Match Made in ML – A Machine Learning Approach to Online Dating

Amber Gonzalez-Pacheco
Abhay Padmanabhan
Sakshi Kumar

Executive Summary

This project explores the use of AI in online dating by leveraging Facebook AI Similarity Search (FAISS) with cosine similarity to match dating profiles from the publicly available OKCupid dataset. We implemented three methods: a baseline Cosine Similarity model, a FAISS-based model, and an optimized Weighted FAISS model tuned via grid search and cross-validation. Our grid search identified the best weight combination (Essay=2.0, Features=0.5, Numerics=0.5) with improved performance metrics, achieving a MAP of 0.7699 and an MRR of 0.9838. These results demonstrate that our optimized Weighted FAISS model provides high ranking quality and efficiency—key factors in making meaningful connections in online dating.

Introduction

The use of AI in online dating is revolutionising the way people connect and find potential partners. There is a diverse range of online dating applications emerging in recent years, and each application possesses its own flair while satiating different needs of the consumer market by leveraging its own matching mechanism. But, all dating applications have a common business objective of making meaningful connections, and to achieve this all applications employ complex algorithms to match platform users on various compatibility factors. In this machine learning project, "Match Made in ML", we aim to utilise Facebook AI Similarity Search (FAISS) with cosine similarity to match dating profiles from the publicly available OKCupid dataset. This report details the approach taken to matchmaking in the context of complex techniques used in online dating applications today.

Background and Motivation

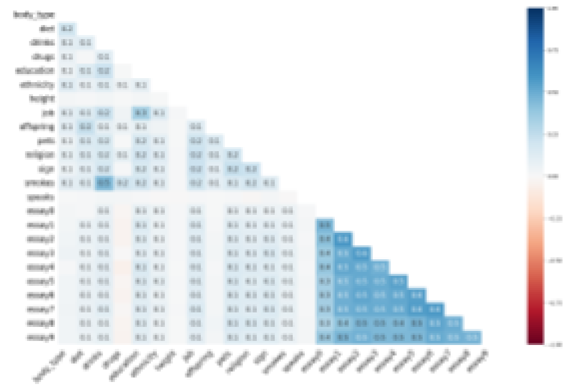
The most popular dating applications today utilize various machine learning algorithms to achieve success in matchmaking - this usually means users end up meeting face-to-face. Dating applications operate on a fundamental principle of connecting individuals based on shared interests, preferences, and potential compatibility. Estimating compatibility is crucial in attaining a 'perfect' match, and some well-known techniques used include: Content-based Filtering, Collaborative Filtering, and Cosine Similarity score. Content-based filtering recommends matches to users based on similar user preferences and profile characteristics, whereas collaborative filtering relies on the fact that users with similar interests most likely also have matches in common with each other. Similarity scores are derived from algorithms that analyze user profiles and assign similarity scores based on shared interests, hobbies, and other relevant attributes; the higher the similarity score, the stronger the potential compatibility. In particular, cosine similarity mathematically leverages the cosine angle between two vectors of user profiles in the multi-dimensional profile features space to determine similarity.

We decided to implement cosine similarity on the large dataset of 60,000 user dating profiles from OKCupid. This dataset includes a combination of structured and unstructured data of categorical user features, and essay prompt answers respectively. OKCupid focuses on making meaningful connections by curating detailed user profiles from an extensive survey, hence the aim of the project was to understand the crux of the workings of cosine similarity on the combination of two types of data to be vectorised. In addition, given this was the only authentic, publicly available dating dataset, we were also limited to the types of algorithms to use due to the absence of user interaction data - collaborative and content-based filtering require user preferences for matchmaking. Thus, we proceeded with a cold-start problem of fresh user profiles using cosine similarity to attain a successful matchmaking algorithm.

Methodology

Data Processing

The data preprocessing pipeline was designed to ensure consistency, manage missing values, and prepare the features for downstream similarity calculations. Missing categorical attributes were replaced with the placeholder category 'Unknown' as we aimed not to falsify information when trying to match people on similarity; this would simply group users with no answers to these questions which would standardise missing values across the board. The missing numerical attributes such as income were imputed with median values to maintain statistical consistency, whilst the missing essay responses were replaced with an empty string before embedding transformation.



The matrix visualization shows the correlation between missing features, with only smoking and drinking standing out.

Categorical data required different encoding techniques based on the number of unique values (cardinality). Low-cardinality features (e.g., sex, orientation, age) were one-hot encoded to preserve interpretability. High-cardinality features (e.g., job, diet, ethnicity) were bucketed into fewer representative categories to reduce dimensionality while retaining meaningful distinctions. The religion feature began with 45 categories that described devoutness to a religion, so we simplified it to only require the name of the Religion. This reduced the cardinality, and Religion was left with 9 categories, which is much nicer to work with. After these transformations, many features could not be label encoded, based on the ordered categories, while nominal data was one-hot encoded. Multi-label categorical variables (e.g., languages spoken, offspring and pets) were binarized, creating separate columns for a reduced number of possible categories. For example, for languages spoken, the top five most common languages were chosen and binarized along with their fluencies to serve as significant features. All numerical features (e.g., age, height, income) were standardized using Min-Max Scaling to ensure that they contributed proportionally to similarity computations.

Additionally, the dataset included 10 optional essay prompts, forming the unstructured text component of each user's profile. To integrate this data effectively, essays were concatenated into a single text field per user, where sentences were tokenized and converted into dense vector representations using DistilBERT, a lightweight transformer-based embedding model. The final step was a concatenated vector composed of the structured categorical and numerical features, and the DistilBERT embeddings.

Model Development

Our analysis began with the Cosine Similarity model, which uses L2-normalized feature vectors to measure similarity through inner products, effectively ranking user profiles based on their proximity in the feature space. While useful, this method proved computationally intensive for our dataset of nearly 60,000 profiles.

To address this, we adopted FAISS, an open-source library by Facebook AI Research, which enables fast approximate nearest neighbor searches in high-dimensional spaces. By indexing normalized training features, FAISS efficiently retrieves the top-k nearest neighbors for any test profile, significantly reducing computation time and memory usage compared to full pairwise similarity calculations.

We further enhanced this approach with a Weighted FAISS model, assigning varying weights to different feature blocks—essay embeddings, categorical features, and numerical features. Through a grid search and 5-fold cross-validation, we identified optimal weights: 2.0 for essay embeddings, and 0.5 for both categorical and numerical features. This weighting underscores the greater discriminative power of essay content, which often reflects personality and preferences, over other feature types.

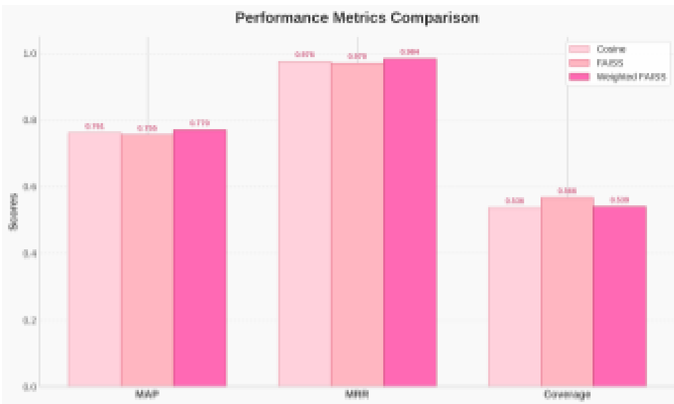
Evaluation Metrics

To evaluate the effectiveness of our models, we focused on ranking-based metrics that provide insight into both the relevance and the ordering of recommendations. We selected **Mean Average Precision (MAP)** as our primary metric because it calculates the average precision across users while considering the entire ranking order of relevant matches. This metric is particularly useful in scenarios where the position of a match in the recommendation list is critical. Additionally, we used **Mean Reciprocal Rank (MRR)**, which measures how quickly the first relevant match appears in the ranking, thereby emphasizing early user satisfaction. **Coverage** was also incorporated as a metric to quantify the proportion of unique profiles recommended across all users, ensuring that the system maintains diversity and avoids repeatedly suggesting the same profiles.

Although metrics like Precision@K and Recall@K are often used in recommendation systems, we opted not to use them as our primary metrics because they provide only a snapshot of the top-k recommendations without accounting for the entire ranking order. Our focus on MAP, MRR, and Coverage offers a more comprehensive assessment of the model's performance.

Analysis

Below are model performance results obtained from experimenting with the three similarity models.



This bar chart compares the three models performance, there is no significant leader

Metrics	Cosine Similarity	FAISS (Baseline)	FAISS (Weighted)
MAP	0.7608	0.7550	**0.7699**
MRR	0.9755	0.9703	**0.9838**
Coverage	0.5364	**0.5663**	0.5392

The cosine similarity technique was implemented first, followed by the baseline FAISS and weighted FAISS respectively. It is clear that when equipped with a more efficient similarity search tool like FAISS, the search performance across around 60,000 user profiles increases modestly. The baseline FAISS shows marginal improvements in MAP and coverage than cosine similarity, whereas the weighted FAISS has higher MRR and MAP in comparison to both models. While the FAISS baseline shows higher coverage (0.5663), the weighted FAISS model achieves balanced performance by slightly reducing coverage to 0.5392—this is acceptable given the improved ranking quality. The latter model also has the best MRR score, suggesting it excels at ranking the most relevant matches at the top. In this case, although the baseline FAISS obtains higher performance scores, the weighted FAISS was deemed superior as the priority is to rank more relevant matches higher up in the user's recommendation list of profiles; users typically pay most attention to the first few profiles they see. This increases the likelihood of successful matching and positive user experience.

Recommendations

The analysis confirms that our optimized Weighted FAISS model offers a superior balance between ranking accuracy and computational efficiency. The emphasis on essay embeddings – by doubling their weight relative to other features – underscores the value of rich textual data in understanding user compatibility. However, the slight reduction in Coverage suggests that while the top recommendations are highly relevant, there is a trade-off in terms of recommendation diversity. Future work might explore dynamic weighting strategies that adjust based on user feedback or integrate additional signals, such as implicit interaction data, to further enhance both the relevance and diversity of recommendations.

Furthermore, while our chosen evaluation metrics (MAP, MRR, and Coverage) provided a robust framework for assessing performance, incorporating metrics like NDCG, novelty, or even user-centric measures could offer additional insights into the quality of the recommendations from a business perspective.

Conclusion

In "Match Made in ML," we demonstrated a practical, efficient approach to matchmaking using FAISS with cosine similarity, enhanced by a weighted feature combination optimized through grid search and cross-validation. Our experiments indicate that by prioritizing essay embeddings, our Weighted FAISS model outperforms baseline methods in key ranking metrics (MAP and MRR) while maintaining reasonable diversity. Through this project, we realised the importance of achieving balance between ranking quality, diversity and efficiency, as well as the fact that choosing the best model also depended on the project priority. Realistically, when a user is shown potential matches, the order matters significantly, meaning if highly compatible matches appear lower in the list, users might stop scrolling before reaching them, were this algorithm implemented in a dating application. This led to the conclusion that weighted FAISS is the best model to leverage, also having feature-engineered user profile features to weigh personality-enhancing features higher. This project highlights the potential of leveraging advanced similarity search techniques in the competitive space of online dating with a focus on most relevant matches ranking higher, paving the way for more intelligent and personalized matchmaking systems.