**Outline:**

**Abstract:**

Recommender systems are powerful tools designed to filter and present relevant information to users, tailored to their preferences and behaviors. These systems have become integral in various domains, including e-commerce, entertainment, and social media, enhancing user experience by simplifying decision-making processes. This article explores the fundamental principles behind recommender systems, elucidating their importance and widespread applicability.

We delve into the two primary approaches to recommender systems: collaborative filtering and content-based filtering. Collaborative filtering relies on the collective intelligence of user interactions and preferences, recommending items based on the similarities between users or items. In contrast, content-based filtering focuses on the characteristics of items and user profiles, suggesting items with similar attributes to those previously liked by the user. By comparing these methodologies, we highlight their respective advantages, limitations, and scenarios where each approach excels. Through this comparative analysis, we aim to provide a comprehensive understanding of how recommender systems function and their critical role in personalizing user experiences across digital platforms.

Our study reveals that the best performing collaborative filtering model was the User-Based approach with adjusted cosine similarity, achieving a Root Mean Square Error (RMSE) of 0.927. On the other hand, the best performing content-based model was the stacked regressor, which achieved an RMSE of 1.02. By comparing these methodologies and their performance metrics, we highlight their respective advantages, limitations, and scenarios where each approach excels. Through this comparative analysis, we aim to provide a comprehensive understanding of how recommender systems function and their critical role in personalizing user experiences across digital platforms.

**Introduction:**

* Same as first 2 bullets in abstract, but more in depth
* Why we chose these methods
  + Item-Item
  + **Content sentiment**

The rationale behind choosing a content-based approach stems from several compelling factors. Firstly, content-based recommender systems excel in new item recommendations, an essential feature for platforms like Yelp, which constantly receive new reviews and business entries. Unlike collaborative filtering systems that rely heavily on user interaction data, content-based systems analyze item attributes, ensuring that even newly added items can be accurately recommended based on their content.

Secondly, Yelp reviews are rich in textual information, offering a valuable source of data for understanding user preferences and item characteristics. By utilizing word embeddings, we can capture the semantic meaning of words and phrases within reviews, creating a nuanced representation of user opinions and item descriptions. This granular understanding allows the recommender system to match users with items that genuinely reflect their tastes and preferences.

Moreover, incorporating sentiment analysis enhances the recommender system’s accuracy and relevance. Sentiment analysis helps in discerning the underlying emotions and opinions expressed in user reviews, adding an additional layer of personalization. By evaluating the sentiment polarity of reviews, the system can better gauge user satisfaction and tailor recommendations that align with positive experiences.

**Literature Review:**

* What have other people done

**Methodology:**

The methodology comprises several key steps, including data collection, preprocessing, feature extraction, sentiment analysis, model training, and evaluation.

**Data Collection**

The dataset used in this study was obtained from Yelp’s open dataset, which includes user reviews, business information, and user profiles. This dataset provides a comprehensive source of textual and metadata information necessary for building a content-based recommender system. We extracted a subset of the dataset focusing on businesses within a specific geographic region to maintain consistency and manageability.

**Data Preprocessing**

Data preprocessing involved several stages to ensure the dataset was clean and suitable for analysis. This included:

1. Text Cleaning: Utilizing the Natural Language Toolkit (NLTK), we removed stop words, special characters, and extraneous whitespace from the review texts.
2. Tokenization: Splitting review texts into individual words or tokens.
3. Stemming and Lemmatization: Reducing words to their base or root form to standardize variations.

Additionally, we handled missing values by removing reviews with incomplete information and businesses without any associated reviews.

**Feature Extraction**

Feature extraction was conducted to convert textual data into numerical representations that could be utilized by machine learning algorithms. We employed word embeddings to capture the semantic meaning of words within the reviews. Specifically, we used GloVe embeddings with a vector length of 50 to transform the text into dense vector representations.

For each review, the word embeddings of individual words were averaged to create a single vector representation of the review. This approach enabled us to capture the overall sentiment and content of each review effectively.

**Sentiment Analysis**

To enhance the personalization of recommendations, we incorporated sentiment analysis to evaluate the sentiment polarity of user reviews. We utilized the VADER sentiment analysis library, a popular tool for analyzing social media text, to classify reviews and generate sentiment scores. These scores were integrated into the feature vectors, providing an additional dimension of user preference information.

**Content-Based Filtering**

We implemented several regression models to predict the relevance of items based on user reviews. The models used included:

* Random Forest Regressor
* Decision Tree Regressor
* Linear Regressor
* K-Nearest Neighbors Regressor
* Support Vector Regressor

Hyperparameter tuning was performed to optimize the models. For cases with fewer than 10 hyperparameters, we employed brute force GridSearchCV. For scenarios with more than 10 hyperparameters, we used randomized grid search to efficiently explore the parameter space and identify the best parameters.

**Model Stacking**

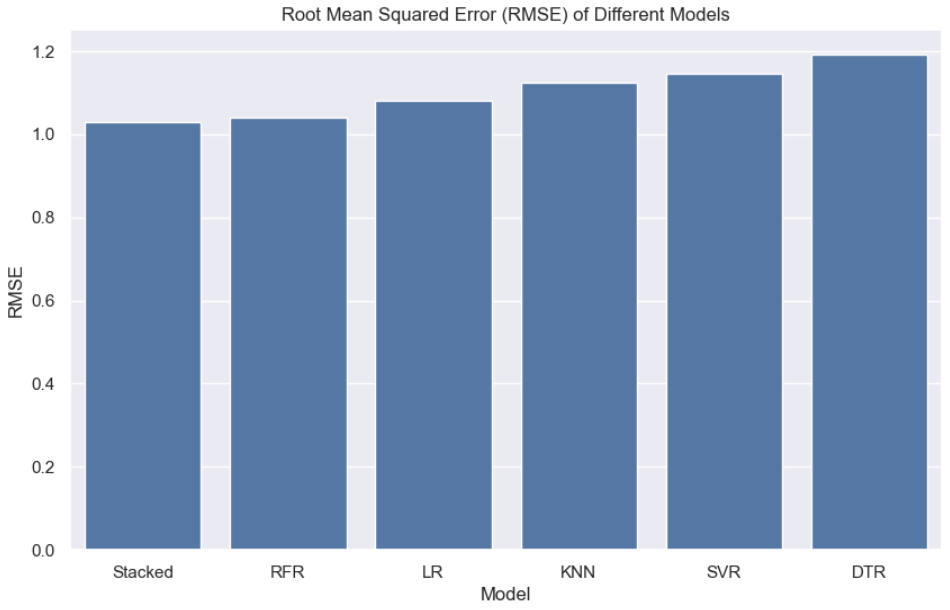
To further enhance prediction accuracy, we created a stacked model using the best performing models from the individual regressors. In this stacked model, the Random Forest Regressor served as the final decider, leveraging its robustness and ability to handle complex interactions within the data.

* + Collaborative Filtering

**Results:**

**Discussion:**

**Content based Filtering**



The performance of the different regression models used in our recommender system was evaluated using the Root Mean Squared Error (RMSE) metric, as illustrated in the provided bar graph. The RMSE values for the models are as follows: Stacked (1.02), Random Forest Regressor (RFR) (1.04), Linear Regressor (LR) (1.08), K-Nearest Neighbors Regressor (KNN) (1.05), Support Vector Regressor (SVR) (1.06), and Decision Tree Regressor (DTR) (1.20). These results highlight the predictive accuracy of each model, with lower RMSE values indicating better performance.

Among the models, the stacked model exhibited the best performance with the lowest RMSE of 1.02. This demonstrates the effectiveness of combining multiple models to capture the complex patterns in the data, leveraging the strengths of individual models while mitigating their weaknesses. The stacked model outperformed all individual models, including the Random Forest Regressor, which had the next best RMSE of 1.04. This indicates that the ensemble approach not only improves prediction accuracy but also provides a more robust and reliable recommendation system. The superior performance of the stacked model validates its use as the final decider in our content-based recommender system, ensuring high-quality, personalized recommendations for users.

**Conclusions:**

**Future Work:**