An Artificial Intelligence Based Insurance Broker

Project Report

Jonas Tischer, Kuan Min Chen, Radost Petrova, Rahul Taneja

submitted to the
Chair of Artificial Intelligence
Prof. Dr. Heiner Stuckenschmidt
University of Mannheim

12.03.2019

Abstract

With its rising popularity and application in areas such as healthcare and business, Artificial Intelligence (AI) is also on the way to revolutionize the insurance broker industry. Many Tech Startups are entering the field of insurances, trying to disrupt the existing business models. Nevertheless, the insurance sector is still dominated by big insurance companies which have not substantially evolved throughout the decades and are suffering a lot of inefficiencies due to repetitive manual work. Introducing AI into their ecosystem could help overcome these problems. In this paper we want to shed light onto the current technological state of the insurance broker industry and how AI may transform it. Furthermore, we provide an Recommender System for dental insurances using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), a popular Multiple Criteria Decision Making Method (MCDM). In addition we design an architectural model which may serve as an example of how to implement an insurance recommender system as a web application with state of the art technology.

Keywords— Recommender System, MCDM, Insurance Broker, AI, TOPSIS

Contents

1 Introduction & Motivation

1.1 The Trend of AI-based Insurance Brokers

Globally the increased use of the Internet and smartphones has contributed to a fundamental disruption of traditional business models. In the course of this digitalization many new market players have emerged and over time competitors have become established companies. Now companies are shifting towards another disruptive trend - Artificial Intelligence (AI). AI is increasingly entering many areas of our lives and transforming current businesses and industries. It can be found in many aspects of the insurance industry and will continue to reshape the future of this sector, which will help companies improve their existing processes and increase customer satisfaction.

This paper reviews one specific part of the whole insurance ecosystem, that is the job of an insurance broker. The traditional role of a broker is to assess the user's coverage needs and negotiate best terms, conditions and pricing with insurers. In other words, the broker is responsible for finding and providing an optimal insurance plan to the customer based on the customer's personal needs. Nowadays there are online insurance brokers available, such as Check24, Verifox and co, that deal with the recommendation of insurances to users. However, they have not been able to completely fill the role of a real broker, that can provide product advice to customers and the demand for human brokers has remained the same.

Despite the growing number of recommendation systems in all areas of our lives, real professionals have been able to assert themselves against online recommendation platforms because they have a certain ”gut feeling” or intuition for the right recommendation based on the personal needs of the customer. The brokers' perception is often difficult to quantify and model and therefore difficult to automate, but with the establishment of AI it would be possible to embody the core competence of a broker, namely to provide the best recommendation of a suitable insurance.

1.2 Classification of Insurance Brokers

For a better understanding of where the trend might lead insurance brokers in the future we created a classification of three different types of insurance brokers and their varying levels of AI-support:

1. Human Insurance Broker - Makes recommendations based on a clear set of questions that he asks the customer, his many years of experience as well as his intuition to some degree. Possibly uses decision support programs, but no AI apart from that.

2. Insurance Broker Portal - Exceeds the basic functionality of an Insurance Comparison Portal by additionally acting as a broker and directly selling Insurance plans to customers. It is a kind of self-checkout service where the user selects the desired insurance type and has to actively search for the information they ares looking for. A score for each policy is calculated based on some user independent value-price metrics and the highest ranked insurance plans are highlighted.

3. Fully Autonomous Insurance Broker- Automatically recommends the customer an insurance based on analyzing the customer's risks. It picks the best plan for the customer suited to the personal needs based on advanced algorithms. On top of the functionalities of a Type 2 insurance broker, the agent also takes care of finding relevant insurances autonomously and is able to negotiate an individual pricing model for the customer with the insurance companies based on the customer's personal data.

1.3 An Innovative Insurance Broker

An example of a company that is aiming to become a Type 3 AI-empowered insurance broker is Clark.[1] Clark is a personal insurance concierge for customer interaction and portfolio management. It collects all of the customer policies in one place and provides easy access to all necessary information. The policies are analyzed and compared to the customer needs to determine whether the cover is appropriate, that is, whether the customer is over- or under-insured, and whether the cover should be purchased by the customer. Yet, the focus of Clark is not only on making the best possible recommendation of an insurance plan for the customer, but to provide the full package of the best possible services. These service include reviewing current insurance policies, recommending new or better insurance policies, having a real contact person and to assist the customer with the claiming process.

1.4 Current Challenges of AI based Insurance Brokers

An underlying challenge of all types of insurance brokers is the non-trivial problem of recommending the best insurance policy to the customer. Therefore, it becomes much more important to have a functioning recommendation system that is able to make a highly personalized recommendation to ensure that the user is provided with the best suitable plan regarding their individual needs.

As this is not easy task and many companies are working on this step, we have decided to develop an algorithm that recommends the most ideal insurance to the customer based on the customer's risks and preferences. To visualize our results we have developed a demo web application, which illustrates the process and can serve as a basis for further developments.

The remainder of this paper is organized as follows: In Chapter 2 we review typical recommender systems from the literature and how they may be applied to our problem. In Chapter 3 we explain the architecture, the algorithm of how we compute the insuranre recommendation. Chapter 4 describes the implementation of the demo Java and web applications. In Chapter 5 we explain how we can evaluate the results. In Chapter 6 we wrap up the paper with conclusion and limitations.

[1] https://www.clark.de/de/ueber-uns

2 Literature Review on Recommender Systems

In the following section we will focus on the problem of recommending the best policy to the customer. Therefore, we will explore different types of recommender systems and see how they can be applied to the insurance domain.

2.1 Types of Recommender System

Recommendations in the online sector are nowadays implemented into almost any available service ranging from booking services and online shopping to entertainment. However, with the broad range of applications for recommender systems, the underlying approaches differ greatly. One classification of different types of Recommendation systems is the following, based on [Burke] and summarized by [Hinduja and Pandey2017]:

- Content-based: Content based recommender system suggests products based on the text information of the item. It recommends items to a user, similar to the ones which have been preferred in the past.
- Collaborative filtering: Collaborative filtering based recommender system is similar to a content-based recommender system with the difference that it calculates the similarity between users instead of items.
- Demographic: This recommendation system categorizes users or items, based on the personal attribute of the user, and make suggestion upon demographic categories.
- Utility-based: The recommendation is done based on the utility of the product to the user; the product with the maximum utility is suggested.
- Knowledge-based: The recommendation relies on domain knowledge, which can be extracted by a domain expert or extensive literature survey.
- Hybrid: A combination of two or more recommendation techniques to overcome the limitations of each system alone.

Since all of those different approaches work differently it is important to understand their individual strengths and weaknesses in order to select a suitable approach for the specific domain.

[Adomavicius and Kwon2007] have outlined in their work the problem with traditional recommender systems such as content-based or collaborative filtering that these mostly use only a single criterion to represent the utility of an item to the user such as the Imdb rating for movies.

If we would transfer the approach of single-rating to our problem of recommending the optimal policy to the user it would turn out that every policy gets the same utility value for each individual user. This phenomenon exists on the Check24 [2] comparison portal which calculates one static rating for each policy which represents the utility for this plan.

[2] www.check24.de

But in order to achieve a personalized recommendation every plan should have a different rating depending on the user's needs and risks. While single-rating recommendations might be applicable for trivial products like books, movies or fashion product where non-personalized or bad recommendations just fail to please a customer in the worst case, the effects might be much more severe in the insurance domain if the policy does not cover the important risks of a customer. This means that our system must be able to predict the policy's overall utility value for each user, who has specific risks, needs and preferences, so that it can compare all the policies based on those individually calculated scores.

2.2 Multi Criteria Decision Making

One of the problems in the insurance domain is the fact that customers or insurance brokers have to deal with a bundle of decision factors that are often quite complex, interrelated or poorly defined and hard to grasp. This means that they are facing multi-criteria problems with an increasing complexity which humans cannot handle with their intuition anymore. Especially when selecting an insurance the stakes are high, so it becomes very important to break down the problem and properly evaluate multiple criteria.

A Recommender System in the insurance domain has been described in the paper [Hinduja and Pandey2017]. In their work they built a system that recommends the user a life insurance based on certain product criteria and user preferences. The system takes users' demographic information and preferences as an input and computes an utility value. As an approach they chose to combine Intuitionistic Fuzzy Sets (IFS's) for figuring out user's preferences with Grey Relational Analysis (GRA) for estimating the utility of a policy for the user. They reported that they had a hit rate of 92.6 percent on test group of 600 insurance buyers.

[Gandhi et al.2015] did an extensive literature review and analysis of methods for solving MCDM problems in which they identified 11 different methods. They reviewed each methodology in detail and listed their advantages, disadvantages and areas of application. After considering all methods carefully we decided to take a deeper look at the Technique for Order Preferences by Similarity to Ideal Solutions (TOPSIS) approach. It seemed to be a very promising approach due to its advantages which are the simplicity of the process, easiness to use and program and the fact that the number of steps remain the same regardless of the number of attributes.

2.3 TOPSIS

According to [Zavadskas et al.2016] TOPSIS has become the second most popular MCDM approach. It was first introduced 1981 by Yoon and Hwang and is *"an approach to identify an alternative which is closest to the ideal solution and farthest to the negative ideal solution in a multi-dimensional computing space"* [Qin et al.]. The idea of how TOPSIS works is shown in figure 1.

A main disadvantage of TOPSIS is that its use of Euclidean Distance does not consider the correlation between attributes. Furthermore it is difficult to weight attributes and keep up consistency of judgment, particularly with more attributes. To deal with the correlation, the datasets will need to be transposed into a new dimension, but the result would be hard

Step 1.	Create an evaluation matrix consisting of m alternatives and n attributes, with the intersection of each alternative and attributes given as x_{ij} , we therefore have a matrix $\left(x_{ij}\right)_{m\times n}$. *Assumptions: 1. The value and suitability of each attribute should be linearly decreasing or increasing.* *2. The attributes should be independent.*
Step 2.	Normalise the matrix $\left(x_{ij}\right)_{m\times n}$ to form the matrix $R=\left(r_{ij}\right)_{m\times n}$, using the normalisation method $$\mathrm{r}_{ij}=\frac{x_{ij}}{\sqrt{\sum_{i=}^{m}x_{ij}^2}}, i=1,2,\dots,n, j=1,2,\dots,n$$
Step 3.	Calculate the weighted normalised decision matrix $V=\left(v_{ij}\right)_{m\times n}=\left(w_j r_{ij}\right)_{m\times n}$, $i=1,2,\dots,m$, where $$w_j=\frac{W_j}{\sum_{j=1}^{n}W_j}, j=1,2,\dots,n,$$ so that $\sum_{j=1}^{n}w_j=1$, and W_j is the original weight given to the attribute $v_j, j=1,2,\dots,n$.
Step 4.	Determine the negative ideal solution (ideal worst alternative) A^- and the ideal positive solution (ideal best alternative) A^+: $A^-=\{\langle max(t_{ij}\|i=1,2,\dots,m)\|j\in J_-\rangle,\langle min(t_{ij}\|i=1,2,\dots,m)\|j\in J_+\rangle,\}\equiv\{t_{wj}\|j=1,2,\dots,n\}$, $A^+=\{\langle min(t_{ij}\|i=1,2,\dots,m)\|j\in J_-\rangle,\langle max(t_{ij}\|i=1,2,\dots,m)\|j\in J_+\rangle,\}\equiv\{t_{bj}\|j=1,2,\dots,n\}$, where, $J_+=\{j=1,2,\dots,n\}$ j associated with the attributes having a positive impact, and $J_-=\{j=1,2,\dots,n\}$ j associated with the attributes having a negative impact.
Step 5.	Hwang and Yoon[33] originally proposed Minkowski's L_p metrics to calculate distance measures between target alternative i and the worst condition A^- $$S^-=\left(\sum_{j=1}^{p}\left\|v_{ij}-v_j^-\right\|^p\right)^{\frac{1}{p}}, i=1,2,\dots,m, j=1,2,\dots,n,$$ and the distance between the target alternative i and the best condition A^+ $$S^+=\left(\sum_{j=1}^{p}\left\|v_{ij}-v_j^+\right\|^p\right)^{\frac{1}{p}}, i=1,2,\dots,m, j=1,2,\dots,n,$$ where S^- and S^+ are distances from the target alternative , i to the worst and best conditions (alternatives), respectively. If $p=n$ then Tchebycheff distance, if $p=2$ then Euclidean distance, and if $p=1$ then Manhattan (city block) distance. *Remark: Commonly in most applications, extensions and modifications is used Euclidean distance*
Step 6.	Calculate the similarity to the worst condition: $$C_i=\frac{S^-}{(S^++S^-)}, 0\le C_i\le 1, i=1,2,\dots,m.$$ $C_i=1$ if and only if the alternative solution has the best condition; and $C_i=0$ if and only if the alternative solution has the worst condition.
Step 7.	Rank the alternatives according to C_i $(i=1,2,\dots,m)$

Figure 1: Process of classical TOPSIS method illustrated by [Zavadskas et al.2016]

to explain with the combination of attributes. However, as TOPSIS is simple to use and we do not have any dependencies between the attributes in the plan datasets, we choose to use it as the core part of our recommendation algorithm.

In conventional MCDM methods such as TOPSIS, the weights and ratings of criteria are known exactly, but human judgments and preferences are often ambiguous and cannot be estimated by an accurate numerical value. Thus, we enhance the traditional weighting scheme of the TOPSIS to give the user the possibility to express their own preferences but at the same time keep the weighting rational. With this approach we try to reflect real risks of the user based on statistics, of which the user might not be aware and reduce the subjectiveness of the user's choices.

3 Insurance Recommender System

3.1 Recommendation as a Multi-Criteria Decision Making Problem

For designing our recommendation problem we follow one of the classical methodologies described by [Roy1996] in the book Multicriteria methodology for decision aiding and summarized by Adomavicius [Adomavicius and Kwon2015] as follows:

1. **Defining the object of decision**. Define the set of alternatives (items) upon which the decision has to be made and the rationale of the recommendation decision.
2. **Defining a consistent family of criteria**. Identify and specify a set of functions that declare the preferences of the decision maker (targeted user) upon the various alternatives. These should cover all the parameters affecting the recommendation decision and be exhaustive and non-redundant.
3. **Developing a global preference model**. Define the function that synthesizes the partial preferences upon each criterion into a model that specifies the total preference of a decision maker regarding a candidate alternative.
4. **Selection of the decision support process**. This covers the design and development of the procedure, methods, or software systems that will support a decision maker when making a decision about the set of alternatives (items), in accordance to the results of the previous steps.

3.2 The Proposed Recommender System

Defining the object of decision

In order to keep the complexity of the project low, we decided to focus on only one specific insurance, namely dental insurances. A dental insurance is complex enough to transfer the described solution easily onto similar problems and serves well for our exploratory approach. As a dataset of possible insurance policies we crawled the check24.com webpage in order to get real data. It contains 48 different dental insurance policies provided by 29 insurance providers. Of course this set is not exhaustive and not all insurance providers are represented in this dataset but we are sure that it is enough for showing the effectiveness of our solution.

Defining a consistent family of criteria

As mentioned before, we will develop an algorithm that proposes the user the most promising policy possible. For the input, we take into account user's medical data and user preferences. The recommendations are based on performance of products upon certain criteria, user preferences, risk statistics and user's medical data.

This results together in four different categories of criteria we are looking at in order to find the optimum weight which would be used for TOPSIS:

1. **Policy Features**
2. **User Medical Data**
3. **Age Related Risks**
4. **User Preferences**

Policy Features. We have narrowed down the pool of policy features and selected only the 12 most important ones for evaluating each Dental Insurance plan. All the information about a plan can be extracted from the policy file that is usually publicly available as a PDF on the insurance providers' website.

1. Dentures - Crowns and bridges
2. Dentures - Implants
3. Dentures - Inlays
4. Dentures - Veneers
5. Treatment - Plastic fillings
6. Treatment - Root Treatment (when health insurance pays share)
7. Treatment - Periodontal Treatment (if health insurance pays part)
8. Treatment - Fissure sealing
9. Professional tooth cleaning
10. Professional tooth cleaning -waiting period
11. Orthodontics for adults
12. Orthodontics for children

User Medical Data. In a question form we ask the user about his personal and medical data.

Age Related Risks. Based on the gender and age of the user we incorporate statistics from the Dental Insurance Domain that reflect the need of a user for certain services in a particular age. Those statistics are partially available for the public where as some are proprietary and have been collected by insurance companies of a long period of time.

User Preferences. The user can select multiple preferences from the below list of preferences. Each of those preferences is related to at least one policy feature. This relationship is modelled in a Preference/Feature Matrix where a relation is shown with binary values.

1. Tooth cleaning,
2. Brackets (e.g. mini, plastic, ceramic brackets)
3. Super elastic bows
4. Lingual retainer
5. Invisible dental splint (Invisalign)
6. Internal braces (lingual technique)
7. Eco friendly company
8. Stable contribution in old age
9. Treatment with private doctors
10. No/Low waiting period
11. Functional diagnostics
12. No/Low personal contribution

Developing a global preference model

Definition 1. Let X be the vector of user answers to the question form.

Definition 2. Let A be the Feature Importance matrix related to the user's answers about his medical data.

Definition 3. Let B be the Age related Risk Matrix. This matrix is a probability matrix which is created based on statistics available in the Dental Insurance domain. It reflects how likely it is that a person in a certain age will need to use a particular service.

Matrix A and B are shown in figures 2 and 3 below.

	User's Answers->	X1	X2	X3	X4	X5	X6	X7
Features	dentures - Crowns and bridges				0.005			0.05
	dentures - implants				0.003		0.004	0.02
	dentures - inlays				0.006			0.03
	dentures - veneers				0.003		0.004	0.02
	treatment - Plastic fillings				0.007			
	treatment - Root treatment (when health insurance pays share)			0.007				
	treatment - Periodontal treatment (if health insurance pays part)		0.05			0.005		
	treatment - fissure sealing				0.007			
	Professional tooth cleaning		0.13					
	Professional tooth cleaning -waiting period		0.07					
	Orthodontics for adults	0.18(age>17)						
	Orthodontics for children	0.18(age<17)						

Figure 2: Feature Importance Matrix

Features-> / Age	dentures - Crowns and bridges	dentures - implants	dentures - inlays	dentures - veneers	treatment - Plastic fillings	treatment - Root treatment (when health insurance pays share)	treatment - Periodontal treatment (if health insurance pays part)	treatment - fissure sealing	Professional tooth cleaning	Professional tooth cleaning - waiting period	Orthodontics for adults	Orthodontics for children
10	0.74	0.68	0.57	0.67	0.63	0.51	0.66	0.73	0.73	0.54	0	0.43
11	0.7	0.52	0.53	0.64	0.47	0.56	0.5	0.71	0.7	0.69	0	0.66
12	0.58	0.63	0.76	0.45	0.59	0.65	0.76	0.66	0.58	0.43	0	0.54
13	0.48	0.53	0.45	0.72	0.59	0.51	0.45	0.49	0.61	0.65	0	0.65
14	0.53	0.5	0.56	0.79	0.72	0.72	0.64	0.51	0.61	0.54	0	0.79
15	0.61	0.71	0.71	0.45	0.58	0.63	0.59	0.65	0.74	0.74	0	0.66
16	0.67	0.46	0.53	0.71	0.75	0.72	0.72	0.66	0.68	0.63	0	0.69
17	0.49	0.72	0.75	0.69	0.75	0.5	0.57	0.78	0.61	0.67	0	0.65
18	0.77	0.78	0.7	0.58	0.68	0.62	0.72	0.56	0.66	0.8	0.78	0
19	0.44	0.78	0.55	0.76	0.52	0.63	0.62	0.67	0.66	0.57	0.5	0
20	0.54	0.71	0.7	0.74	0.42	0.59	0.48	0.45	0.65	0.41	0.57	0
21	0.45	0.53	0.45	0.42	0.43	0.71	0.5	0.6	0.42	0.76	0.68	0
22	0.74	0.72	0.58	0.66	0.69	0.76	0.64	0.7	0.74	0.65	0.69	0
23	0.59	0.63	0.63	0.65	0.77	0.46	0.73	0.71	0.4	0.46	0.44	0
24	0.76	0.44	0.49	0.62	0.67	0.6	0.75	0.61	0.75	0.68	0.47	0
25	0.53	0.67	0.51	0.47	0.63	0.76	0.61	0.54	0.51	0.8	0.46	0
26	0.45	0.64	0.76	0.58	0.67	0.54	0.45	0.78	0.74	0.65	0.67	0
27	0.74	0.7	0.41	0.51	0.73	0.65	0.41	0.6	0.51	0.75	0.78	0
28	0.45	0.64	0.57	0.65	0.58	0.43	0.51	0.77	0.62	0.45	0.54	0
29	0.48	0.68	0.68	0.5	0.71	0.7	0.58	0.47	0.56	0.75	0.41	0

Figure 3: Age related Risk Matrix

Depending on the user's answers, the particular features are given importance. We do so by multiplying matrix A with vector X. To include the age related risks in our final weight, we add the matrix B to the final result. We get a final vector from the below equation.

$$AX + B \tag{1}$$

The 'Preferences' can cover multiple features in the plan. To normalize the 'Preferences', we use the softmax function. We call this normalized preference vector as P.

In order to calculate the final weight for the TOPSIS, we give 80% weight-age to AX+B and 20% weight-age to the normalized preferences vector P and add them according to below equation.

$$W = 0.8 * (AX + B) + 0.2 * P \tag{2}$$

After we have calculated the final weight vector for the TOPSIS method, we multiply the plan matrix with the features with weight vector W we calculated above in equation 2. Plan matrix is the preprocessed matrix of numeric values of each plan from the original plans. For example, 70% coverage for different features is converted into 0.7 and so on. If we denote the Plan matrix as 'Plan', then we get the below equation 3 for the final vector. Plan matrix can be seen in Figure 4.

$$Plan\_final = Plan * W \tag{3}$$

PlanID	Company	Name	dentures - Crowns and bridges	dentures implants	dentures inlays	dentures veneers	treatment Plastic fillings	treatment - Root treatment (when health insurance pays share)	treatment - Periodontal treatment (if health insurance pays part)	treatment fissure sealing	Professional tooth cleaning	Professional tooth cleaning -waiting period	Orthodontics for adults	Orthodontics for children
P374	Continentale	CEZP-U	0.14	0.14	0.14	0.15	0.15	0	0.16	0	0.08	0.1	0	0
P751	AXA	DENT Komfort	0.13	0.13	0.12	0.13	0.11	0.12	0.12	0.12	0.1	0.04	0.18	0.16
P516	AXA	DENT Premiur	0.15	0.15	0.14	0.15	0.15	0.16	0.16	0.16	0.12	0.04	0.22	0.19
P502	ARAG	Dent70	0.12	0.12	0.11	0.12	0.11	0.12	0.12	0.12	0.07	0.16	0	0.14
P260	ARAG	Dent90	0.14	0.14	0.14	0.15	0.15	0.16	0.16	0.16	0.15	0.16	0.22	0.17
P438	ARAG	Dent90 +	0.15	0.15	0.15	0.15	0.15	0.16	0.16	0.16	0.15	0.16	0.22	0.19
P632	Janitos	Dental Max	0.15	0.15	0.14	0.15	0.15	0.16	0.16	0.16	0.14	0.16	0.25	0.17
P070	Janitos	Dental Plus	0.14	0.14	0.14	0.15	0.13	0	0.16	0	0.09	0.1	0	0.17
P582	Allianz	DentalBest (D	0.15	0.15	0.15	0.15	0.15	0.16	0.09	0.16	0.12	0.16	0.22	0.19
P878	Allianz	DentalPlus (D	0.13	0.13	0.12	0.13	0.11	0.12	0.06	0.12	0.1	0.16	0.18	0.16
P655	Württembergis	DUOSchutz D2	0.12	0.12	0.11	0.12	0.15	0.16	0.16	0.16	0.29	0.16	0.25	0.21
P572	HanseMerkur	EZK	0.15	0.15	0.15	0.15	0.15	0.16	0	0.16	0.12	0.16	0	0
P805	HanseMerkur	EZL	0.17	0.17	0.16	0.17	0.15	0.16	0	0.16	0.12	0.16	0	0
P059	Hallesche	GIGA.Dent - d	0.16	0.16	0.16	0.16	0.15	0.16	0.16	0.16	0.15	0.16	0.25	0.21
P228	DKV	KDT70 + KDBS	0.12	0.12	0.11	0.12	0.11	0.12	0.12	0.12	0.07	0.16	0	0
P984	DKV	KDT85 + KDBS	0.14	0.14	0.14	0.15	0.13	0.12	0.12	0.12	0.07	0.16	0	0
P563	DKV	KDT85+KDBE	0.14	0.14	0.14	0.15	0.13	0.16	0.16	0.16	0.14	0.16	0	0.21
P830	DKV	KDTP100+KDB	0.17	0.17	0.16	0.17	0.15	0.12	0.12	0.12	0.07	0.16	0	0
P771	Gothaer	MediProphy +	0.09	0.09	0.09	0.09	0.15	0.16	0.13	0.16	0.1	0.16	0	0

Figure 4: Numerical Plan Matrix

As per the TOPSIS algorithm described in section 2.3, we calculate the ideal positive value(Plan_final_max) and ideal negative value (Plan_final_min) value from the final Plan vector Plan_final. We calculate the Euclidean distance of each plan from the (Plan_final_max) and (Plan_final_min) and call them as Dist_max and Dist_min respectively. We calculate the final performance score as per the below equation.

$$PerformanceScore = (Dist\_min)/(Dist\_min + Dist\_max) \tag{4}$$

We then select the top 10 best plans based on the decreasing order of performance score value and recommend to the user.

Selection of the decision support process

In order to create a holistic system we developed a system architecture(Figure 5) first that covers all necessary modules of the application. We then used this architecture as a template to implement the Recommender System as a standalone application. The application itself and the flow of the process for the user are described in the following Chapter.

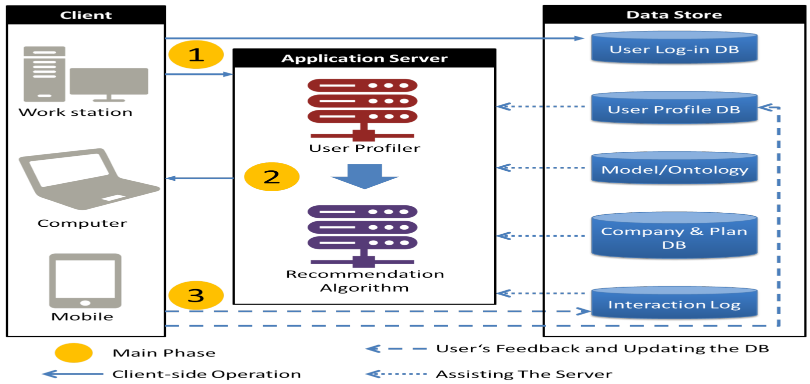

Figure 5: Exemplary Architecture

4 Implementation

4.1 Java AWT with MongoDB

Aiming at the web application, we first built a prototype in JAVA. The process about how to come up a customized recommendation is realized as an implementation of Java AWT step by step. Java AWT is an API to develop GUI or window-based applications in JAVA.

The medical data and preference from the user is collected as an input. After the process as described in section 3.2, a list of Top 10 best insurance plans is recommended to the user. There is also a possibility that the user clicks on a 'Detail' button to have a general view of the plan. The questionnaire and the results can be seen in Figure 7 and 8.

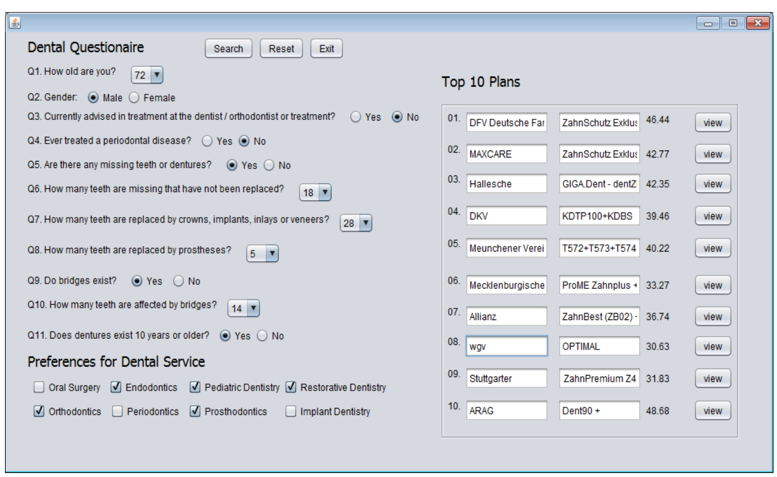

Figure 6: JAVA AWT Questionnaire

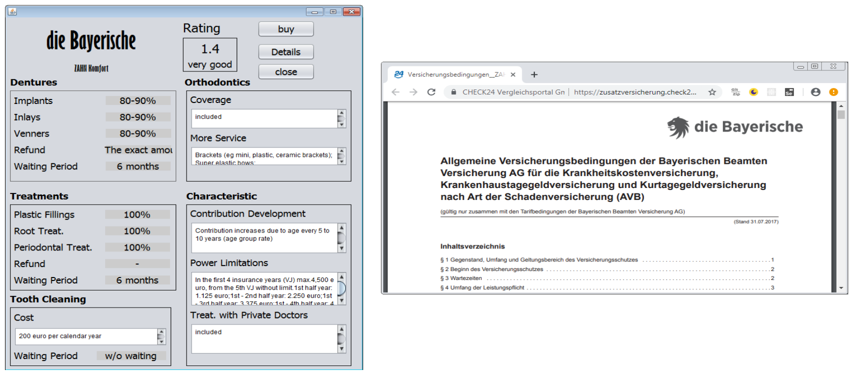

Figure 7: JAVA AWT Results

For the database, there are many options when it comes to data storage, such as SQL and NoSQL in perspective of schemata. For our project, we use MongoDB, which is one of the most popular NoSQL database to store the plans we acquired from insurance websites.

Many advantages can be provided by applying NoSQL database, such as the ability to deal with large-scale data storage, the efficiency in CRUD operations, and the flexibility to expand and stretch the data structure. Regarding the possible usages in the future, NoSQL database will be of great help for our project.

Furthermore, the database will consist of multiple smaller databases. The customer database is with basic personal information, such as name, age, sex, and job. An insurance database consists of insurance plans. And the customer database is extended to feature additional databases: activity, health, and property.

These databases will complete the information about the customer and can be used to derive possible risks that the customer may have. For example, if a customer is recorded to own a house in the property database, then our system may derive a risk of the house burning down. By dividing the customer data across multiple databases we also achieve simpler database tables, so as to avoid redundancy and losing information.

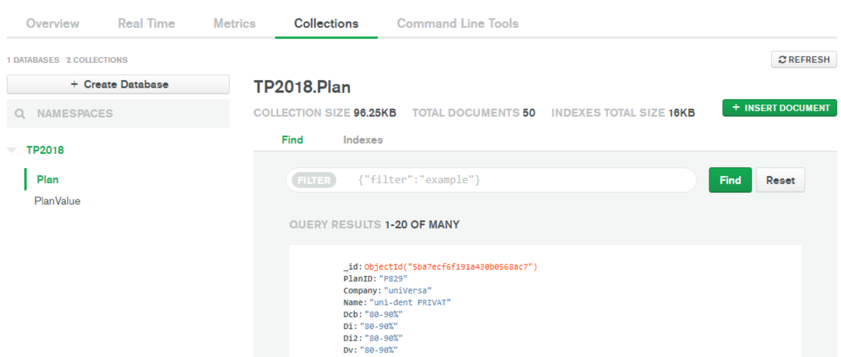

Figure 8: MongoDB Atlas - Cloud Storage

4.2 AI-based Insurance Broker Web Application

4.2.1 React Frontend

To make the project more realistic, we have developed a web application for Insurances in React and Flask. React is a popular JavaScript library for developing user interfaces.[3] We chose React to build the frontend of the insurance recommendation app, because of its simple and elegant UI design. Based on Material-UI, a framework that implements Google's Material Interface in React components, we created a simple interactive web app, that supports user's registration and login. Once a user has signed in the app, the option to select an insurance category, such as Health or Mobile Phone Insurance, will be enabled (see 9, 10). For the demo purposes of this application, only the Health Insurance category will be activated for the specific case of dental insurance.

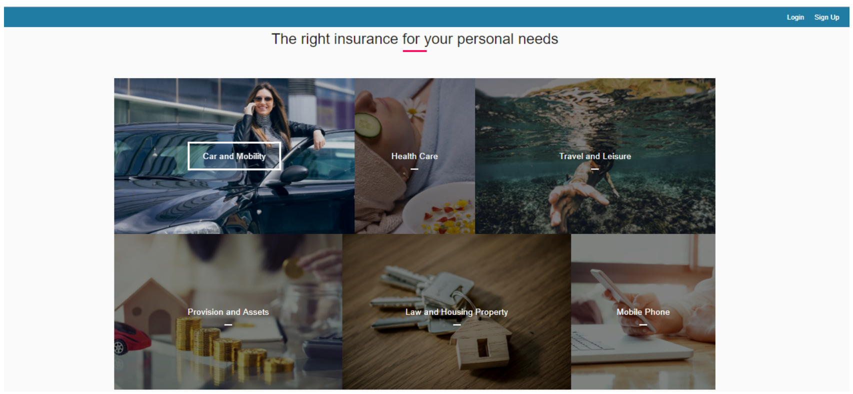

Figure 9: Landing Page

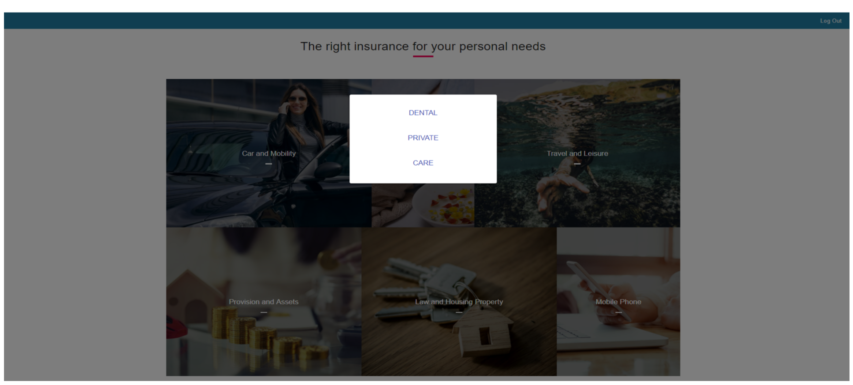

Figure 10: Health Insurance - Subcategories

[3]https://reactjs.org/

Once the user selects the desired insurance, the user will be redirected to a new page to fill out a short interactive questionnaire, which is relevant for the recommending dental insurances (see 11).

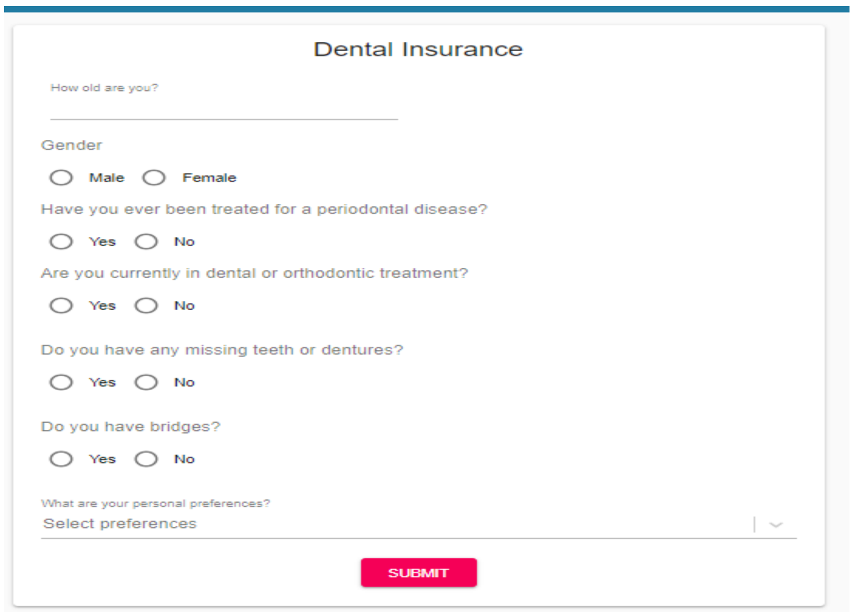

Figure 11: Question Form

At the end of the questionnaire the user is offered the possibility to express their preferences regarding specific plan traits (e.g., teeth cleaning, braces or waiting period) via a compact dropdown box for multiselection (see 12).

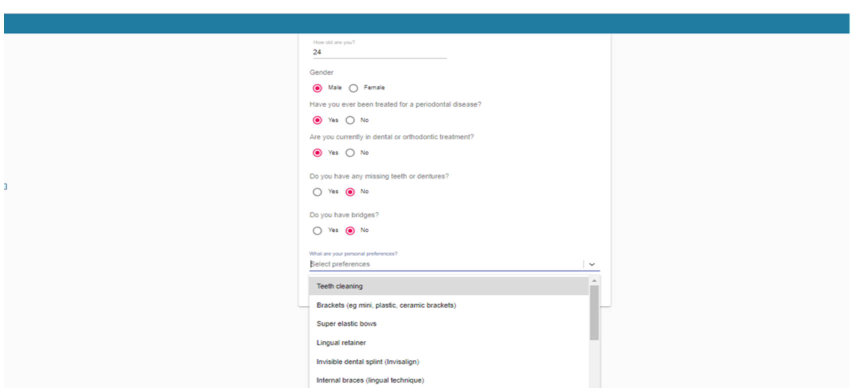

Figure 12: Selection of Preferences

After the user submits their answers, the recommendation results for the Top 10 dental insurances based on the user's preferences are displayed on a new page as a compact list, containing the company and insurance name, the plan price and score (see 13).

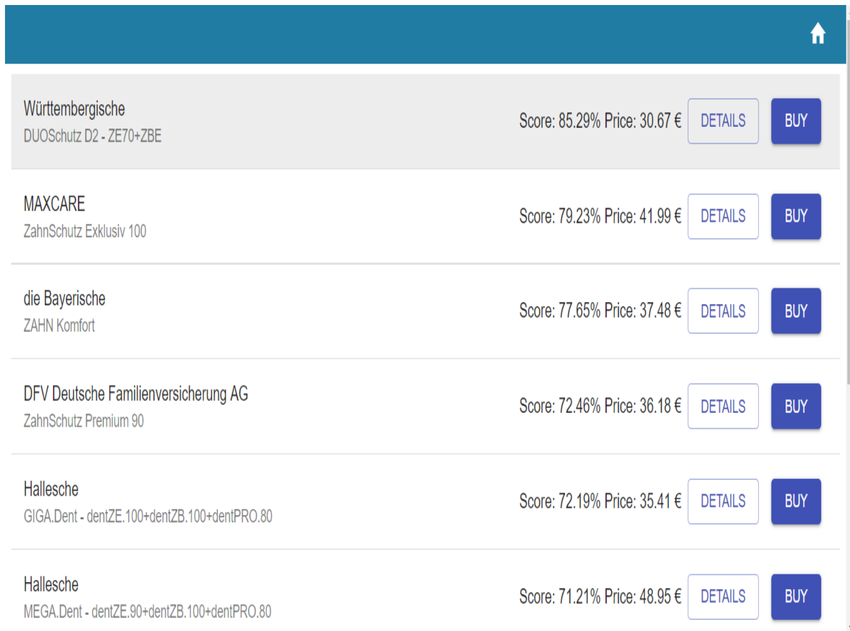

Figure 13: Results

For information regarding the plan's features, the user can click on the details button, where a table with the coverage data and policy details is displayed. There is link to the PDF for a detailed information about the plan. If a user wants to buy a specific plan, there is a "Buy" button that would trigger the purchase process in a real application. The design of our AI-based insurance broker application is very generic, which makes the adaptation of the UI to other insurance categories easy.

4.2.2 Flask Backend

A web framework is a code library that makes web development faster and easier by providing common patterns for building reliable, scalable and maintainable web applications. After the early 2000s, professional web development projects always use an existing web framework except in very unusual situations.Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. There are many Python web frameworks like Django, Pyramid, Flask, Web2Py etc.[4] For our project, we have used Flask.

[4] https://hackernoon.com/top-10-python-web-frameworks-to-learn-in-2018-b2ebab969d1a

Flask is a micro web framework written in Python.[5] It is classified as a microframework because it does not require particular tools or libraries (except for some basics standard libraries such as bottom.py). It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.Flask is more open-ended to unique systems. A developer can create a backend system just about however they want.
The TOPSIS algorithm for the project is also implemented in the backend in Python. It interacts with React components in the frontend which runs on a different server.

5 Output and Evaluation

Typically, there does not exist a unique optimal solution for recommendation problems and empirical evaluation measures only show how accurate the system is on items the user decided to rate. [Adomavicius and Tuzhilin] But especially in the insurance domain it is hard to say if a policy is a very good fit as the value of a policy can often only be assessed by a user who tried it out. These circumstances make it difficult to ask the customer for an evaluation and can also distort the actual quality of a policy. Similar cases can be seen on Check24 where users rate individual policies with stars. If many users are satisfied with the price they often give 5 stars but only a few users who actually have to use the service of an insurance company evaluate the insurance again afterwards. Therefore, a rating of users only makes sense if they rate the quality of a policy after they have made a claim.

Another way could be to ask insurance experts how they rate the quality of the recommendations, as they often have many years of experience. Nevertheless, these evaluations are as well subjective. It is therefore not easy to create numerical metrics that allow for a replicable baseline.

One way to objectively measure the quality of the complete recommendation system would be to ask for ratings of the whole recommendation system itself. Possible metrics could be satisfaction with process flow, customer service, and perceived trust level of the service. These values can then be related to objective measures such as business success and accomplishment of the mission, usually encompassed in the company's top line metric, like number of weekly number of visits, for example. With this approach it could be shown that the users are satisfied with the system also in the long term and that the success is proven by company key figures.

6 Conclusion and Limitations

AI has the potential to fundamentally reshape the role of an Insurance Broker. While many trivial tasks of a insurance brokers have been automated already, the core competence of

[5] http://flask.pocoo.org/

recommending an insurance personally suitable to the client remains missing. We tried to build an insurance recommender system that will model this missing behavior. We reviewed the literature for suitable recommendation algorithms and proposed a dental insurance recommender system based on TOPSIS. We extended the classical weighting scheme of TOPSIS by our own designed variant that incorporates both user preferences, objective plan data, as well as statistics from the medical domain. This way we want to simulate a holistic consulting approach in order to be able to give an ideal recommendation. In addition, we have designed an exemplary software architecture that can serve as a template for a complete AI-based Insurance Broker. We have embedded the recommender system in Java as well as in a web application.

Although our approach follows comprehensible mathematical steps, it is difficult to evaluate the results with clear comparable metrics. There are many unknowns in our system that need to be assessed more accurately by experts from the insurance field beforehand. The quality of a single insurance policy can be described by clearly defined numbers, but an ideal recommendation still depends on the customer's personal risks and needs. Therefore, our model also includes human misjudgments that cannot be completely eliminated. An alternative evaluation of our results must therefore be developed to prove the long-term success of our method.

References

[Adomavicius and Kwon2007] Gediminas Adomavicius and Youngok Kwon. 2007. for Multicriteria. *IEEE Intelligent Systems*.

[Adomavicius and Kwon2015] Gediminas Adomavicius and Youngok Kwon. 2015. Multi-criteria recommender systems. In *Recommender Systems Handbook, Second Edition*, pages 847–880.

[Adomavicius and Tuzhilin] Gediminas Adomavicius and Alexander Tuzhilin. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. Technical report.

[Burke] Robin Burke. Hybrid Recommender Systems: Survey and Experiments †. Technical report.

[Gandhi et al.2015] Kajal Gandhi, Aleš Popovič, Ray Hackney, Pedro Simões Coelho, and Jurij Jaklič. 2015. An Analysis of Multi-Criteria Decision Making Methods. *Expert Systems with Applications*, 5(1):49–58.

[Hinduja and Pandey2017] Akshay Hinduja and Manju Pandey. 2017. Multicriteria Recommender System for Life Insurance Plans based on Utility Theory. *Indian Journal of Science and Technology*, 10(14):1–8.

[Qin et al.] X S Qin, G H Huang, A Chakma, X H Nie, and Q G Lin. A MCDM-based expert system for climate-change impact assessment and adaptation planning-A case study for the Georgia Basin, Canada.

[Roy1996] Bernard Roy. 1996. *Multicriteria methodology for decision aiding*. Nonconvex optimization and its applications; 12. Kluwer Acad. Publ., Dordrecht [u.a.].

[Zavadskas et al.2016] Edmundas Kazimieras Zavadskas, Abbas Mardani, Ahmad Jusoh, Zenonas Turskis, and Khalil MD Nor. 2016. *Development of TOPSIS Method to Solve Complicated Decision-Making Problems — An Overview on Developments from 2000 to 2015*, volume 15.