**Project: Recommender Systems (Virtual Presentation)**

Introduction:

- This project focuses on developing and implementing recommender systems, commonly used in data science.

- The project requires using Python and Jupyter Notebook to create solutions and present the results virtually.

General Requirements:

- Use Jupyter Notebook and appropriate Python libraries.

- Choose the MovieLens 1M Dataset for movie recommendation.

- Implement three tasks and create a presentation:

Task 1: User-based Collaborative Filtering

- Develop user-based collaborative filtering using KNN.

- Choose a similarity metric.

- Implement in Python, study the impact of parameter K, and use RMSE for evaluation.

- Summarize results concisely in a Word document and presentation.

Task 2: Item-based Filtering

- Develop item-based collaborative filtering using KNN.

- Choose K and compare at least two similarity metrics.

- Implement in Python, use RMSE for evaluation, and summarize results.

Task 3: A Better Recommender System

- Choose Option 1 (based on related publications) or Option 2 (propose a new algorithm) for creating a better recommender system.

- Implement in Python with detailed comments.

- Evaluate the system and compare it with Movie Average and KNN-based Collaborative Filtering using AP and NDCG metrics.

- Visualize the results and summarize findings.

Task 4: Presentation

- Create a Word document and slides. (ppt?)

- Include key results for Tasks 1, 2, and 3.

- Describe the "new" solution for Task 3.1 with proper citations.

- Include literature review if applicable.

- Provide necessary details of the algorithm.

- Present key results, visualizations, and findings for Task 3.2.

- Include a list of references.

Presentation Requirements:

- Create no more than 10 slides.

- There's no template provided.

Submission:

- Submit a Jupyter Notebook file named `assignment3.ipynb`.

- Clean the code and remove unnecessary lines.

- Ensure comments are included.

Remember to follow the guidelines closely for each task, and make sure to document your work thoroughly and present your findings clearly in your presentation.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
RP1: summary

This is topic of recommendation systems, with a specific focus on collaborative filtering and the use of autoencoders. This introduction sets the stage for the content of the paper, outlining the importance of recommendation systems in today's data-driven world and their relevance in various domains.

introduction:

1. The Information Age and the Internet of Things (IoT) have led to the explosion of data, giving rise to the concept of Big Data.

2. The sheer volume of data can overwhelm users, making it challenging to find what they need or prefer.

3. Recommendation systems have emerged as a solution to this problem, helping users discover relevant items and make better choices in various domains.

4. Recommendation systems aim to reduce information overload and provide personalized suggestions.

5. The paper discusses the origins and evolution of recommendation systems, highlighting the role of collaborative filtering techniques.

6. Collaborative filtering can be categorized into memory-based and model-based approaches.

7. Matrix Factorization is a well-established algorithm in recommendation systems, but deep learning approaches, such as autoencoders, are gaining attention for their potential to capture complex relationships.

8. The paper focuses on using autoencoders for collaborative filtering tasks and compares their performance with Singular Value Decomposition (SVD) using the Root Mean Square Error (RMSE) metric.

9. The paper outlines the structure of the paper, including sections on related work, methodology, datasets, findings, discussion, and conclusions.

This introduction provides a comprehensive overview of the importance of recommendation systems and the research focus of the paper. It sets the stage for the reader to understand the context and significance of the study.  
  
  
  
Related Work :

1. Collaborative Filtering Using SVD: Various studies have implemented recommendation systems using collaborative filtering techniques, particularly based on Singular Value Decomposition (SVD) technology. These methods predict item ratings for users by finding neighbors and making recommendations using Pearson's similarity correlation measurement.

2. Scalability of SVD-Based Models: Some researchers have proposed techniques to incrementally build SVD-based models, aiming to make recommender systems more scalable.

3. Combining Clustering and SVD: A novel approach combines a clustering algorithm with an SVD algorithm to decompose the rating matrix and calculate the similarity between users. This combined method is used for collaborative filtering recommendations.

4. Challenges in Handling Data Complexity: Recommendation systems face challenges due to the enormous volume, complexity, and dynamics of data. To address these challenges, researchers have turned to deep learning techniques.

5. Use of Autoencoders: Recent approaches have been utilizing autoencoders for recommendation systems. These include FlexEncoder, AutoRec, stacked auto-encoders with denoising, and other autoencoder-based models. Autoencoders are considered efficient and have shown promising results in collaborative filtering tasks.

6. Collaborative Variational Autoencoder (CVAE): A Bayesian generative model called CVAE considers both ratings and content for making recommendations in multimedia scenarios. CVAE outperforms state-of-the-art recommendation methods.

7. Deep Autoencoder with Iterative Output Re-feeding: A deep autoencoder with six layers is proposed, which is trained end-to-end without layer-wise pre-training. An innovative training algorithm based on iterative output re-feeding is used to overcome sparseness in collaborative filtering.

8. Encouragement for Autoencoder Research: A comparative analysis of different autoencoder-based recommender systems suggests that the application of autoencoders in recommendation systems is still in its early stages, encouraging further research in this direction.

9. Proposed Approach: The paper introduces a recommendation system based on an autoencoder with a drop-out layer. This approach is aimed at addressing the limitations of matrix factorization techniques like SVD, especially in handling sparse and large data.

In summary, the "Related Work" section highlights the evolution of recommendation systems, with a growing interest in using autoencoders as a promising approach. It also emphasizes the need for further research and innovation in this field.

Methodology:

1. Introduction to Machine Learning and Data Mining: The section highlights the significance of machine learning and data mining in engineering and computer science. Machine learning involves building computer systems that can adapt, learn, and improve their performance through experience, while data mining incorporates mathematical functions, machine learning techniques, and statistical analysis to discover hidden patterns and extract meaningful knowledge from data.

2. Categories of Data Mining Strategies: Data mining strategies are categorized into three main types - supervised learning, unsupervised learning, and semi-supervised learning. In supervised learning, a training set is used to predict a target variable. In unsupervised learning, there is no target variable, and the focus is on discovering patterns and relationships in the data. Semi-supervised learning combines elements of both, with a limited set of examples available for predicting target values.

3. Usage of Autoencoder for Collaborative Filtering: The study employs an unsupervised learning technique called a neural network autoencoder, based on collaborative filtering, to create a product recommendation system. Autoencoders are used for feature extraction and dimensionality reduction.

4. TensorFlow 2.0.0: The research utilizes TensorFlow 2.0.0 for model creation and training. TensorFlow is a flexible and scalable framework that supports large-scale training and inference, making it suitable for machine learning experiments and research.

5. CRISP-DM Methodology: The Data Mining process follows the CRISP-DM Methodology (Cross Industry Standard Process for Data Mining). This methodology comprises six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. It is depicted as a cyclic process.

Business Understanding:

1. Consumer Demand for Differentiated Content: Consumers are increasingly seeking differentiated content and demand high standards of innovation, sophistication, and customization from brands. Recommendation systems are seen as a strategy to address this demand by providing users with intelligent and proactive information services.

2. Impact on Company Revenue: In the context of online shopping, the effectiveness of a recommendation system can directly affect a company's revenue. Well-targeted recommendations can significantly influence users' purchase decisions.

3. Ethical Concerns: Ethical considerations surrounding recommendation systems are an important aspect to address. Concerns about privacy and the rights of consumers have prompted organizations to maintain secrecy regarding the operational details of their recommendation systems, which can hinder research progress.

4. Business Objectives: The study identifies several key objectives guiding the project, including increasing sales, improving company revenue, encouraging engagement and activity on products and services, gaining a competitive advantage, calibrating user preferences, making personalized recommendations, and finding the recommendation algorithm and parameterization that results in the highest overall performance of the product recommendation system.

5. Relationship between Business Goals and Data Mining: The first five objectives are directly related to the business goals, with a focus on improving product recommendation systems, customer satisfaction, and company profitability. The remaining objectives are associated with the data mining (DM) process, which aims to provide insights into the recommendation system through the application of DM techniques.

6. Project Plan: A project plan is designed to assess the techniques to be used in the subsequent stages of the project. The plan outlines the steps to be executed to achieve both DM and business goals.

In summary, the "Business Understanding" section emphasizes the importance of recommendation systems in addressing consumer demands and their impact on company revenue. It also acknowledges ethical concerns related to recommendation systems and sets clear objectives for the study to improve recommendation quality and refine the DM techniques. A project plan is developed to guide the project toward its goals.

Data Understanding:

1. Dataset Description:

- The study uses two MovieLens (ML) datasets: MovieLens1M and MovieLens10M.

- MovieLens1M contains 1 million ratings from 6040 users for 3706 movies.

- MovieLens10M contains 10 million ratings from 69,878 users for 10,677 products (movies).

- Ratings are on a scale from 1 to 5, with 1 indicating a bad rating and 5 indicating an excellent rating.

- The study treats movie IDs as product IDs, as the goal is to recommend products.

- The datasets' sparsity is high, with a sparsity value of 0.996, indicating that many user-movie interactions are missing.

2. Additional Product Dataset:

- An additional dataset with 180,249 products is used.

- This dataset is a combination of four different datasets found on the Kaggle platform, which includes data from Amazon, Macy's, Shop\_norstrom, and Thrift Store.

- This dataset contains information related to the products.

3. Dataset Usage:

- Only the MovieLens datasets, which contain user ratings for products (movies), are used for model training.

- The additional product dataset is utilized later to retrieve product information when providing recommendations to users.

- The product ID from the model's output is matched with the product dataset to retrieve detailed product information for recommendations.

Data Preparation:

1. Data Preparation Significance: Data preparation is highlighted as a critical stage, as the success of the data mining process depends on it. This stage involves data transformation and cleaning tasks.

2. Data Treatment for MovieLens Datasets:

- In the case of the MovieLens datasets, items with fewer than 20 reviews and users who had not provided more than 20 reviews were eliminated. This selection aimed to enhance results, as collaborative filtering relies on sufficient data to avoid issues.

- Duplicate instances were identified and removed to prevent ambiguity.

- Ratings were normalized, converting the values into floats and restructuring them to fit within the range of [-1, 1]. This changed the scale from [1–5] to [0.20–1].

3. Data Splitting: Both MovieLens datasets were divided into an 80% training set and a 20% testing set.

4. User and Item Counts: Table 1 is mentioned, which likely provides information on the number of users and items remaining after the data preparation stage. However, the specific details from Table 1 are not provided in the summary.

In summary, the "Data Preparation" section underscores the importance of data preparation in data mining. It explains the specific steps taken to clean, transform, and split the data, especially focusing on the MovieLens datasets, and their impact on the scale of ratings.

Modeling:

1. Introduction to Modeling: The section emphasizes the importance of applying modeling techniques and their parameters to the learning dataset to determine the best-performing model in the evaluation stage.

2. Autoencoder Architecture: The study uses an autoencoder for building the product recommendation system. While the details of the architecture are not provided in this summary, it is likely explained in the full section.

3. Explanation of Autoencoders: The section briefly explains what autoencoders are and how they function. Autoencoders are neural networks designed to reconstruct their input data, and they consist of an encoder and a decoder. They are often used for dimensionality reduction and feature extraction.

4. Parameters for Achieving Results: The study mentions that the section provides details about the parameters used in the autoencoder model to achieve the results discussed in Section 4. These parameters are critical in determining the performance of the recommendation system.

In summary, the "Modeling" section introduces the use of an autoencoder in the product recommendation system and provides information about the architecture, how autoencoders work, and the parameters used to achieve the results.

Autoencoder Overview:

1. Autoencoder Definition: An autoencoder is an unsupervised deep learning method that learns to compress and encode data effectively. It then reconstructs the data from the encoded representation, aiming to make the reconstructed representation identical to the original input.

2. Structure of an Autoencoder: A typical autoencoder consists of three layers: the input layer, the hidden layer, and the output layer. The input and hidden layers form an encoder, while the hidden and output layers form a decoder. A code represents the compressed input.

3. Encoder and Decoder Functions:

- Encoder: It compresses high-dimensional input data into a low-dimensional hidden representation using a function.

- Code: Represents the compressed input, which is then fed to the decoder.

- Decoder: Reconstructs the original information from the hidden representation, moving from the latent space back to the original information space.

4. Mathematical Representation: The encoder maps input data "x" to the hidden representation "h" using an activation function and a weight matrix "W" along with a bias vector "b." Similarly, the decoder maps "h" back to a reconstruction "x0" using another activation function, a weight matrix "W0," and a bias vector "b0."

5. Autoencoder's Purpose: The primary goal of an autoencoder is to obtain a lower-dimensional representation of data, such that the error between the original input and the reconstruction is minimized. Autoencoders are known for their performance in data dimensionality reduction, noise removal, feature extraction, and data reconstruction.

6. Integration in Recommendation Systems: Autoencoders have shown high efficiency in information retrieval and recommendation tasks. Researchers have integrated autoencoders into recommendation systems to improve the accuracy of recommendations by better understanding user-item relationships, learning non-linear patterns, and encoding complex abstractions into data representations.

In summary, the "Autoencoder Overview" section explains the concept of autoencoders, their structure, mathematical representation, and their application in recommendation systems. It highlights their role in improving the accuracy of recommendations by leveraging their capacity to learn and represent data efficiently.

Initialization of Parameters:

1. Parameter Selection: The choice of parameters for the autoencoder model was based on an extensive experimentation process, given the diverse set of parameters used in various articles. The approach followed was "trial and error."

2. Loss Function: The loss function chosen for this model is Mean Square Error (MSE). This decision is because the problem at hand doesn't involve classification but rather involves making recommendations.

3. Optimizer: The Adam optimizer was selected for its speed, memory efficiency, and suitability for handling large datasets.

4. Dropout: Dropout is applied during the training phase to prevent overfitting and is automatically disabled during execution.

5. Activation Function: The choice of activation functions, such as Tanh for the encoder layer, latent space, and Linear for the output layer, was made based on the range of network values and the results of experiments. Other functions were tested, but these choices yielded better loss and RMSE results.

Architecture:

1. Dropout Layer: The final architecture includes a dropout layer. This layer helps prevent overfitting and allows for the efficient combination of various neural network architectures.

2. Architecture Visualization: Figure 6 provides a visual representation of the architecture used for achieving the results discussed in Section 4.

3. Dropout in Neural Network: The dropout layer operates by temporarily removing units (both hidden and visible) from the neural network, along with their incoming and outgoing connections, to prevent overfitting.

In summary, the study details the specific parameters chosen for the autoencoder model, explains the rationale behind their selection, and introduces the architecture used for the recommendation system, highlighting the role of the dropout layer in preventing overfitting.

In the "Evaluation" section, the study discusses the evaluation metrics used to assess the accuracy of the recommendation system. Here's a summary of the key points in this section:

Evaluation Metrics:

1. Classification Accuracy Metrics: Recommendation systems for online retailing shops can be evaluated using classification accuracy metrics. These metrics go beyond accuracy and encompass aspects related to the technical performance and life-cycle of the system, including responsiveness, scalability, reliability, and maintenance.

2. Aspects to Consider: Other critical aspects that should be considered in the evaluation of recommendation systems include coverage, confidence, trust, and security. Measuring these aspects is a complex task that extends beyond the scope of the current study.

3. Metrics Used: The study focuses on metrics for accuracy evaluation, which are categorized into statistical precision and precision of decision support.

Statistical Accuracy Metric: RMSE

- The Root Mean Square Error (RMSE) is used as the statistical accuracy metric. It places greater importance on larger absolute errors.

- RMSE is calculated using the expected evaluation (p(u, i)) and the actual user evaluation (r(u, i)) for each user-item pair (u, i).

- The formula for RMSE is provided in the article, and the lower the RMSE, the more accurate the recommendation mechanism will be when predicting user reviews.

Decision Support Accuracy Metrics: Precision and Recall

- Precision and recall are used as decision support accuracy metrics.

- Precision represents the fraction of recommended items that are relevant to the user. It quantifies the number of recommended items that are of interest to the user in relation to the total number of recommended items.

- Recall, on the other hand, measures the fraction of relevant items that are part of the recommended item set. It indicates the number of items of interest to the user that are included in the recommended items.

- Formulas for calculating precision and recall are mentioned in the article.

In summary, the study uses RMSE as a statistical accuracy metric to measure the accuracy of the recommendation system in predicting user reviews. It also employs precision and recall as decision support accuracy metrics to evaluate the relevance of recommended items to users. The lower RMSE and the higher precision and recall values indicate a more accurate and effective recommendation system.

In the "Results and Discussion" section, the study presents and discusses the performance of the recommendation system using different datasets and architectures. Here is a summary of the key findings and discussions in this section:

Model Performance with ML-1M Dataset:

- Figure 8 displays the performance of the model using the ML-1M dataset.

- Both the loss and val\_loss are decreasing over each epoch and begin to stabilize. The dropout layer, which is only active during training, contributes to the difference between the loss and val\_loss curves.

- The model avoids overfitting, as shown in Figure 9.

- Figure 11 demonstrates the growth of precision and recall values throughout each epoch.

Model Performance with ML-10M Dataset:

- Figure 10 shows the loss model using the ML-10M dataset, and it exhibits a similar behavior to the ML-1M dataset but with lower loss and val\_loss values.

- The accuracy/recall graph also shows a good behavior towards the data with a threshold of 0.6 for both metrics.

Comparison of Different Autoencoder Architectures:

- The study experimented with different numbers of layers in the autoencoder architecture, including one, two, and three layers.

- The results showed that increasing the number of layers led to higher RMSE values, indicating that a simpler architecture (one layer) is more effective.

Comparison with SVD Model:

- A comparison was made between the autoencoder model and a Singular Value Decomposition (SVD) model.

- The RMSE results demonstrated that the autoencoder outperformed the SVD model in both the ML-1M and ML-10M datasets.

Visual Comparison of Recommendations:

- Figure 12 displays ten products that the user had already evaluated with high ratings.

- Figure 13 shows ten products suggested by the model for comparison.

In summary, the results indicate that the autoencoder-based recommendation system outperforms the SVD model, especially for sparse datasets like ML-1M and ML-10M. The study also confirms that a simpler architecture with one layer is more effective than adding additional layers. The visual comparison of recommendations suggests that the model is providing relevant and accurate recommendations for users.