ECM3401 Sample Lit Review

Literature Review and Project Specification

How effectively can multiple machine learning techniques recommend outfits?

[redacted]

Abstract

Machine learning (ML) is a branch of artificial intelligence (AI) that programs a system to learn and evolve without being explicitly programmed. Over the past two decades research has surged in the field, leading to many real-world applications, including; medical diagnosis [76], speech recognition [84], image recognition [46], video games [85] and personalized recommendations [79]. More specifically, with the development of e-commerce markets, novel recommendation technologies are becoming essential. Applications include; books [86], movies [3] and music [51]. However, research is sparse for outfit recommendation due to the complexity of the field. This paper aims to answer the research question: "How effectively can multiple machine learning techniques recommend outfits?". I will propose 3 recommender systems; a Short-Term Memory Content-Based Neural Network; a Demographic Decision Tree and a Utility-Based Bayesian Network. Each system should be capable of recommending aesthetically matching outfits to the active user.

ok, clear but perhaps too informative

I certify that all material in this dissertation which is not my own work has been identified

1. Introduction

Machine learning (ML) is a branch of artificial intelligence (AI) that programs a system to learn and evolve without being explicitly programmed. Machine learning techniques such as neural networks and decision trees have been a staple for learning models. Due to their flexibility, they have been continuously researched since the 1970s. In the 1990s machine learning shifted from a knowledge-driven approach to a data driven approach. Leading to a surge of research in analysing large datasets to draw conclusions, in turn evolving recommender systems. Recommender systems rely on data from their user base to make recommendations to new user's. One of the most popular recommender systems, collaborative filtering, is a data-driven approach that makes recommendations for a user based on other user's collective preference [1].

cite

?

well-known?

Companies notorious for their recommender systems such as Amazon [2], and Netflix [3], have created a demand in the e-commerce industry. Outfit selection has proven to be a complex problem however. The complexity stems from the relationship between humans and aesthetic items. Different religions, societies, and cultural practices adhere to different dress code. One must also consider what counts as an aesthetic pleasing outfit to each individual; e.g. an individual may wear outfits that are not deemed 'fashionable'. Together, these reasons are why research in outfit recommender systems is sparse.

briefly describe this here

Initially, this paper will present an overview of current ML techniques. There are two forms of ML; supervised and unsupervised. The proposed project is a supervised task, but for the readers benefit I have included both forms in my overview. It aims to show the reader the various varieties of each technique. Most notably, neural networks have been hugely popularized as there are many different models, meaning a host of applications have been researched [4]. The reader will then be presented with a high-level overview of current recommender systems and their applications. Most existing recommendation systems use collaborative-based or content-based approaches. Content-based approaches differ from collaborative-based by using data about the item itself, rather than the preference history of other users. For other approaches, research is sparse due to the strength of the aforementioned two. This paper will compare each recommender system, along with each ML technique, to justify my reasoning for my 3 proposed systems: A Short-Term Memory Content-Based Neural Network; A Demographic Decision Tree; A Utility-Based Bayesian Network.

why?

My specification will detail how I plan to develop each system. I have a range of ML techniques and recommender approaches to accurately compare performances. For this project, two datasets have been chosen, the first will be used on all systems, the second will be used if time permits. Both datasets contain a variety of outfits that are tagged with garments and accessories. One of the difficulties in outfit recommendation is the subjective nature of clothing. Therefore, not only will I test my systems against the validation data, I will also be conducting tests using active participants to see if they are pleased with the recommendations.

Overall, I see this to be an exciting project to explore new areas of ML. Many of the recommender systems and ML techniques have shown to be viable in a multitude of applications. I hope this paper will provide the reader with a comprehensive overview of ML techniques, recommender systems, and more specifically, outfit recommendations.

good intro

2. Literature Review

This literature review aims to give a comprehensive overview of four of the main ML techniques; neural networks (NNs), Bayesian networks (BNs), decision trees (DTs) and collaborative filtering (CF). These techniques have been chosen because of their proven applicability in recommender system. Following this, the various categories of recommender system will be discussed. Finished with all current research in outfit recommender systems.

good justification

2.1. Overview of Machine Learning

In this overview, each section will contain both unsupervised and supervised ML techniques. The rest of the paper will take a focus on supervised techniques.

cite

2.1.1. Neural Networks

Neural networks are learning systems that are loosely modelled after the neural structure of the human brain. They are formed of layers that transfer values to each other through nodes. Systems differ by either the function they apply to the value, or the architecture of the model. cite

Feed-forward neural networks (FFNNs) were the first and simplest type of artificial neural network (ANN), where data can only run in a

Very Comprehensive Lit Rev. It could have been clearer if more examples were provided.

singular direction. There are two types of FFNN; single-layer [5], which contains just the input and output layer and multi-layer [6], that contains additional, hidden-layers. Hidden-layers apply a function to the previous layer to transform inputs into a unique value, allowing multi-layer FFNN to solve more complex problems, such as a notoriously challenging game, Go [7]. *Convolutional neural networks* (CNNs) express data as a "map", where the proximity between the two data points indicates how related they are. *Deconvolutional networks* (DNs) [8] are used to visualize the information learnt by a CNN. *Deep convolutional inverse graphics networks* (DCIGN) are similar to DNs, but improve on the technique by semantically splitting neurons into groups that distinctly represent a specific transformation [9].

Recurrent Neural Networks (RNNs) [10] are fed values from themselves from the previous pass, as well as the previous layer. *Long/short term memory* (LTSM) networks [11] and *echo state networks* (ESN) [12] are types of RNN that attempt to solve the vanishing gradient problem. LTSMs preserves the back-propagated ~~gated~~ error, whilst ESNs observe the output values over time, to update weights within the systems. *Markov chain* (MC) networks [13] are subtype of RNNs that are memoryless, the state you're in therefore depends on the previous state. MC is the predecessor to *Hopfield networks* [14] and *Boltzmann machines* [15] that are designed to solve combinatoric problems. BMs are an extension of HNs, increasing the number of layers from one to two.

could say a little how back prop works

Extreme learning machines [16] are very similar to FFNNs, however have random connections. *Kohonen networks* [17] are a self-organising map that is used for visualization and analysis of high-dimensional data. *Neural Turing machines* [18] combine the efficiency and permanency of regular digital storage with the expressive power of neural networks.

2.1.2. Decision Trees

Decision trees (DTs) can be used to explicitly represent decisions and the decision-making process. They create a model to perform classifications by splitting variables based on their attributes. Credit to Wei-Yin Loh for his dedication to the advancement of DTs [23].

Iterative Dichotomiser 3 (ID3) is one of the best-known DT algorithms and is based on Occam's razor [19]. ID3 generates a decision tree by employing a top-down, greedy search, to test each attribute at every node [20]. C4.5 [21] is an extension of ID3 that generates a decision tree by recursively splitting the data. Then evaluates all possible tests and selects the best information gain [22]. Classification and Regression Trees (CART) are represented by a binary tree and follows a greedy, depth-first approach. It grows a large tree and then prunes the tree based on the cross-validation estimate of error [23][24].

Chi-squared Automatic Interaction Detectors (CHAID) [25] uses the Holm-Bonferroni [26] method to evaluate multiple comparisons on a non-binary tree. *Fact and Accurate Classification Trees* (FACT) [23] generate linear splits, that split each node into a number of children nodes, based on the number of classes. FACT is inherently biased towards categorical data. To overcome this, *Quick, Unbiased and Efficient Statistical Tree* (QUEST) [27] uses contingency table chi-squared tests on categorical variables [23]. *Classification Rule and Unbiased Interaction Selection and Estimate* (CRUISE) is a descendent of QUEST that uses chi-squared tests for all variables [23]. Cruise can have too many interactions tests, resulting in negative effects on main tests. *Generalized, Unbiased, Interaction Detection and Estimation* (GUIDE) [28] restricts the frequency of test interactions [23].

2.1.3. Bayesian Networks

Bayesian networks (BNs) calculate the probability of an uncertain cause given some observed data [29]. They create probabilistic models that represents a set of variables and their conditionals dependencies.

Naïve Bayes (NB) is the simplest form of BN classifiers. It takes advantage of Bayes theorem by calculating the conditional probability that an event will happen, given another event that has already occurred [30]. *Selective NB* uses 3 approaches; a filter approach for feature selection; a wrapper approach that asses each subset using the classifier performance, and finally, an embedded approach that selects features using the information obtained from training a classifier [31]. *Semi-NB* has been extended from NB to detect the dependencies between attributes. It optimises the trade-off between the 'non-naivety' and the reliability of approximations of probabilities by introducing a new attribute set [32]. *Tree-augmented NB* (TAN) maintains the structure of NB and augments it by adding edges between the variables in order to capture for correlations between attributes [31]. *General Bayesian Networks* (GBNs) are an unrestricted BN, which treats the class node as an ordinary node (e.g. the class node can also be a child of some attribute node) [33]. GBNs are a special directed graphical model with conditional Gaussian distributions. Each variable is defined by a Gaussian marginal or conditional distribution and variables are linearly related to their parents [34].

2.1.4. Collaborative Filtering

define

Collaborative filtering is a recommender system; however, it is also one of the most popular machine learning techniques. It has therefore been discussed here but in future sections it will be discussed under the Recommender Systems subcategory.

Collaborative filtering (CF) makes recommendations about the preferences of a user based on other user's collective taste information [35]. There are two types of CF, *memory-based* and *model-based.*

Memory-based CF uses the entire user-item database to find users that are similar to the active user. *Neighbourhood-based CF* relies on users displaying similar patterns of rating behaviour and similar items receiving similar ratings [36]. There are two types of neighbourhood-based CF, *user-based* and *item-based.* User-based CF uses ratings from a collection of users as a basis to make recommendations for the active user [37]. Item-Based CF, was popularized by Amazon [2]. Instead of recommendations being made based on neighbouring users, they are based on the history of ratings from the active user. Both user-based [38] and item-based [39] have a top-N version of the algorithm. User-based Top-~~N CF~~ uses Pearson's correlation [40] to identify a set of items the user has not 'purchased'. Item-Based Top-N CF addresses the scalability problem of user-based top-N. Item-based top-N removes the union of the items the active user has already purchased from the list, and then calculates the similarities between recommended items and the items the active user has not bought [36].

what is this?

Model-Based CF has been investigated to solve the shortcomings of memory-based CF algorithms [41]. They design a model that allows the systems to learn to recognize complex patterns, and then make intelligent predictions from real-world data [36]. Clustering CF group users based on their preferences to reduce the sparsity and improve the scalability [42]. Markov Decision Based (MDP)-based CF [43] view recommendations as a sequential optimization problem that consider the long-term effects of each recommendation [44]. Latent-Semantic CF introduces latent class variables in a mixture model setting to discover communities and prototypical interest profiles [36][45].

What are these shortcomings

2.1.5. Datasets

This section will detail the datasets used in current research on outfit recommender systems. McAuley et al. [46] and Veit et al. [47] adapted data from the Amazon web store that contains over 6 million fashion related objects. Liu et al. create a dataset that contains 7 multi-value clothing attributes and 10 occasion categories via Amazon Mechanic Turk [48]. Hu et al. uses a data from Polyvore, an online fashion forum [49], to create a dataset with 80,000 images split into 3 subcategories. Yu et al. used a dataset of clothing items from a video game, The Sims [50]. Radio Frequency Identification RFID technology has also been used to create datasets of the user's own clothes [1][35].

2.2. Recommender Systems

~~Recommender systems~~ this section aims to give the reader a high-level overview of the five main types of recommender systems. Each section will contain a description and applications of the subgroup.

2.2.1 Collaborative Recommender System

Collaborative recommender systems have been discussed in section 2.1.4 as it is one of the most popular and widely implemented ML techniques. For this reason, only its applications will be discussed. Lee et al. [51] used CF for music recommendations. Using a new implicit rating, mobile web usage as a new implicit rating, their system worked better than existing CF-based recommender systems. Zaiane combined the users purchasing history with the history of other users that bought similar items to make e-learning recommendations [52]. Yu et al. recommended television shows by merging the profiles of user together, to create a richer pool of information.

2.2.2 Content-Based Recommender System

Content-based Recommender Systems define objects by their associated features. It then recommends objects to a user based on what objects/features that user has had interactions with in the past. Wintrode et al extracted a wide variety of features to characterize non-linguistic aspects of audio. Then made recommendations on similar documents using a ranked, score fusion [53].

Rohani et al. utilises the interests and preferences of the user's friends and faculty to suggest the most relevant items to users. Resulting in a 4% improvement of users that were satisfied with the recommendations [54]. Ferdous & Ali uses a semantic matching factor that indicates normalized actual similarity between the matching documents [56]. Ghauth & Abdullah incorporates a peer learning rating to recommend learning materials. Their system was 12.6% more accurate than a normal content-based approach [57].

2.2.3 Demographic-Based Recommender System

Demographic based Recommender Systems categorize users based on their attributes, then proceed to make recommendations based on their demographic classes. such as? Krulwich used demographic groups from marketing research to suggest a range of products and services [58]. Zhu et al. used a neural network (NN) for online survey recommendations. Users were split into classes dependent on how often they took part in online surveys. It was found sending surveys to participants who were more willing to respond greatly increased the success rate, and made the replies of a higher quality [59]. Wang et al. uses a demographic approach for tourist attractions. They represent the demographic information of tourists as a vector, which is used to classify the tourists into separate demographic classes [60].

2.2.4 Utility-Based & Knowledge-Based Recommender System

Utility based Recommender Systems (UBRSs) make recommendations based on an evaluation of the match between a user's need and the set of options available [58]. example? UBRS functions by ranking objects based on the multi-attribute utility theory (MUAT) value. However, due to the difficult in creating a utility function for each user, there is little research in this area. Liang et al. categorized utility into two different types, objective and subjective. This allowed users to adjust their 'subjective' value, without interfering with the objective value (object information) [61].

Knowledge-based recommender systems differ by having a functional knowledge about how an item meets some user's particular needs [58]. For example, Google uses links between web pages to represent knowledge. Towle and Quinn found combining two models (user and product), gave a richer picture than either alone [61]. Martinez et al. defines a flexible framework using a fuzzy linguistic approach to capture the uncertainty of the user's preferences, giving the users more control to express their necessities [62].

2.3. Machine Learning in Outfit Recommendation

This section will discuss what research has been published specifically in outfit recommender systems and what makes their respected system unique. A note to the reader, this section will not be discussed in critical analysis as for the most part they do not relate to my specific task: Recommending an outfit to a user given a selected number of clothing items.

Justifica not complet clear

Gulla and Litlehamar's collaborative filtering approach [35] groups users based on the similarity of clothes in their wardrobe, then uses an improved utility matrix that incorporates how many users have favourited each outfit. Resulting in capturing the co-occurrence interaction between two clothing items.

McAuley et al. captured the largest dataset possible, and developed a scalable method for uncovering human notions of visual clothing relationships. Combined with their content-based approach, their method could model a variety of visual relationships beyond simple visual similarity [46]. Liu et al. uses a latent Support Vector Machine (SVM) approach. Given a photo of the user, combined with the occasion, they build a unified framework of 'matching rules' for outfit recommendation [48]. Yu-Chu et al.'s Bayesian network system utilizes user's feedback from the top when recommending the bottom. They combine the feedback with a unique probability function that ensures items are recommended with equal frequency [35]. Yu et al. uses a Tree Augmented Naïve Bayes approach to capture the relationships amongst clothing items. They use the Expectation-Maximization algorithm to train the system, as it maximises conditional mutual information between attributes [50].Hu et al. created a Tensor Factorization method that uses a gradient boosting based approach to learn the nonlinear functions that map the feature vectors of clothing items from the feature space to a latent space [49].

Multiple other hybrid techniques have been used. Kokol et al. used a vector decision tree (VDT) for multiple outputs, resulting in more accurate outputs. They optimized the system with a genetic algorithm that achieved an accuracy percentage of 85% against their validation data [63].

systematic and thorough critical analysis

Veit et al. used a Siamese Convolutional Neural Network framework that can recover a style space for clothing items from co-occurrence information as well as category labels. Combined with a robust nearest-neighbour retrieval method [47].

3. Critical Analysis

3.1. Machine Learning Techniques

This section gives a high-level overview of the advantages and disadvantages to each system. I then compare selected variants of each technique that are the most applicable to my project.

3.1.1 Neural Networks

A key advantage of neural networks (NNs) is their nonlinear model is very easy to use and understand. not robust? NNs models are non-parametric, which gives more choice in the chosen dataset [31] and system design. The infamous back-propagation (BP) training algorithm has been successfully applied to many tasks. However, even though BP convergence is guaranteed, it has been shown to be slow [64]. NNs black-box architecture also result in a less-flexible system, as you are generally given an answer with no indication of the process of reaching it.

Single-Layer FFNN vs Multi-Layer FFNN: Single layer FFNNs have an extremely fast computation time and have been shown to be enough layers to form a close enough approximation to any non-linear decision boundary [31]. Single-layer models have also shown that if they can fit the training examples almost as well as a multi-layer model, then the single-layer can be expected to give a better performance on unseen test data. However, single-layer FFNN; can only represent a limited set of functions; decision boundaries must be hyperplanes and can only perfectly separate linearly separable data [66].

Multi-Layer FFNNs can handle more complex problems and have been shown to be robust when interacting with many variables [66]. However, dependent on the number of hidden nodes and layers, they are very costly to train [65]. Multi-layer FFNNs commonly gets stuck in some undesired local optima, which prevents lower layers learning useful features. Finally, if ? the proposed dataset is small, then a greater difference in fit is required before multi-layer models are preferred.

3.1.2 Decision Trees

Decision trees (DTs) are an easy to understand subcategory of machine learning with extensive applications in classification tasks. An advantage to the usability of DTs is that you can easily graphically represent the tree with the explicit relationships between nodes. They have also been shown to create complex alternatives when additional data is added [67]. However, DTs struggle dealing with many complex interactions. Their 'divide and conquer' [19] approach means they only tend to perform well if there are few highly relevant attributes. The greedy characteristic of DTs also often leads to over-sensitivity to irrelevant attributes and noise [68].

ID3 vs C4.5 vs CART: ID3 has been shown to be short and compressive [22]. The readability of the prediction rules created from the training dating is hugely useful in debugging, interpretation and usability. However, ID3 can only handle nominal data, it's also not able to deal with noisy data [20]. C4.5 is an extension of ID3 that can handle continuous and discrete attributes. Another improvement on ID3 is C4.5 is robust in the presence of noise, it avoids over fitting and it can also handle missing attributes [69]. C4.5 will also prune itself at the end, to remove unhelpful branches and replaces them with leaf nodes, resulting in faster classifications. However, C4.5 often constructs empty branches, which lead to nodes having zero, or close to zero values. These values do not help the system and are redundant for the task, making the tree larger and more complex [20].

Classification and Regression Trees (CARTs) are easily able to handle both numerical and categorical variables. CART's results are usually invariant to monotone transformations of its independent variables [70]. CART has also been shown to easily handle outliers. However, CART can create unstable trees, leading to insignificant modifications resulting in radical changes [71]. CART also only splits by one variable, which slows down the classification process.

3.1.3 Bayesian Networks

Bayesian networks (BNs) readily facilitate use of prior knowledge by constructing 'causal' edges between two factors that are believed to be correlated. BNs have been shown to handle incomplete datasets and to provide efficient methods to prevent over-fitting [72]. However, BNs are extremely computationally expensive as they calculate the probability of all branches, which is an NP-hard task [73]. They also tend to perform poorly on high dimensional data, and can be hard to interpret.

Naïve Bayes vs Tan-Augmented Naïve Bayes (TAN): The main advantage of Naïve Bayes (NB) is that it will not only return the prediction, but also the degree of certainty. Compared to other BN techniques, NB is one of the faster methods to train and classify. It can also handle real and discrete data, and can be updated with new data efficiently. However, NB makes a very strong assumption on the shape of your data distribution, and can give bad ('naïve') results [30]. NB does not capture the correlations between the variables in the system.

An advantage of Tree-augmented NB (TAN) is having a maximum of two parents make the computational complexity of the model reduce greatly. Resulting in TAN maintaining the robustness of the NB model and at the same time, displaying better accuracy, more efficiently [31]. However, if the dataset does not have correlations between variables, NB is the best to use.

3.2. Recommender Systems

Collaborate Filtering (CF) algorithms are particularly useful in domains where the analysis of content is very expensive or difficult, like music and film suggestion. They can also hinder deficient recommendations by taking the precedence of users which are more relevant and active [74]. They have been shown to be better than content-based methods because of their focus on user history. However, they suffer from the 'cold start' problem, where the system cannot produce any recommendations until enough information is acquired. There is also the 'gray sheep' problem, where a user may not belong to a particular group. Thirdly, CF has shown to suffer from a scalability problem when the number of objects or users increase [75].

An advantage of content-based methods over CF is that it only need ratings from the active user. This helps alleviate the 'cold start' problem as far less data is needed. Content-based systems are also have a transparent functionality, each recommendation can be detailed by explicitly listing content features. They also do not suffer from the 'first-rater' problem, as they do not require a history of ratings from other users [76]. However, content-based approaches require users to detail their preferences before any recommendations can be made. Content-based methods also need a large amount of domain knowledge [77]. For example, a movie would need the actors, the genre etc. They also cannot make novel recommendations, as it relies on recommending similar items to those the user has rated [78].

Unlike collaborative-based and content-based methods, demographic-based recommender systems do not require a history of user ratings. However, demographic based systems require complete user information, which is impractical to collect and can cause ethical issues due to privacy. 'Demographic-based systems, however, are too general, assuming that all users in the same demographic profile have the same interests [79]. Therefore, this system does not usually provide the best recommendations alone, but lend themselves to bolster hybrid recommender systems, often improving efficiency and performance.

Utility-based and Knowledge-based based recommender systems do not suffer from sparsity problems, as they do not base their recommendations on accumulated evidence. Utility-based techniques require that the system build a complete utility function across all features of the objects under consideration [80]. Therefore, many non-product related features can be incorporated to provide heavily personalised recommendations. However, making a unique function for each user is a difficult task. A user must explicitly select a preference for each feature that describes an item of interest. Which becomes greatly unfeasible for complex and subjective domains such as movies.

Knowledge-based systems are useful for casual exploration, as it demands less of the user than utility-based systems. It does not experience cold-start problems as it should have all required information before beginning recommendations. These recommendations can also be as wide-ranging as its knowledge base allows [79]. On the other hand, it cannot discover unique niches to recommend to the user the way collaborative systems can. Lastly, it is very difficult to collect all the required data for both knowledge-based and utility-based approaches.

Related

3.3. Project Specification Critical Analysis

This section aims to provide the reader with my opinion on the revised literature with reference to my proposed project. why?

3.3.1 Machine Learning Techniques

Neural network: NN's accessibility, along with their non-parametric means it should create an efficient prototype. Back propagation has had a host of succession which make me confident it can efficiently train the attributes between clothing items. I would prefer to use a multi-layer FFNN as the hidden-layers have shown to greatly improve accuracy. However, the significantly larger dataset required cannot be labelled for this project. Despite this, dependent on the training data, single-layer FFNNs have shown to perform better than multi-layer FFNNs. Therefore, as long as the model I create fits the training dataset well, the system should still have good accuracy.

Decision Tree: As I am creating a model to perform recommendations I have decided to use C4.5 for its ability to prune itself, resulting in faster, and more accurate, predictions. My proposed dataset also has few highly relevant attributes, meeting DTs basic requirements. DTs are also useful for their ability to calculate information gain [81]. Which can be used to pick the attributes that are the most effective in splitting the data. e.g.?

Bayesian Network: A major advantage to Bayesian networks (BNs) is their querying ability. This ability lends itself to recommender systems as inherently all systems make recommendations based on a query. It also provides efficient methods to prevent the overfitting of data. Finally, the low dimensionality of the data I use plays into BNs strengths. I have proposed a naïve Bayes system at it is the only approach to return a degree of certainty, which greatly helps in recommender systems. However, a big drawback of naïve Bayes (NB) is the computational cost.

3.3.2 Recommender Systems

Collaborative Filtering Recommender System: CF systems ability to find relationships between the history of users and items, provide extremely fast and accurate recommendations. For my project it would be an exceptionally useful method as it has shown to work complex objects, such as film, without demanding any domain knowledge. However, due to its need for a history of ratings, it would prove difficult. Current available clothing datasets do not contain unique user preferences. Additionally, my project would fall into the cold-start problem and the gray-sheep problem.

Content-Based Recommender System: Content-based methods are largely outdated due to their inflexibility. However, they only require ratings from the active user. Therefore, they have been shown to be efficient in hybrid ML techniques [82]. In my project I can incorporate a weight value representing the user's opinion on each outfit recommendation.

Demographic-Based Recommender System: Demographic-based systems lend themselves extremely well to outfit selection, as fashion inherently forms 'groups' of people dependent on their style [83]. To correctly recommend outfits, a demographic-based system will require a user to explicitly state their 'fashion group', or characteristics that can place them into a fashion group.

Utility-Based & Knowledge-Based Recommender System: A utility-based approach is what every recommender system should strive for. Providing personalised recommendations utilising a function that changes dependent on the active user. Incorporating the multi-attribute utility theory (MUAT) into my system should provide intricate relationships between user's and clothing items. If I had more time, knowledge based would be my first choice as it would act as a 'personal stylist'.

4. Specification

could give overall description of the systems before explaining detail

Specification aims to detail the plan and method for my 3 proposed systems. → over architecture

4.1. Variables & Attributes

An outfit will consist of 4 variables: top1; top2; bottom; shoe. Top is split into two variables as it is common to wear multiple items on your top half (e.g. coat and shirt). The attributes for these variables are split into primary and secondary attributes. Primary attributes are ones that will be used in every system. Secondary attributes are either used in specific circumstances, or to evolve a system that is already working with the primary attributes. When evolving my systems, a decision tree will be used to collect the information gain and evaluate which attributes to include. These attributes are used in specific systems detailed below. The primary attributes will be style (e.g. shirt, blouse, jeans, high-heels), primary colour and secondary colour. Secondary variables will be demographic (section 3.3.2), dress code [50] and weather. These variables are either used in specific systems detailed below. If time permits they will be used to evolve the prototype as well. They will be hand-labelled external to negate any bias. train?

4.2. Machine Learning Techniques

The first system will be a neural network due to their accessibility. This system will work as a functioning prototype, allowing me to test various parameters, variables and datasets. It will be a single-layer FFNN that has 2 inputs (e.g. top & bottom) and 1 output (e.g. shoes). every 'info' is a vector of texture

4.2.1. Short-Term Memory Content-Based Neural Network

• This is overambitious, and you should focus on a system or two.
• There is perhaps too much detail and a overall picture of the envisioned system is missing

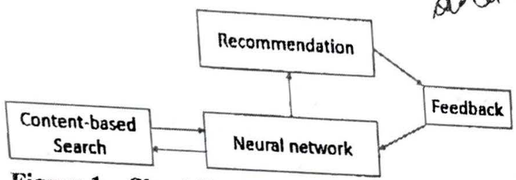

Figure 1 – Short-Term Memory Content-Based Neural Network Model

The second system will be the short-term memory content-based neural network, which is modelled in figure 1. It will follow the same design as the prototype in the sense that it will be a single-layer FFNN. However, it will allow the active user to rate the recommend item of clothing. This rating will be used as a weighting factor for the following recommendations. The system will have 3 inputs and 1 output. The additional input is for the users rating, it can take the input of 1 (like), 0 (dislike) and 0.5 (didn't choose). The hope is that the system will develop into recommending two items of clothing from one input, where the system will increase to 5 inputs. This is so the user can rate the outfit, as well as the individual items of clothing.

not entirely clear

4.2.2. Demographic Decision Tree

The demographic decision tree will use additional variables to recommend outfits. Two variables will be added to the dataset, demographic and dress code [50]. These are added because decision trees need a certain number of variables to function efficiently. Demographic will class each item of clothing into 10 different categories, related to the 'fashion group' that item belongs in. I have chosen 10 as DTs perform more efficiently with variables they can split into a good amount of sub categories. The C4.5 decision tree will be used for its ability to prune itself once it's finished training. As my system will be a model for future recommendations, it should make my system faster. C4.5 has also been shown to handle missing values, which will be tested using my secondary dataset, fashion 144k, detailed in section 4.3.

4.2.3. Utility-Based Bayesian Network

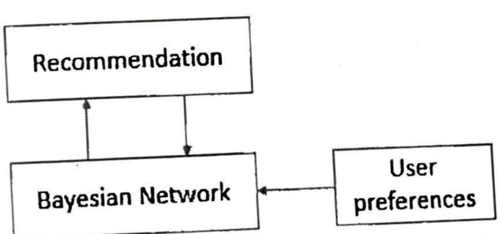

Figure 2 – Utility-Based Bayesian Network Model

The utility-based Bayesian network (UBN), shown in figure 2, will initially ask the user to select five photos from a selected range of outfits. The user will then be asked three questions about their preferences on various clothing items. These two features will be combined in a function f that is unique to each user. When the user uses the system, f will be applied to the Bayesian network query. I have chosen to use a Bayesian network as it is easy to apply functions to probability. They can also find relationships between factors which they believe are related, a useful tool when trying to find unique correlations for each user. Naïve Bayes will be used as it has been shown to be efficient in classification tasks. The added layer of information provided by f should solve the short coming of naïve Bayes.

4.3. Dataset

There are two datasets for this project, the primary dataset will be used for every technique, and if time is available, the secondary data will then be used to either train new systems or evolve current systems. ?

Primary Dataset: The Clothing Co-Parsing dataset contains 2098 street fashion images with a total of 59 tags. The CCP dataset provides a mix of genders, as well as outfit styles, accessories and garments. Initially I will engineer the features of the dataset by running a script to achieve the following; each image will have 4 tags, 1 for each variable category (top1, top2, bottom, shoe). All other attributes will have to be hand-labelled.

Secondary Dataset: Fashion144k contains 144,169 images from social media posts. Each image is tagged with the number of people that liked the outfit, general tags on the outfit (e.g. everyday, chic), colours and garments. The fashion144k dataset provides a rating for each outfit, which gives a measure of **certainty**. Also, the additional variables help in the splitting and classification of data. These two uses combined present a dataset that lends itself to recommender systems.

4.4. Language

Python will be used to create the systems due to the vast array of libraries available. The modules used will be TensorFlow, scikit-learn, pattern and matplotlib. To build the UI for testing purposes, javascript will be used.

4.5. Research Question

"How effectively can multiple machine learning techniques recommend outfits?". The research questions exposes a gap in ML research, with only a few outfit recommender systems being created for outfit selection. Currently, no available system can select an outfit using a given number of clothing items. The systems created are designed to evaluate why this problem is so complex and whether various novel recommendations systems can be modelled to correctly select outfits. I am confident that the short-term memory content-based neural network will produce a good model for recommendations due to their aforementioned wide variety of applications.

5. Evaluation Criteria

This section aims to present the reader with how I plan on evaluating each system. As this is a complex task, each proposed system should fair differently. In order to draw valid conclusions, the following guidelines will be met.

5.1. Design

this should be earlier

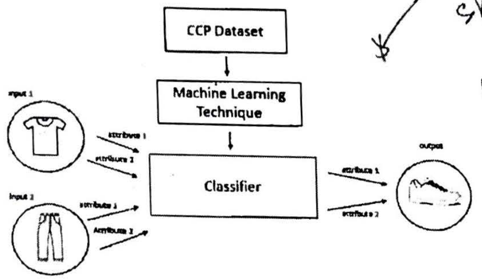

Figure 3 – Model of the Minimum Expectation for Each Technique

Figure 3 displays a diagram of the minimum requirement for the systems I have proposed to be considered 'working'. Each proposed system should be able to take in two inputs, each with two variables, and make a recommendation based on the training data from the CCP dataset.

5.2. Performance

Technical: To be considered a good system a model should achieve a 75% accuracy compared to the validation data in the CCP dataset. This is based on existing systems results [50][49][35]. I will also evaluate if there were any faults during training or recommendations.

ok good

Subjective: Outfit selection is an inherently subjective topic. Because of this the system will also be tested on active users. They will be allowed to use the recommendation system, through a UI, for as long as they want. Certain system detailed in section 4.2. will need longer than others to provide accurate recommendations for the user. Each recommendation will contain either 3 or 5 items, to allow for personal taste. After, the user will be asked to fill out a short survey, with the aim of 65% of the users finding the system correctly recommended outfits they like.

5.3. Analysis

I will ask the following questions of each systems to compare them:

1) How long does it take to train the system?
2) Were there any issues with the training?
3) Can they learn to make a good outfit selection?
4) How long does it take to make a recommendation?
5) How accurate are the recommendations compared to the validation data?
6) How the recommendations are perceived by active users?
7) Could the recommender system be improved in any way?

6. Conclusion

Machine learnings surge over the past two decades has provided an enriching array of applications. Neural networks usability and 'black-box' architecture provide an accessible model to create my first system. More specifically, single-layer FFNNs have been shown to perform efficiently in classification tasks. Decision trees have been shown to be extremely good in classification tasks for decades. However, a big drawback is they cannot handle the multi-dimensional data systems require today. Having said that, I believe using a demographic approach will provide my proposed DT with an effective variable to split the data by. Bayesian networks querying ability provide them with huge utility in recommender systems. Research has already used them in outfit recommendation [35], however their computational cost may be an issue with the time frame of this project.

fair comparison conclusion

Collaborative Recommender systems ability to make recommendations based on the interests of all users has made the technique dominate the market. User-based CF and item-based CF techniques take an intuitive focus on making personalised recommendations. However, research in content-based approaches is slowing down as problems are becoming too complex. I decided to use them as I believe combining the feedback of users with a content-based approach would lead to efficient outfit recommendation. A demographic approach also lends itself well to my project as outfits naturally splits into groups. Research is sparse for knowledge-based and utility-based recommender systems due to their systems requiring too much from the user. However, theoretically, utility-based approaches provide excellent personalized recommendations. I plan to incorporate it with the Bayesian networks probability value as it is easy to apply a function to. To be used in my proposed system, the two-proposed datasets have been meticulously selected as they contain a wide range of outfits with detailed tags. These tags however are what causes some of the biggest problems to researchers.

The multi-dimensional nature and subjectivity of outfit selection has resulted in a difficult problem. Current research described in section 2.3. provides intuitive solutions to building outfit recommendation systems. However, none provide an efficient way for the user to input a selected number of clothing items and be recommended the rest of the outfit. There are three systems I have proposed to fill this gap. A Short-Term Memory Content-Based Neural Network, a Demographic Decision Tree and a Utility-Based Bayesian Network. I have chosen these systems to give me as much of a range as possible when conducting my evaluations. In my evaluations I have ensured to address the issue of subjectivity by allowing active users to test each system. These test results, combined with the results from the validation testing, should give a comprehensive overview of how accurate each system is.

Following on from this report, I will initially create a functioning neural network prototype. The prototype will be able to take in two items of clothing, and output the third to complete the outfit. I will have to set aside time to hand label the CCP dataset. Secondly will be to learn what existing libraries exist in Python for machine learning. Frameworks such as TensorFlow have been created by Google specifically for machine learning This project looks to be an exciting exploration into a gap in both machine learning and recommender systems.

this explains next steps well

7. References

[1] - Yu-Chu, L., Kawakita, Y., Suzuki, E. and Ichikawa, H., 2012, July. Personalized clothing-recommendation system based on a modified Bayesian network. In *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on* (pp. 414-417). IEEE.

[2] - Linden, G., Smith, B. and York, J., 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing, 7*(1), pp.76-80.

[3] - Gomez-Uribe, C.A. and Hunt, N., 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS), 6*(4), p.13.

[4] - Vellido, A., Lisboa, P.J. and Vaughan, J., 1999. Neural networks in business: a survey of applications (1992–1998). *Expert Systems with applications, 17*(1), pp.51-70.

[5] - Sanger, T.D., 1989. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks, 2*(6), pp.459-473.

[6] - Parlos, A.G., Chong, K.T. and Atiya, A.F., 1994. Application of the recurrent multilayer perceptron in modeling complex process dynamics. *IEEE Transactions on Neural Networks, 5*(2), pp.255-266.

[7] - Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. and Dieleman, S., 2016. Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), pp.484-489.

[8] - Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H., 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

[9] - Kulkarni, T.D., Whitney, W.F., Kohli, P. and Tenenbaum, J., 2015. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems* (pp. 2539-2547).

[10] - Kamijo, K.I. and Tanigawa, T., 1990, June. Stock price pattern recognition-a recurrent neural network approach. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 215-221). IEEE.

[11] - Sak, H., Senior, A. and Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.

[12] - vJaeger, H. and Haas, H., 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science, 304*(5667), pp.78-80.

[13] - Kani, S.P. and Riahy, G.H., 2008, October. A new ANN-based methodology for very short-term wind speed prediction using Markov chain approach. In *Electric Power Conference, 2008. EPEC 2008. IEEE Canada* (pp. 1-6). IEEE.

[14] - van den Driessche, P. and Zou, X., 1998. Global attractivity in delayed Hopfield neural network models. *SIAM Journal on Applied Mathematics*, *58*(6), pp.1878-1890. [15] - Salakhutdinov, R. and Hinton, G., 2009, April. Deep boltzmann machines. In *Artificial Intelligence and Statistics* (pp. 448-455).

[16] - Huang, G.B., Zhu, Q.Y. and Siew, C.K., 2004, July. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on* (Vol. 2, pp. 985-990). IEEE.

[17] - Hoffmann, M., 2005. Numerical control of Kohonen neural network for scattered data approximation. *Numerical Algorithms*, *39*(1), pp.175-186.

[18] - Graves, A., Wayne, G. and Danihelka, I., 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401.*

[19] - Quinlan, J.R., 1986. Induction of decision trees. *Machine learning*, *1*(1), pp.81-106.

[20] - Navada, A., Ansari, A.N., Patil, S. and Sonkamble, B.A., 2011, June. Overview of use of decision tree algorithms in machine learning. In *Control and System Graduate Research Colloquium (ICSGRC), 2011 IEEE* (pp. 37-42). IEEE.

[21] - Salzberg, S.L., 1994. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, *16*(3), pp.235-240. [22] - Compartive Study ID3, Cart and C4.5 decision tree algoritm

[23] - Loh, W.Y., 2014. Fifty years of classification and regression trees. *International Statistical Review*, *82*(3), pp.329-348.

[24] - Cho, Y.H., Kim, J.K. and Kim, S.H., 2002. A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications*, *23*(3), pp.329-342.

25] - Ozgulbas, N. and Koyuncugil, A.S., 2009, April. Developing Road Maps for Financial Decision Making by CHAID Decision Tree: CHAID Decision Tree Application. In *Information Management and Engineering, 2009. ICIME'09. International Conference on* (pp. 723-727). IEEE.

[26] - Abdi, H., 2010. Holm's sequential Bonferroni procedure. *Encyclopedia of research design*, *1*(8).

[27] - Loh, W.Y. and Shih, Y.S., 1997. Split selection methods for classification trees. *Statistica sinica*, pp.815-840.

[28] - Loh, W.Y., 2009. Improving the precision of classification trees. *The Annals of Applied Statistics*, pp.1710-1737.

[29] - Horný, M., 2014. Bayesian Networks. Boston University.

[30] - Rane, N.P. and Patil, D.D., 2015, September. Automatic annotating SRRs from web databases using Naive Bayes approach. In *Computer, Communication and Control (IC4), 2015 International Conference on* (pp. 1-6). IEEE.

[31] - Bielza, C. and Larrañaga, P., 2014. Discrete Bayesian network classifiers: a survey. *ACM Computing Surveys (CSUR)*, *47*(1), p.5.

[32] - Kononenko, I., 1991. Semi-naive Bayesian classifier. In *Machine Learning—EWSL-91* (pp. 206-219). Springer Berlin/Heidelberg.

[33] - https://www.researchgate.net/profile/Petr_Musilek/publication/4046564_Discriminative_parameter_learning_of_general_Bayesian_network_classifiers/links/09e4150fd723108072000000.pdf

[34] - Shen, B., Su, X., Greiner, R., Musilek, P. and Cheng, C., 2003, November. Discriminative parameter learning of general Bayesian network classifiers. In *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on* (pp. 296-305). IEEE.

[35] - Kolstad, A., Özgöbek, O., Gulla, J.A. and Litlehamar, S., 2017. Rethinking Conventional Collaborative Filtering for Recommending Daily Fashion Outfits.

[36] - Su, X. and Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence, 2009*, p.4.

[37] - Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. *ACM*

Transactions on Information Systems (TOIS), *22*(1), pp.5-53. [38] - Wang, B., Tao, Z. and Hu, J., 2010, August. Improving the diversity of user-based top-N recommendation by cloud model. In *Computer Science and Education (ICCSE), 2010 5th International Conference on* (pp. 1323-1327). IEEE.

[39] - Karypis, G., 2001, October. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 247-254). ACM.

[40] - Hauke. J. and Kossowski, T., 2011. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. *Quaestiones geographicae*, *30*(2), p.87.

[41] - J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98), 1998.

[42] - Pham, M.C., Cao, Y., Klamma, R. and Jarke, M., 2011. A clustering approach for collaborative filtering recommendation using social network analysis. *J. UCS*, *17*(4), pp.583-604.

[43] - Shani, G., Heckerman, D. and Brafman, R.I., 2005. An MDP-based recommender system. *Journal of Machine Learning Research*, *6*(Sep), pp.1265-1295.

[44] - Thorat, P.B., Goudar, R.M. and Barve, S., 2015. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, *110*(4).

[45] - Hofmann, T., 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1). pp.89-115.

[46] - McAuley, J., Targett, C., Shi, Q. and Van Den Hengel, A., 2015, August. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 43-52). ACM.

[47] - Veit, A., Kovacs, B., Bell, S., McAuley, J., Bala, K. and Belongie, S., 2015. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4642-4650).

[48] - Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C. and Yan. S., 2012, October. Hi, magic closet, tell me what to wear!. In *Proceedings of the 20th ACM international conference on Multimedia* (pp. 619-628). ACM.

[49] - Hu, Y., Yi, X. and Davis, L.S., 2015, October. Collaborative fashion recommendation: A functional tensor factorization approach. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 129-138). ACM.

[50] - Yu, L.F., Yeung, S.K., Terzopoulos, D. and Chan, T.F., 2012. DressUp!: outfit synthesis through automatic optimization. *ACM Trans. Graph.*, *31*(6), pp.134-1.

[51] - S.K. Lee, Y.H. Cho, S.H. Kim, Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations, Information Sciences 180 (11) (2010) 2142–2155

[52] - O. Zaiane, Building a recommender agent for e-learning systems, in: Proceedings of the International Conference on Computers Education (ICCE'02), vol. 1, 2002, pp. 55–59.

[53] - Wintrode, J., Sell, G., Jansen, A., Fox, M., Garcia-Romero, D. and McCree, A., 2015, April. Content-based recommender systems for spoken documents. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (pp. 5201-5205). IEEE.

[54] - Rohani, V.A., Kasirun, Z.M. and Ratnavelu, K., 2014, December. An Enhanced Content-Based Recommender System for Academic Social Networks. In *Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on* (pp. 424-431). IEEE.

[56] - Ferdous, S.N. and Ali, M.M., 2017, February. A semantic content based recommendation system for cross-lingual news. In *Imaging, Vision & Pattern Recognition (icIVPR), 2017 IEEE International Conference on* (pp. 1-6). IEEE.

[57] - Ghauth, K.I. and Abdullah, N.A., 2011. The effect of incorporating good learners' ratings in e-Learning content-based recommender System. *Journal of Educational Technology & Society*, *14*(2), p.248.

[58] - Burke, R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), pp.331-370.

[59] - Krulwich, B.: 1997, 'Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data'. Artificial Intelligence Magazine 18 (2), 37-45.

[60] - Wang, Y., Chan, S.C.F. and Ngai, G., 2012, December. Applicability of demographic recommender system to tourist attractions: A case study on trip advisor. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*(pp. 97-101). IEEE Computer Society.

[61] - Liang, S., Liu, Y., Jian, L., Gao, Y. and Lin, Z., 2011, August. A utility-based recommendation approach for academic literatures. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on* (Vol. 3, pp. 229-232). IEEE. [62] - Brin, S. and Page, L.: 1998, 'The anatomy of a large-scale hypertextual {Web} search engine'. Computer Networks and ISDN Systems, 30(1-7), 107-117.

[63] - Kokol, P., Verlic, M. and Krizmaric, M., 2006. Modeling teens clothing fashion preferences using machine learning. *WSEAS Transactions on Information Science and Applications*, *3*(10). pp.2054-2065.

[64] - Magoulas, G.D., Vrahatis, M.N. and Androulakis, G.S., 1999. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. *Neural Computation*, *11*(7), pp.1769-1796.

[65] - Multi-Layer Versus Single Layer Neural Networks and an application

[66] - Kim, Y.S. and Yum, B.J., 2004. Robust design of multilayer feedforward neural networks: an experimental approach. *Engineering Applica-*

methodology. *IEEE transactions on systems, man, and cybernetics*, *21*(3), pp.660-674.

[69] - Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R. and Honrao, V., 2013. Predicting students' performance using ID3 and C4. 5 classification algorithms. *arXiv preprint arXiv:1310.2071*.

[70] - Soni, S., 2010. Implementation of multivariate data set by CART algorithm. *Journal of Information Technology and Knowledge Management*, *2*(2), pp.455-459.

[71] - Gordon, L., 2013. Using classification and regression trees (CART) in SAS® enterprise miner TM for applications in public health. *Public Health*.

[72] - Leray, P. and François, O., 2005, June. Bayesian network structural learning and incomplete data. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005)* (pp. 33-40). Espoo, Finland.

[73] - Chickering, D.M., Geiger, D. and Heckerman, D., 1994. *Learning Bayesian networks is NP-hard* (Vol. 196). Technical Report MSR-TR-94-17, Microsoft Research.

[74] - Cacheda, F., Carneiro, V., Fernández, D. and Formoso, V., 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, *5*(1), p.2.

[75] - Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), pp.5-53.

[76] - Müller, H., Michoux, N., Bandon, D. and Geissbuhler, A., 2004. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *Interna-*

[78] - Pazzani, M.J. and Billsus, D., 2007. Content-based recommendation systems. In *The adaptive web* (pp. 325-341). Springer, Berlin, Heidelberg.

[79] - Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), pp.734-749.

[80] - Huang, S.L., 2008. Comparision of utility-based recommendation methods. *PACIS 2008 Proceedings*, p.21

[81] - Larose, D.T., 2004. Decision trees. *Discovering Knowledge in Data: An Introduction to Data Mining*, pp.107-127.

[82] - Burke, R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction, 12*(4), pp.331-370.

[83] - Crane, D., 2012. *Fashion and its social agendas: Class, gender, and identity in clothing.* University of Chicago Press.

[84] - Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), pp.82-97.

[85] - Gee, J.P., 2005. Learning by design: Good video games as learning machines. *E-learning and Digital Media*, 2(1), pp.5-16.

[86] - Mooney, R.J. and Roy, L., 2000, June. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*(pp. 195-204). ACM.

An impressive amount of references, in a standard academic format

ECM3401 Individual Literature Review and Project

Feedback Sheet for Literature Review and Project Specification

This assessment contributes 20% of the marks for the module.

Student:[redacted] **Marker:**[redacted]

Title: How effectively can multiple machine learning techniques recommend outfits?

INTRODUCTION TO TOPIC

Criterion: The topic area has been clearly defined and motivated with reference to the project.
The abstract is clear but perhaps could have been more concise. There is a very good introduction to the topic. The goal of the project is clear and interesting.

BREADTH AND DEPTH OF REVIEW

Criterion: An appropriate number of sources has been reviewed, in sufficient depth.
Excellent literature review, both broad to contexualise the area, and deep on the specific subject. At time, it could have been clearer if more examples were provided. An impressive number of references are cited. However, in some cases the presentation is simply mentioning the names of different techniques, without going to sufficient depth. Given the very large number of papers cited, this is understandable. In addition, there is an important category of recommender systems missing from the relevant section (Section 2.2): Methods based on low-rank matrix completion. See for example: E. J. Candès and B. Recht. Exact matrix completion via convex optimization. Found. of Comput. Math., 9 717-772. There are also a few citations missing in some important parts of the presentation: please see hand-written annotations.

CRITICAL ANALYSIS AND SYNTHESIS

Criterion: A good overall understanding of the topic has been demonstrated, using ideas brought together from different sources.
A good, systematic critical analysis is presented, which shows a good understanding of the corresponding methodologies.

SPECIFICATION

Criterion: There is a good discussion of functional and non-functional requirements and, where applicable, research hypotheses.
The specification is very good. Perhaps, figure 3 with an overall description of the system should have been provided before explaining the details of the system. Much details about the system is provided. The envisioned project is perhaps a little bit ambitious, and it would be better to focus on one or two approaches.

EVALUATION CRITERIA

Criterion: Clear criteria have been provided for evaluating the final product and results.
Well-justified and measurable criteria. It is worth mentioning that due to the nature of the problem of automatic outfit recommendation, it is very challenging to design a meaningful quantitative evaluation.

SUMMARY AND CONCLUSION

Criterion: There is a good overall summary, with conclusions to take forward to the project itself.
Good comprehensive summary and conclusion with reference to the next step in the project.

REFERENCES

Criterion: References are correctly and consistently cited in an approved style, with all external material explicitly and unambiguously acknowledged.
Consistent references in a uniform, approved format.

PRESENTATION

Criterion: The document is well-structured and attractively produced, written in good English, with few or no typographical, spelling, or grammatical errors.
The document has a clear structure. The English is generally good, but at times it uses terms without defining them. The paper is a little too packed. The presentation is generally clear. The advantages and disadvantages of each methodology could have been discussed when the methodology was introduced. Then, the presentation would be smoother and the introduction of additional methods would in most of the cases be able to be motivated better, as trying to overcome the