of a company’s Business Intelligence explained in Section 1.1. To be able to create a set of similar customer groups, an extensive analysis of the available data combined with research and evaluation of clustering algorithms is needed.

3.2 Data

The available data is the most vital part of any clustering algorithm. The most important aspects are the quality and amount of the available data. In order to run some sort of similarity function to cluster items or users in a system, the data needs to be arranged into feature vectors with a set of feature values. To achieve the best results, a large amount of data is needed and more importantly the absence of data points needs to be minimal. The amount of data in most cases is not a problem nowadays since companies store all kinds of user and item data in large databases.

Another important aspect in customer segmentation is to understand the available data. In a system where items are rated using some sort of scale, e.g. a rating from zero to five, it is fairly easy to interpret a user’s preferences. However in systems where the set of items is not predefined, as in a E-marketplace where users upload items which are removed when sold, it is much harder to determine a user’s preference. In this type of
market, it is usual to use data of the type user-item interaction as a measure for preference. These interactions can be e.g. “likes” on items, time spent looking at an item, a comment on an item. This would mean that users that view, “like” and comment the same set of items are similar to each other. However for new users that have a small amount of item interaction, it is impossible to determine in which group they should be placed.

An analysis of the available data points should be made in order to pick a suitable clustering algorithm.

### 3.3 Clustering Algorithms

Clustering algorithms [12] are used to assign users into groups so that users belonging to the same group are more similar than users in another group. The goal of this division is to find meaningful underlying patterns within the data space. User similarity is determined by a distance measure. This section will introduce the most common similarity measures and clustering algorithms.

#### 3.3.1 Similarity Measures

Clustering is highly dependent on defining a relevant similarity or distance measure.

The simplest and most common distance measure [12] is Euclidian distance:

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

where $n$ is the number of features in the data objects $x$ and $y$, and $x_k$ and $y_k$ are the $k^{th}$ attribute of the feature data objects $x$ and $y$ respectively.

Cosine correlation [12] is widely used in the field of recommender systems [1], mainly in collaborative filtering. The main idea behind cosine correlation is to compute the cosine value of the angle that two $n$-dimensional feature vectors form. This is possible using the following equation, where $n$ is the number of features in the data objects $x$ and $y$, · indicates the vector dot product and $\| x \|$ is the norm of vector $x$:

$$\cos(x, y) = \frac{(x \cdot y)}{\|x\| \|y\|}$$

The final distance measure that will be covered in this report is Pearson correlation [1]. This distance measure is also widely used in recommender systems. Pearson correlation computes the linear relationship between two
feature vectors, in other words two feature vectors are similar if a best fitting straight line is close to all data points in both vectors. It is computed using the following function:

\[ Pearson(x, y) = \frac{\Sigma(x, y)}{\sigma_x \cdot \sigma_y} \]  

where \( x \) and \( y \) are two feature vectors, \( \Sigma \) is the covariance of the data points \( x \) and \( y \) and \( \sigma \) is the standard deviation of a feature vector. The result is a value between -1 and 1 where a value close to 1 or -1 means that all values are located on the best fitting line, and values closer to 0 shows that there is little correlation between the given feature vectors.

### 3.3.2 Principal Component Analysis

PCA [13] is a widely used method in statistical analysis. The algorithm is often used whenever high dimensional data is involved, such as facial recognition applications [14], speech/music segmentation [15] and customer segmentation [16] [17]. PCA is used to find underlying structures (Principal Components) in the data by linearly combining the features with each other. PCA is tasked with finding the best principal components (characteristics that differ the most in the dataset) among all possible linear combinations of the available data. The result is an ordered list of principal components that account for the largest amount of variance in the data. A principal component is a summary of multiple features in the original data combined into one. In order to use PCA further as a dimension reduction algorithm, the least contributing principal components can be discarded. Furthermore, the number of principal components is always less than or equal to the number of original features.

PCA is defined as an *orthogonal linear transformation* of the original data. An orthogonal linear transformation is a simple linear transformation which preserves the inner product of the vectors. In other words, the lengths of the vectors and the angles between the vectors are preserved. Orthogonal transformations are therefore only rotations, reflections or a combination of rotations and reflections of the original feature space. This transformation forms a new coordinate system where the values in each axis is based on the computed principal components.
Figure 1: Visual example of how PCA can be used to achieve dimensionality reduction.

Given the data in the leftmost plot in the figure above, PCA is tasked with finding correlations in the data and create a new coordinate system that explains the data with as little information as possible. In this simple two-dimensional example, a correlation between x and y can be seen. More precisely, points with low values for x also have low values for y and points with high values for x have high values for y. In this plot three groups of similar points can be identified with Euclidian similarity. The first group, consisting of two points, is located between $x = 2$ and $x = 4$, the second group also consists of two points located between $x = 4$ and $x = 6$ and finally the last point at $x = 8$ which is well separated from the other groups. Given this correlation, PCA rotates the plane, creating two new axis Principle Component 1 and Principle Component 2. In the rightmost plot above, the same conclusions about the group of points can be drawn only by using PC1 meaning that PCA has successfully reduced the dimensionality of the data from two dimensions to one. For this simple example PCA is not really useful since a two dimensional dataset can be plotted and analysed rather easily, however PCA provides the possible to reduce the dimensionality of high dimensional data which is almost always a necessity in data mining applications.

**Mathematical Details**

Given a data matrix $X$ which $n$ rows and $m$ columns, where each row represents an observation in the dataset and each column represents a particular kind of observation. In the case of customer segmentation in an E-marketplace, a row is a user and a column is an item, a brand or any similar
type of data available in the marketplace. The transformation explained above is defined by a set of $m$-dimensional weight vectors $w_k = (w_1, ..., w_m)_k$. Each row vector $x_i$ of $X$ is then mapped to a new vector of PCA scores $t_i = (t_1, ..., t_m)_i$, such that:

$$t_{ki} = x_i \cdot w_k, \text{ where } i = 1, ..., n \text{ and } k = 1, ..., m$$  \hspace{1cm} (4)

The resulting set of the $t$ vectors are ordered such that the individual variables of $t$ in the whole dataset inherits the maximum amount of variance of $x$.

Without diving into the mathematical formulae, the optimization of the first weight vector can be recognized with a Rayleigh quotient [18] and thus the optimization problem of PCA has its root in this quotient.

Mathematically speaking, PCA is done in five different steps. These steps are briefly discussed below:

**Standardizing**

Standardizing is important in PCA since the variance in the data is maximized to produce linear separability. This means that the data needs to be in the same scale. Standardized scores are derived by subtracting the sample mean from an individual score and then dividing the difference by the sample standard deviation.

**Calculate the covariance matrix**

Calculate the covariance matrix for the standardized input matrix. The covariance between two variables or features can be seen as a description of the similarities between the variance of the variables. In other words, how do the two variables relate to each other.

**Calculate the eigenvectors and eigenvalues of the covariance matrix**

As previously stated, the principal components are used to create the axes of new coordinate system. In order to rotate the data into the new coordinate system, all data points are multiplied with the eigenvectors which indicate the direction of the new axes deduced from the principal components. The eigenvalues are used to determine the magnitude of the new feature space. The eigenvectors and eigenvalues can be deduced from the covariance matrix calculated in the step above.

**Choosing principal components and deriving the new features**

Finally the new dataset is derived by choosing the desired principal components. The amount of principal components chosen depends on the dimension
of the original data and the percentage of variance explained by each of the selected components, i.e. only contributing components should be chosen.

As stated this is a fairly brief overview of the mathematics behind PCA since this is not within the scope of this project. However, In [13], Jolliffe provide a more in depth explanation of these steps and the mathematical formulas behind them.

As any other algorithm, PCA has some limitations and some assumptions that need to hold in order to get meaningful principal components. Mainly the issue is that the original data set needs to have some underlying structure that can be linearly separated since PCA finds “hidden” linear correlations in high dimensional data. Also, during the training phase, the training data needs to be chosen such that all underlying patterns in the data are caught. Therefore choosing a good training set is the most important prerequisite of PCA.

3.3.3 K-means

K-means clustering [19] is widely used in the field of cluster analysis and customer segmentation. K-means is an algorithm designed to group a set of items into \( K \) subgroup or clusters. The algorithm is dependent on a manually set value for \( K \). The \( K \) centroids are initialized to random observations in the dataset. K-means is then tasked with iteratively moving these centroids to minimize the cluster variance using two steps:

- for each centroid \( c \) identify the subset of items that are closer to \( c \) than any other centroid using some similarity measure.

- calculate a new centroid each cluster after every iteration which is equal to the mean vector of all the vectors in the cluster.

This two-step process is repeated until convergence is reached.

The standard implementation of K-means uses Euclidian distance measure described in the section above to find the subset of items that corresponds to each cluster. This is done by calculating mean squared error, which in this case is equivalent with the Euclidian distance, of each item’s feature vector with the \( K \) centroid and choosing the closest result. However, other distance measures can be used instead of Euclidian distance. Aggarwal et al. claim that for high dimensional data, the choice of distance measure used in clustering is vital for its success [20].
3.4 Collaborative Filtering

Collaborative filtering is the most commonly used recommender system in the industry [1]. The main reason for the recent interest and progress is the competition Netflix Prize, in which computer science, data mining and machine learning communities were challenged with the task of beating the accuracy of Cinematch [21] [22], the recommender system developed by Netflix for movie recommendations. The winners, the Belkor’s Pragmatic Chaos team, outperformed the Cinematch system with 10% [23] [24] [25].

Collaborative filtering have even had breakthroughs in the E-commerce scenes such as Amazon.com [6]. The fact that collaborative filtering algorithms are relatively easy to implement and to understand but still manage to provide high accuracy has caught the attention of developers and researchers in the recommender system community.

In this section two collaborative filtering algorithms, user-based and item-based collaborative filtering, are briefly introduced. First, in both algorithms, a “neighbourhood” of users or items needs to be created based on similarity measure. The most common similarity measures used in collaborative filtering are cosine similarity and Pearson correlation presented in Section 3.3.1. The similarities are produced given the history of the users and/or items. In most cases, such as large E-commerce platforms, this history data is large and the algorithms need to operate directly on the history data. Therefore a well thought scalability plan based on the data needs to be constructed.

User-based Collaborative Filtering

The user-based collaborative filtering algorithm tries to identify groups of similar users given their behaviour. Given this history data, the neighbourhood of users explained above is calculated and the top $n$ neighbours are picked for the recommendations. The recommendation items that will be shown to a user within a group is based on the top rated items within the group that the user has not yet viewed. As these neighbourhoods play a huge role in the success of the algorithm, the user-base needs to be stable and each user should have a rather large amount of data for the recommendations to be correct. This fits the scene of E-marketplaces since the neighbourhoods are constructed from each user’s behaviour and not by the items themselves since they are unique and disappear from the system.

Item-based Collaborative Filtering

The item-based approach is fairly similar but focuses on the items of the
system instead. Each item will be placed in a group of similar items. The recommendations given to a user by the system is based on the items that the user seems interested in according to previous views, ratings and purchases. The most similar items in the neighbourhoods compared to the user’s most viewed, liked and purchased items are chosen as recommendations to the user. Item-based collaborative filtering is mostly used in the E-commerce scene such as Amazon.com, since there is a fixed set of available items that are unlikely to be removed.
4 Plick System Information

Plick is an E-marketplace application that was first launched in 2013. It has grown significantly over the years and boast today a user-base in the tens of thousands and over a hundred thousand clothing items. In this section, a short description of the system, its features and the data that can be mined from the database are be described.

4.1 System Description

The presentation of items in the Plick application is mainly done in feeds. The first page of the application is a tab view where users can either view a feed with all the clothing articles in the system ordered by a weighted combination of the amount of “likes” of the item and its upload time or a feed of clothing articles uploaded by sellers that the user is “following”. The initial view is the large feed including all the items in the system where highly “liked” and recently uploaded items are ordered first. Additionally there is a search bar on the top of the application where users can filter the feed by category, geographical location, gender and custom search queries. A user can also search for a specific user or browse through a list of users. The figure below shows the first page of the Plick application.
As shown in Figure 2, each item in the feed includes the price, size, excerpt of the description and image of the item. The full item description can be found by tapping/clicking on the image. In this view, Figure 3, a user can also contact the seller by starting a conversation.

Figure 2: First page of the Plick application showing the trending feed.
A major issue with this type of feed is that all the users see the same items. A better approach would be to present a personalized feed for each user based on user behaviour. This is where the cluster analysis and customer segmentation comes into play. This type of feed may also rekindle interest in items that are lost in the “never ending” feed that exists today since it is mainly ordered by upload time.

4.2 Features

Plick’s vision is to create an intuitive and simple design but still manage to provide a set of essential features. Users can follow each other and “like” items in the system. By following a user, a subscription to all the items uploaded by that user is made and displayed in the “following” tab. Users also have the possibility to connect to Facebook, giving them the possibility to post their items on Facebook to attract more buyers. Facebook is also used for verification.
Another appreciated feature is the guide on how to upload an item using the camera on their phone. This benefits both Plick since better pictures are displayed in the applications and the users since their items are more aesthetically pleasing and therefore attract more potential buyers.

4.3 Data in Plick

The data in Plick is stored in a PostgreSQL database and include information about users, items and interactions between these two entities. Every user in the system have a name, email, location and description connected to them. These fields are however not required in order to create a Plick account. Furthermore, the database includes information about items. Each item has a seller, location, price, gender (male, female or unisex), size and description. However, only the seller location and prize is required by the system. This choice was made since Plick want to keep the upload process as smooth and easy as possible.

Furthermore each item in the system is associated with a clothing brand, which is used to group a subset of items together.

In order to create a good classification of the Plick users, their behaviour in the application is used to construct feature vectors. A user’s preference is based on three types of item interactions:

- Whenever a user views an item it indicates a certain interest in that item.
- A “like” on an item indicates an even stronger interest in that item.
- A started conversation with a seller is a strong interest indicator since the user in question is a potential buyer.
5 Evaluation of Clustering Analysis

In this section the data analysis is presented and a method is chosen. This choice is motivated using results drawn from various tests done on the Plick data.

5.1 Data Preprocessing

Firstly, in order to compare users, a rating system needs to be defined. In the case of Plick, a user’s interest in an item can be defined by several weighted variables. The rating is therefore defined as an aggregated score which is derived from the data points discussed in Section 4.3. In order to be able to analyse the data and compare users, these scores needs to be standardized into the same scale. The primary data points used in the analysis of the Plick users during this project were item views and item likes. A visualization of the resulting user-item ratings matrix for \( n \) users and \( m \) items is shown below, where \( n \approx 35000 \) and \( m \approx 140000 \).

<table>
<thead>
<tr>
<th>User</th>
<th>( i_1 )</th>
<th>( i_2 )</th>
<th>( i_3 )</th>
<th>( i_4 )</th>
<th>...</th>
<th>( i_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>( r_{11} )</td>
<td>( r_{12} )</td>
<td>( r_{13} )</td>
<td>( r_{14} )</td>
<td>( \rightarrow )</td>
<td>( r_{1m} )</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>( r_{21} )</td>
<td>( r_{22} )</td>
<td>( r_{23} )</td>
<td>( r_{24} )</td>
<td>( \rightarrow )</td>
<td>( r_{2m} )</td>
</tr>
<tr>
<td>( u_3 )</td>
<td>( r_{31} )</td>
<td>( r_{32} )</td>
<td>( r_{33} )</td>
<td>( r_{34} )</td>
<td>( \rightarrow )</td>
<td>( r_{3m} )</td>
</tr>
<tr>
<td>( u_4 )</td>
<td>( r_{41} )</td>
<td>( r_{42} )</td>
<td>( r_{43} )</td>
<td>( r_{44} )</td>
<td>( \rightarrow )</td>
<td>( r_{4m} )</td>
</tr>
<tr>
<td>...</td>
<td>( \downarrow )</td>
<td>( \downarrow )</td>
<td>( \downarrow )</td>
<td>( \downarrow )</td>
<td>( \rightarrow )</td>
<td>( \rightarrow )</td>
</tr>
<tr>
<td>( u_n )</td>
<td>( r_{n1} )</td>
<td>( r_{n2} )</td>
<td>( r_{n3} )</td>
<td>( r_{n4} )</td>
<td>( - )</td>
<td>( r_{nm} )</td>
</tr>
</tbody>
</table>

Table 1: User-item ratings matrix.

Due to the large amount of items in the system, it is quite obvious that a user has interacted with a small subset of the items. This is an issue when trying to analyse the data since the user-item ratings matrix is dominated by zeroes. More precisely, a 96% data sparseness is detected which means that only 4% of the matrix values are non-zero.

In order to tackle this problem, item brands were used in the analysis instead of the items themselves. Each item is assigned a clothing brand on creation out of the 76 brands available in the system. By using brands
instead of items, the dimension of the resulting matrix is more compact and much less sparse.

To further reduce the sparseness of the matrix, only the top 20 brands are chosen out of the available 76 since they account for approximately 89% of the items in the Plick dataset. This results in a $35000 \times 20$ User-Brand matrix. An analysis of these users was made, and the conclusion was that a majority of the users should be excluded from the research since they have not interacted enough with the items in the Plick marketplace. A minimum of 50 item interactions per user was set during the evaluation of the system resulting into a $7000 \times 20$ matrix.

### 5.1.1 Principal Component Analysis

Once the User-Brand matrix was constructed, the dimensionality reduction algorithm Principal Component Analysis was used in order to be able to visualize and understand the data. To get a better picture of the data, the first two Principal Components were used as matrix columns:

![Figure 4: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user.](image)

The first and second Principal Components account for approximately 76% of the variance of the original data, which simply means that 76% of the original 20 dimensional matrix is explained by a two dimensional matrix instead. A large concentration of users was detected near the origin of the plot and no clear separation of clusters can be detected using Euclidian distance.
The same experiment was made using three Principal Components, creating the three dimensional plot below:

![Scatter plot of Plick users](image)

Figure 5: Scatter plot of the Plick users where the first, second and third Principal Components are used as x-, y- and z-axes. Each dot represents a user.

The first three components result in a 83% explained variance of the original dataset, however no clear cluster separations can be detected in the figure above either. Therefore, a different approach needs to be introduced for clustering which is not based on Euclidian distance due to the large concentration of data points near the origin. This method is discussed in Section 5.2.

### 5.1.2 Data analysis

Since no clear cluster separations can be deduced from Figure 4 and 5, a deeper data analysis was made in order to find underlying properties that can be used to get a clearer picture of the Plick data. These properties are based on the brand ratings system described above.

The main focus during this phase of the project was to figure out if and how the behaviour of the Plick users change over time. In other words, do users that have registered early browse content in the application differently compared to users that have registered recently? In order to answer this question two methods were tested.

Firstly a video plot was constructed, where a hundred users are plotted per step in chronological order.
Figure 6: Video scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user.

In the figure above a single step of the video plot is shown. The idea with this plot was to determine if the values of the Principal Components discussed in the section above change over time, however no such conclusions could be made since there is a large concentration of users near the origin of the plot throughout all the steps in the video.

Another idea that leads to the same conclusion was to create a histogram plot of a single dimensional array constructed by the following equation:

\[ \text{histogram
data} = X \bullet PCA_1(X^T), \]  
where \( \bullet \) is the dot product operand, \( PCA_1 \) is a function that returns the first principal component of a matrix, \( X \) is the user-brand matrix discussed in Section 5.1 and \( X^T \) is the transposed user-brand matrix.

The resulting matrix is of dimension 7000 \( \times \) 1 which is inserted into a histogram plot. The first and last thousand values of this matrix, representing the first and last 1000 users, are selected and plotted in the histogram below.
Figure 7: Histogram comparing the trends/behaviour of the first thousand users with the last thousand users registered in the Plick system.

However, in the figure above, it is clear that the first and last thousand registered users do not differ much in terms of behaviour since we can see the same trends in the blue and orange histograms.

5.2 Cluster Analysis

In order to segment the users into clusters, a non-Euclidian approach is needed due to the analysis described above. To get a better picture of the data shown in Figure 4, each user is assigned a favourite brand based on the amount of views of the brands. Each brand is assigned a colour and a new plot is constructed where the users are grouped by favourite brand.
Figure 8: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user where the colour represents the most viewed brand for that user.

In this figure, an angular correlation can be seen, where most users with the same favourite brand are confined between two angles, in a cone-like shape, in the graph as shown in the figure below.
Figure 9: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user where the colour represents the most viewed brand for that user. Only the brands Adidas and Zara are highlighted.

In the figure above, the angular correlation of brands is much more visible. In this case, two different types of clothing brands *Adidas* and *Zara* were highlighted. As the figure shows, these two groups are well separated further proving the point that some type of angular similarity measure can be used to cluster users.
Figure 10: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user where the colour represents the most viewed brand for that user. Only the brands Adidas, Nike and Zara are highlighted.

By adding Nike to the scatter plot, the Nike and Adidas points are expected to be similarly grouped since both brands manufacture similar items, which is the case in the figure above. Since the original data is only based on user behaviour, it is quite interesting to see that similar brands are close to each other.

Given this angular correlation shown above, a clustering algorithm with an angular similarity measure can be constructed. To achieve this, a Python implementation of K-means was used with the similarity measure \textit{cosine similarity} described in Section 3.3.1.
Figure 11: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user where the colour represents the cluster of that user.

In the plot above, the same angular clustering can be seen. However it is pretty clear that ten clusters is not the optimal number of clusters in this dataset since multiple clusters are overlapping in the plot above. In order to determine the correct amount of clusters, some evaluation methods were used:

- The first method called Silhouette score method, is used to determine a score between 1 and -1 for each sample in the dataset, where a data point with a score close to 1 is considered good and -1 is considered bad. This score is calculated with the equation $\frac{b - a}{\max(a, b)}$ where $a$ is the mean intra-cluster distance and $b$ is the mean nearest-cluster distance. The resulting silhouette scores are rather low, shown in the plot below.
According to the figure above, the best silhouette score can be observed when using 3 clusters with the score of approximately 0.32. Since an optimal silhouette score is equal to 1, it is obvious that this set of clusters is far from optimal.

- The second method used to identify the correct amount of clusters is the elbow method. This method looks at the average distance between cluster centroids and the matrix vectors for a range of number of clusters. A new cluster should be added if the average distance decreases noticeably. By plotting the average distances produced, an "elbow point" can be detected, where the distance starts to converge. In the figure below, the largest drop, and therefore also the "elbow point", can be observed at $K = 3$ where $K$ is the number of clusters.
However, the "elbow point" is not perfect since the average distance does not seem to converge to a value. As stated above, it seems that this set of clusters is far from optimal. In Section 7, a few possible suggestions to optimize these clusters are presented.

The resulting clusters are visualized in the figure below:

Figure 14: Scatter plot of the Plick users where the first and second Principal Components are used as x- and y-axes. Each dot represents a user where the colour represents the cluster of that user.

In the two dimensional plot above it seems like the orange cluster overlaps with the two other clusters, however when viewing the three dimensional version of the plot, a clear separation can be seen.
Figure 15: Scatter plot of the Plick users where the first, second and third Principal Components are used as x-, y- and z-axes. Each dot represents a user where the colour represents the cluster of that user.
6 Implementation

In this section, the implementation of the final system is described. The implementation was divided into two stages: a cluster analysis stage and a data visualization stage.

6.1 Cluster Analysis

The first part of the project consisted of gathering and analysing the data of the Plick PostgreSQL database. To achieve this, the programming language Python was used in combination with the Python libraries Psycopg2, a PostgreSQL database adapter, and SciPy, used for scientific computing. This program is responsible of three major parts of the system:

1. Querying the Plick PostgreSQL database for all the brand views and likes for each user and construct a User-Brand matrix described in Section 5.1. This part of the system is also responsible for the data preprocessing and converting brand views and likes into ratings.

2. Convert the highly dimensional User-Brand ratings matrix constructed in the previous step into a three dimensional matrix using Principal Components Analysis. This was done using SciPy. The latter was also used in the development stages for generating clusters given some clustering algorithm such as K-means. At this initial stage of the development, all the plotting and data visualization was made with the Python plotting library Matplotlib.

3. Finally, the User-Brand ratings matrix is used in the clustering algorithm K-means with a cosine similarity measure.

6.2 Data Visualization

During the initial stages of the project it became clear that no straightforward clustering could be deduced from the available data. Therefore, a visualization tool was needed to get a better picture of the data and how it can be used to produce better user segmentation. To make this visualization tool as easily accessible as possible, a website was constructed.
Since the backend discussed above was written in Python, the web framework Django was used. Django was a pretty obvious choice since it is by far the most common and well suited web framework written in Python. There exists other Python web frameworks such as web2py or TurboGears, however Django was a better choice due to its stable nature and popularity.

The frontend of the visualization tool is constructed using the regular web frontend programming language JavaScript and the markup languages HTML and CSS. The python plotting tool Plotly was used to construct plots in the Django backend which are rendered in the frontend templates via HTML blocks.

Also, page speed optimization was considered during the development of the visualization webpage. Since a rather large database is queried by the Python backend it takes a few minutes before the User-Brand matrix is constructed and filtered according to Section 5.1. The first solution was to save the large matrix in the browser’s cache the first time the website is visited which means that the website will only have a long loading time the first time it is visited. Going forward, the Python Multithreading should be used in order to speed up the construction of the User-Brand ratings matrix, however this is not present in the current version of the visualization tool.
6.3 Final system

![System architecture flowchart of the final system.](image)

The algorithm used during the project is shown above. Briefly, data is gathered from the Plick Database and reconstructed into a User-Brand ratings matrix. The dimensionality reduction algorithm Principal Components Analysis is then used to reduce the highly dimensional matrix into a two or three column matrix which can be plotted in the visualization tool described above. Furthermore, a K-means algorithm is used to create cluster groups based on the conclusions drawn above. Finally, all the users are given a cluster identifier which is stored in the Plick database for further use.


7 Conclusion

In the past years, research in the fields of big data analysis, machine learning and data mining techniques is getting more frequent. Initiatives such as the Netflix Prize have sparked life into these fields which have become a central part of many businesses for analytics. Furthermore, machines with high computational power provided by e.g. cloud computing services such as Amazon Web Services are getting more easily accessible and cheaper. This computational power combined with the new advanced machine learning and data analysis techniques work in symbiosis to achieve the goals of businesses, which is to understand the market better and to be able to make decisions directly based on user data.

This project is based around a second hand vintage clothing E-marketplace called Plick. The ultimate goal of this project is to segment users into smaller groups which can be viewed as groups of users that seem to like the same type of clothing. These groups can be used to send out personal newsletters and recommendations based on clothing articles that are highly rated by the group but that has not yet been viewed by the user in question. This is a quite obviously an extremely powerful tool for reactivating users. If this idea is taken even further, the Plick application and website could include a personal feed with these items. Given that these items really are of interest to the users, the amount of sold items should increase since users see interesting items that would have ”been lost” in the long feed of items which is mainly ordered by upload time. This benefits both the buyers and the sellers which in turn are more likely to recommend Plick to friends and family.

In this report, a customer segmentation approach is presented and evaluated. The available data points in the Plick database is item views, likes and conversations, however the initial version of the system evaluated in this report only makes use of views. During the final stages of the implementation a version where likes were incorporated in the system was tested, however no significant improvements were observed. That said, the focus during this project was on the clustering analysis, the preprocessing stage of this project could be improved by incorporating likes and conversation to the ratings calculations using some weighting of these three data points. Another interesting data point is purchases but unfortunately since Plick is a second hand store, most users decide to meet up in order to purchase or sell items. An idea discussed internally in Plick to resolve this issue is to require the seller to add a buyer to an item whenever he/she sets the item state to sold in the application.

Another issue with the Plick data is that most users are not that active in
the application, meaning that they only view a few of the 160 thousand items available in the system resulting in a highly sparse matrix. Therefore, during this project, only a subset of the user base were included in the analysis. Due to this fact, new users are not included into the clustering. Since the end goal of this project is item recommendations, a hybrid solution combining the K-means cluster analysis shown in this report for highly active users and User-based Collaborative Filtering recommender system approach [4] for less active users could be of interest. Elvander’s approach is better suited for users with low activity since the Collaborative Filtering algorithm constructs a neighbourhood of similar users for each user, meaning that the groups are not generalized. Similarly to the clusters discussed in this report, the neighbourhoods can be used to find clothing articles not yet seen by the user in question.

Going forward, the recommendations created by the system should be evaluated with real users from the marketplace. Feedback from users should give a clear picture of whether or not the clusters are good. As pointed out in Section 5.2, it was quite clear that the results from the silhouette scores and elbow method were not optimal. In order to achieve better clusters, a better rating system for brands should be implemented that utilizes views, likes and conversations to construct ratings. A more extensive analysis of how the data should be preprocessed and weighted needs to be made in order to get a better result from the clustering algorithm. Furthermore, K-means was used as the only clustering algorithm which might not be optimal in this case. Therefore, other clustering techniques should be tested and compared to the K-means implementation. Another vital part of the success of the algorithm is which users and brands that should be filtered out. In order to test different thresholds for minimum user activity and brand views, the visualization tool shown in this report should give the Plick team the possibility to alter the User-Brand matrix analysed by the clustering algorithm by e.g. setting different thresholds and by adding/removing brands. For each run, evaluation plots should be updated and made visible in order to compare different runs.
References


