Talent.AI - Recommendation Algorithms Results and Conclusion

Yasmin Adler - 208462184

Shenkar College
Faculty of Engineering

Supervisor: Prof. Avivit Levy
Supervisor: Dr. Riva Bracha Shalom
Supervisor: Dr. Michal halamish

GitHub Repository - Includes all results and code

Abstract

This research focuses on building and testing a recommendation system for matching employees to positions. It uses variations in datasets of gender and age, and K means to analyze how gender and age affect the results. By creating datasets with different combinations of gender and age attributes, we tested two main recommendation methods: multi-clustering and position-to-applicant. These algorithms were evaluated using metrics like precision, recall, F1-score, and ranking quality (NDCG). We also examined how to reduce bias in recommendations to make the process fairer

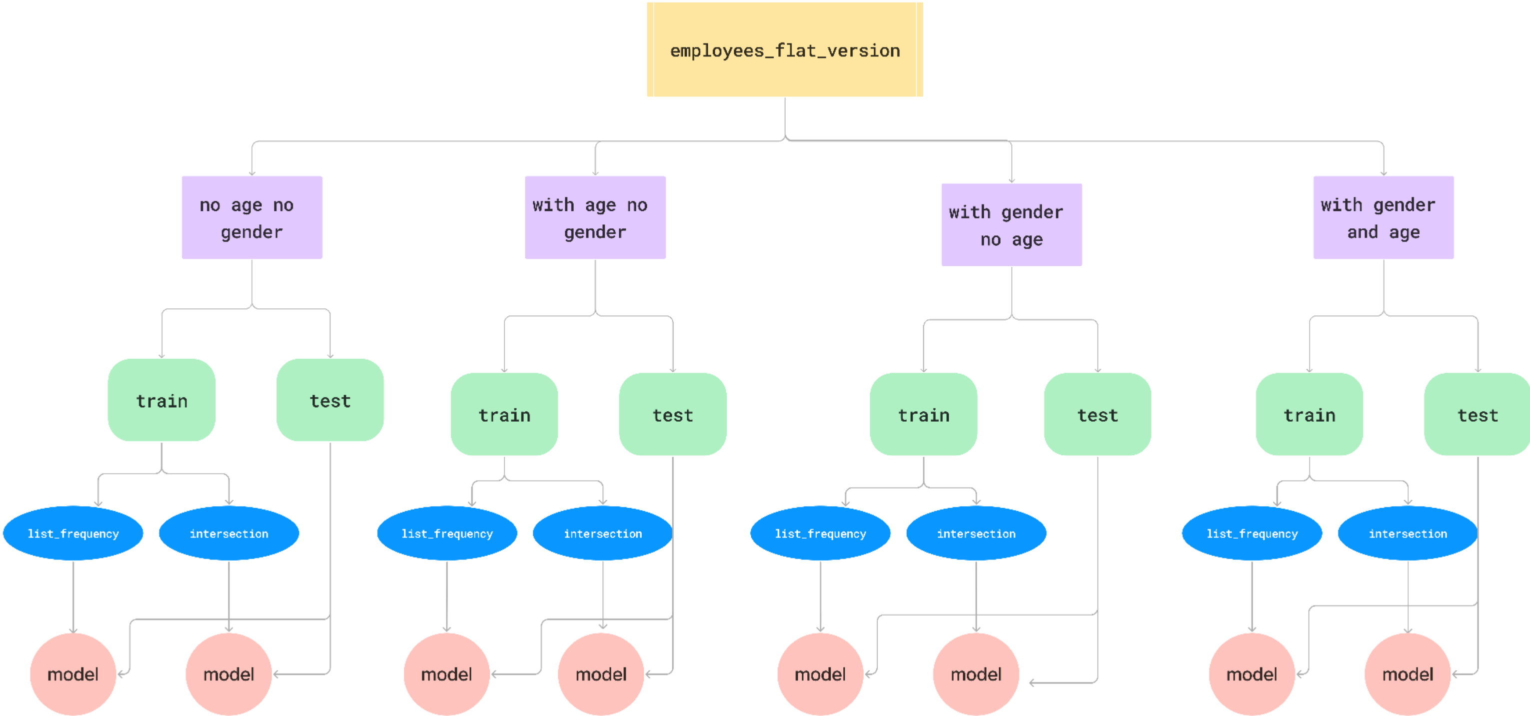
---

General Description

The dataset we used contains details about employees, such as their names, gender, date of birth, current company, and other traits. To see how sensitive information like gender and age influences recommendations, we created four versions of the dataset:

1. **With Gender and Age:** Includes both gender and age information.
2. **Without Gender (With Age):** Gender is removed, but age is included.
3. **Without Age (With Gender):** Age is removed, but gender is included.
4. **Without Gender and Age:** Both gender and age are removed.

Each version was split into training and testing sets, creating eight groups in total. We then applied clustering models to these datasets, using two distance measurement methods: “**Statistic_list_frequency**” and “**Statistic_intersection**.” The results from these clusters were used to test our recommendation algorithms.



The Approach

Dataset Variations

The dataset variations allowed us to study how the presence or absence of gender and age data affects clustering and recommendations. By comparing the results from these different versions, we aimed to identify any biases and understand the role of these sensitive attributes in the process.

Clustering Models

For each dataset version, we followed these steps to create the clustering models:

1. **Data Preparation:**

- Handled missing information carefully. For example, numbers were normalized (scaled), and text-based categories were converted into numerical values so they could be processed by the algorithms.
- Ensured that the data format met the requirements of the clustering process.

2. **Choosing Distance Functions:**

    - **Statistic_list_frequency:** Measures similarity by counting how often certain values appear.
    - **Statistic_intersection:** Focuses on overlapping features between data points to determine similarity.

3. **Clustering Process:**
    - Used the KMeans method to divide the data into clusters, with the number of clusters (= 8) chosen based on prior experiments (made by Dana).
    - Checked that the clusters were meaningful and that there were no empty or duplicate groups.

System Flow

The system works in three main steps:

1. **Creating Models:**

    - **Set-Up:** The dataset version and distance function are chosen, and the system is configured accordingly.
    - **Clustering:**
        - The full dataset is used to create clusters that capture overall patterns.
        - A training set is clustered separately to train a model that understands the relationships in the data.
    - **Validation:** We checked that the clusters made sense and represented the data well.

2. **Running Recommendation Algorithms:**
    - Tested the recommendation methods using the models and the test data.
    - Two algorithms were applied:
        - **Multiclustering**: Focuses on hierarchical groupings.
        - **Position-to-Applicant**: Matches candidates to predefined positions (a selected query is treated as an open position).
3. **Evaluating Results:**
    - Measured the quality of the recommendations using metrics like precision, recall, F1-score, and ranking accuracy (NDCG).
    - Calculated the distances between clusters to see how distinct they were.

Recommendations Algorithms in Detail

Multiclustering

The multiclustering method breaks the process into three steps:

1. **Finding the Closest Cluster:**
    - For a given query (like an employee's profile), the system finds the cluster that is most similar using the chosen distance function.
2. **Ranking Subclusters:**
    - Once the main cluster is identified, smaller subclusters within it are ranked based on how close they are to the query.
3. **Providing Results:**
    - The top subclusters and their associated companies are returned, along with distance scores.

This method provides more detailed and precise recommendations because it uses both main clusters and subclusters.

Standard Recommendation

This approach is simpler and works as follows:

1. **Find the Closest Cluster:**
   - The system identifies the cluster that is closest to the query.
2. **Rank Individual Records:**
   - The records in the cluster are ranked by their similarity to the query.
3. **Return Results:**
   - The top-ranked records are presented as recommendations.

This straightforward method is fast and easy to use but may not capture fine-grained details like multiclustering does.

Position-to-Applicant

In this method, queries are treated as positions, and candidates are ranked for each position:

1. **Choosing Positions:**
   - Five sample positions are picked for each company to act as queries.
2. **Ranking Candidates:**
   - All candidates are compared to these positions, and their distances are calculated.
3. **Output Results:**

- The top 11 candidates for each position are listed, showing how well they match the position requirements.

This approach is useful for identifying the best-fit candidates for specific roles.

Statistical Results

Ranking Accuracy with NDCG

The NDCG score is calculated based on the following steps:

1. **Calculate DCG:** For a given query, compute the Discounted Cumulative Gain (DCG), which measures the quality of the ranking. Each item's relevance is divided by the logarithm of its position. Relevant items close to the top receive higher weights.
Here, represents the relevance score (1 for correct, 0 for incorrect), and is the rank.
2. **Calculate IDCG:** Compute the Ideal DCG (IDCG), which represents the best possible ranking score for the query.
3. **Compute NDCG:** Normalize the DCG by dividing it by the IDCG:

Each query receives an NDCG score between 0 and 1. The final result is an average of these scores over all queries. For example, a query where Google was ranked as the target correctly and appeared at the top yielded a high NDCG value. However, when Google appeared further down the ranking, the score dropped proportionally, reflecting the system's weaker performance.

Precision, Recall, and F1-Score

These metrics evaluate how well the recommendation algorithms perform in identifying relevant items:

1. **Precision:** Measures the proportion of recommended companies that were actually relevant. It is defined as:
Where is the number of true positives (correct recommendations), and is the number of false positives (incorrect recommendations).
2. **Recall:** Measures how many relevant companies were successfully recommended. Recall is defined as:
Where is the number of false negatives (missed recommendations).

3. **F1-Score:** Combines precision and recall into a balanced measure, particularly useful when one metric outperforms the other. It is defined as:

Example Analysis: In some queries, companies such as Tesla or Nvidia showed high recall but low precision, meaning that while most relevant items were included, irrelevant items were also recommended. This balance was reflected in moderate F1-scores. Conversely, for companies like Adobe, a high precision but low recall scenario occurred when fewer relevant results were included but were more accurate.

Inter-Cluster Distances and WCSS

1. **Inter-Cluster Distances:**
    - Measures the separation between cluster centers. Greater distances signify distinct and non-overlapping clusters, leading to clearer recommendations. For example, the inter-cluster distance between Cluster 0 and Cluster 1 for the "Statistic_list_frequency" function was consistently larger than for the "Statistic_intersection."
2. **WCSS (Within-Cluster Sum of Squares):**
    - Assesses how tightly clustered the data points are around their centroids. Lower WCSS indicates more compact clusters. WCSS was recalculated for all dataset variations to validate the quality of clustering.

Accuracy Based on X Rankings

Accuracy was evaluated by analyzing the performance of recommendations at varying top-X ranks (e.g., top 1, 3, 5, 11, or 13 results). For each X, the algorithm identified whether the target company was present within the top-X recommendations. This analysis helped assess the system's consistency and precision at different levels of granularity.

For example, a test query targeted at "Google" with X=5 would count as correct if Google appeared anywhere within the top five recommendations. Such measurements provide insights into the broader effectiveness of the system's ranking approach.

Unbiasing

Identifying Bias

Bias evaluation focused on:

1. **Cluster Composition:** Whether larger clusters dominated the results, leading to over-representation of certain companies.
2. **Demographic Sensitivity:** Checked whether adding or removing gender and age attributes affected recommendation accuracy.
3. **Company Representation:** Analyzed the balance of small and large companies in recommendations.

For example, companies like Adobe and IBM were underrepresented in some clusters due to their smaller presence in the dataset. This led to biases in recommendations.

Mitigating Bias

1. **Cluster Rebalancing:**
    - Ensured that cluster composition was not skewed toward larger companies by adjusting sampling within clusters. This was particularly effective for addressing imbalances for smaller companies like Adobe.
2. **Demographic Fairness:**
    - Analyzed results across all dataset variations (with and without gender/age) to identify significant differences in rankings. For example, the inclusion of gender and age sometimes amplified biases, which were mitigated by normalizing these attributes.
3. **Multiclustering Stability:**
    - Validated that the multiclustering method consistently preserved company rankings irrespective of cluster size. This was achieved by ensuring that subcluster rankings were primarily distance-based rather than volume-driven.

By integrating these adjustments, the system provided fairer recommendations without over-prioritizing any specific group or company.

Conclusion

This research highlights how dataset attributes and clustering methods impact the performance of recommendation systems. By comparing different dataset versions and algorithms, we showed how sensitive information like gender and age affects accuracy and fairness. The multiclustering and position-to-applicant methods offer unique advantages, addressing both detailed and broad recommendation needs. Future work could focus on real-time implementation, adding more fairness checks, and testing new distance functions for better results.

Results and folders order:

“**Datasets**” - holds all initial datasets, the main one and all variations of it
“**Final measurements**” **-** holds te f1, ndcg, recall, precision measurements and their code
“**Final_measurements\unbiasing**” -

for_age_And_gender:

holds dififerent files for showing the affect of **age** and **gender** on the recommendation results:

Two from the original point of view (applicant to position):

- STANDARD_biasing_for_age_and_gender.xlsx
- MULTICLUSTERING_biasing_for_age_and_gender.xlsx

And two files to show the same biasing for age and gender but for the second point of view:

- Position_to_applicant_women_in_top_X.xlsx
- Position_to_applicant_Age_in_top_X.xlsx

For_company_size:

All files are showing the unbiasing for company size in each **dataset variation** for all companys and for both algorithms: **standard** and **multiclustering**

Final_measurements\inter_cluster_and_wcss:

- intercluster distance for all datasets variations
- the wcss in a different file

Results -

- Shows the recommendations results themselves, for each dataset variation
- Shows recommendations of direction 2 (position to applicant)