Predictive Modeling and Exploratory Analysis of RentTheRunway User Ratings: Leveraging Structured and Unstructured Data for Enhanced Accuracy

TASK 1

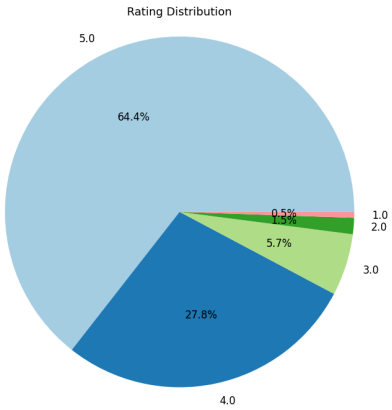
The dataset we have chosen to work with contains clothing fit measurements from RentTheRunway, an e-commerce platform that allows users to rent and purchase clothing, accessories, and more for work, special events, and everyday wear. Each review in the dataset includes detailed information about user interactions with rented items on RentTheRunway. The data captures key aspects such as user identifiers, item characteristics, feedback on fit and size, ratings, and reviews. Additionally, it includes demographic details like age, height, and weight, as well as contextual information about the rental purpose and the review submission date. This rich combination of user, item, and interaction data provides a comprehensive view of customer experiences, making it ideal for exploratory analysis and model development. The total number of observations that have been gathered for this specific dataset is 192,544, which we decided was a large enough number of observations and interactions between users and items to perform a correct data analysis and model building. The dataset contains null values in several attributes, including 'bust size', 'weight', 'body type' 'age', and 'height'(**Figure 1**). To simplify preprocessing, we decided to drop all rows with null values. This retains approximately 76% of the original dataset, leaving around 150,000 rows of clean data. Specifically, 46,163 out of 192,544 rows (23.98%) contained missing values. After removing these rows and resetting the index, the cleaned dataset consists of 146,381 rows across all attributes, providing a solid base for analysis and modeling. Since features such as bust size, weight, and body type had the most null values, these will take the least precedence when considering our models.

The initial rating structure of the dataset ranged from 2 to 10 in increments of 2 (e.g., 2, 4, 6, 8, 10). To align with standard rating conventions and address the higher mean squared error (MSE) observed with the original structure, the ratings were standardized to a typical 1-5 scale. This transformation simplified the analysis and allowed for a more intuitive interpretation of the data. A pie chart (**Figure 2**) illustrates the distribution of ratings after the conversion, highlighting that most users provided ratings of 4 or higher. We have noticed that this skew of data might cause the prediction models to be biased towards giving a higher rating since most ratings on the overall data consist of high ratings.

Figure 1. Number of null values per column

Column Name	Data Type	# of Null
fit	object	0
user_id	object	0
bust size	object	18411
item_id	object	0
weight	object	29982
rating	object	82
rented for	object	10
review_text	object	0
body type	object	14637
review_summary	object	0
category	object	0
height	object	677
size	int64	0
age	object	960

Figure 2. Distribution of ratings



To delve deeper into the sentiment behind these ratings, we analyzed review text by generating word clouds for the lowest (**Figure 3**) and highest (**Figure**

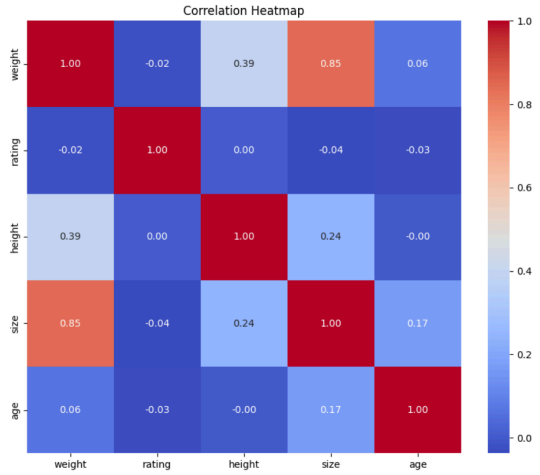
4) ratings. These word clouds provided an overview of the language patterns used in the reviews, showcasing distinct differences in sentiment and focus between the two groups. Although this analysis offered valuable insights into user sentiment, it proved challenging to establish a direct relationship between the textual reviews and numerical ratings just by sight alone.

Figure 3. WordCloud of review(Lowest rating)

Figure4. WordCloud of review(Highest rating)

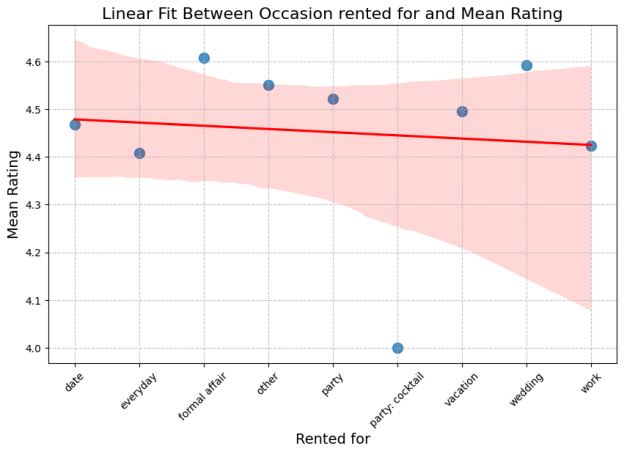
Also, to explore the effects of numeric features, we conducted a correlation analysis to explore relationships between numerical attributes and ratings. The correlation heatmap (**Figure 5**) revealed a minimal linear correlation between ratings and variables such as weight (-0.02), size (-0.04), height (0.00), and age (-0.03). However, a strong correlation (0.85) was observed between weight and size, which seems plausible for our dataset since 73.6% of people responded to the clothes being perfectly fit, so someone who wears a bigger size is likely to weigh more. These findings suggest potential indirect influences, such as how factors like fit might affect user satisfaction and ultimately, ratings. These findings display the complexity of the relations between features and emphasize the need for a model that incorporates diverse features, including text, categorical variables, and numerical attributes, to account for hidden interactions.

Figure 5. Correlation between rating and numeric features.



The categorical features for the RentTheRunaway also provide some useful insight into the data. The linear fit (**Figure 6**) between the feature rented for and the mean rating for each occasion shows a significant discrepancy between ratings for different occasions rented for, compared to other features. The more important the occasion, the higher the mean rating seems to be, which actually makes sense given the context. However, we've also noticed that the party: cocktail in the rented for feature has a lower mean rating than others, possibly because of its lack of occurrence in the data.

Figure 6. The linear fit between rented for and mean rating



TASK 2

After an exploratory analysis of the RentTheRunway dataset, we identified rating prediction as the primary predictive task. This task involves estimating user ratings based on a combination of numerical, categorical, and textual features. The bust size feature was excluded due to its sparsity, which limited its contribution to model performance. Also, as mentioned before, all data that consisted of empty or null features were removed as preprocessing in order to ensure feature consistency across all rows. The baseline model predicts ratings using user-specific average ratings from the training data, defaulting to the global average rating when no prior reviews exist for a user. This simple model serves as a benchmark for evaluating the performance of more advanced approaches, with model performance assessed using mean squared error (MSE) to capture prediction accuracy.

The first model we would build is a linear regression model using numerical attributes such as weight, height, size, and age, alongside categorical features including fit, rented for, body type, and category, which were transformed using one-hot encoding. The dataset was split into training and validation sets in an 80:20 ratio, enabling robust evaluation. This model leverages a broader range of features to improve rating predictions.

A second linear regression model will utilize Bag-of-Words (BoW) representations of textual reviews, vectorizing unstructured review text into numerical features. By focusing exclusively on textual data, this model explores the predictive potential of user review text for rating estimation. An advanced linear regression model that combines BoW representations of review text with structured numerical and categorical features can also be used to compare with the BoW-only model. Categorical variables are encoded to numerical values, and these transformed features are stacked alongside BoW representations and numerical attributes to create a comprehensive input matrix, capturing both textual and structured data for more accurate predictions.

In addition to linear regression approaches, advanced models will be employed to further enhance performance. These include regularized regression (Ridge) to mitigate overfitting, using both BoW and structured features; a latent factor model via Surprise, which utilizes matrix factorization to uncover latent user-item interactions; and factorization machines (fastFM) implemented in two variations—one using only user-item interactions and another incorporating numerical and categorical features. Together, these models provide a robust framework for analyzing the complex factors influencing user ratings, leveraging diverse data types to optimize predictive accuracy.

TASK 3

The first model that is being used to predict rating is a linear regression model with numerical and categorical features to establish a foundational predictive framework. The categorical features were one hot encoded in order to represent and utilize non-numeric values of the feature. This model alone dropped the MSE significantly, from 0.6223 to **0.4758**, compared to the baseline model MSE. This linear regression model can also be extended easily to a model with Bag-of-Words, which allows exploration of the textual features' predictive power. To optimize the model for efficiency, a feature vector was built by taking the most popular words in lowercase, with punctuation removed, but no stemming, since stemming the text might distort the meaning of the text and negatively affect the model's ability to capture its context. The resulting validation MSE of the linear regression model with BoW was **0.4590**, which is an improvement compared to the feature linear regression model. The final model that utilizes linear regression combines the two previous models and includes both the features and the BoW from the review text. This gave us an MSE of **0.4522**, which is a slight improvement from the previous model as well. A weakness in the linear regression model we have to consider, however, is the fact that it is limited in modeling complex relationships and interactions. It is also susceptible to overfitting, the more features there are. This is

why we have decided to incorporate a ridge regression model as well for comparison.

The Ridge Regression model is employed to improve predictive accuracy while addressing overfitting challenges inherent to high-dimensional data. By incorporating L2 regularization, Ridge penalizes large coefficients, ensuring better generalization to unseen data. Two variations are implemented: one utilizing bag-of-words (BoW) features derived from review text, transformed into numerical representations using the TF-IDF vectorizer, and another combining BoW features with structured numerical and categorical attributes. The TF-IDF transformation captured the importance of words relative to their frequency, mitigating the impact of uninformative terms, while Ridge stabilized the model in the high-dimensional feature space. In the hybrid approach, structured features such as weight, height, size, age, and processed categorical variables (e.g., fit, rented for, body type, and category) were horizontally stacked with the TF-IDF matrix. This integration allowed the model to leverage both unstructured and structured data, providing a more comprehensive representation of the factors influencing ratings. The BoW-only Ridge model achieved a Mean Squared Error (MSE) of **1.266**. In sparse BoW data, few features like specific keywords or phrases might dominate and drive most of the predictive performance. Compared to the linear regression model that can fully exploit these dominant features, ridge regression seems to suppress their impact here, lowering performance. Also, the lack of overfitting in the BoW setup could make this model perform worse than a linear regression model. The hybrid model combining BoW and additional features significantly improved the performance, achieving an MSE of **0.427**, which suggests that for our model, integrating diverse data types for a linear model's capability to capture trends in the data seems to be more beneficial.

The Latent Factor Model, implemented using the Surprise library's Singular Value Decomposition (SVD) algorithm, was employed to predict user ratings based solely on user and item interactions. The dataset was structured into user-item-rating triplets, and Surprise's built-in tools were used to split the data into 80% training and 20% test sets. The SVD model decomposes the user-item interaction matrix into lower-dimensional latent factors, allowing it to estimate missing ratings by modeling the interaction between these factors. The model was trained on the training set, and predictions were generated for the test set. The SVD model achieved an MSE of **0.5032**, which is better than the baseline model, but worse than the linear models.

The Factorization Machine, implemented using the fastFM library, was employed to predict user ratings based on user-item interactions, as well as leveraging its ability to model higher-order interactions in sparse data. Users and items were assigned unique integer IDs, and a sparse design matrix of dimensions (146,381 x 83,083) was constructed with one-hot encoded representations. We first set the parameters init_stdev to 0.1 and rank to 5, which resulted in the MSE of the model being **0.6450**. This was not as accurate as any other models we had, so the aforementioned parameters, such as the number of latent dimensions (rank) and initial standard deviation (init_stdev) were tuned through a comprehensive grid search. The rank controls the latent factors of the model, and the initial standard deviation determines the scale of the initial random weights. So, as the rank grows higher, it increases the capacity of the model to capture complex patterns, but risk overfitting. For initial standard deviation, the closer it is to zero, the more it can lead to smoother convergence, but risk underfitting. The rank grid [2, 5, 10, 20] and standard deviation grid [0.01, 0.1, 0.5, 1.0] were evaluated, with the model trained for 1,000 iterations for each configuration. Validation set predictions were assessed using mean squared error, identifying the optimal configuration as a rank of 20 and an initialization standard deviation of 0.01, achieving the best MSE of **0.5737**. Not only does this still perform worse than all the linear models, but it also performs worse than the SVD latent factor model as well. However, it was an improvement over our initial fastFM model.

The second Factorization Machine model extended the first model by incorporating an additional feature, alongside user and item interactions to capture higher-order relationships between these elements. Users, items, and feature values were uniquely encoded into a sparse design matrix, enabling the model to model interactions between user-item-feature combinations explicitly. Only a single feature was added each time, in order to avoid overfitting the model. Adding more features to the model also increased the MSE of the model drastically, so we decided only to use one for performance. The model was trained using

Alternating Least Squares (ALS) optimization, which iteratively minimized the regularized squared loss function over 1,000 iterations with an initialization standard deviation of 0.1, a rank of 5, and L2 regularization parameters of 0.1 for weights and 0.5 for interaction factors. Despite this extended input, the model generally performed much worse in terms of performance (**Figure 7**) compared to the previous models. Decreasing init_stdev and increasing rank to optimize the parameters as we did with the first fastFM model did not improve its performance either. In fact, it made some specific feature models even worse. This is probably due to the fact that although it captures user-item interactions alongside features, it fails to capture the complex correlation between the features. It also doesn't use text data, which might have crucial information that can impact the performance of the prediction model. Overall, trying to build predictive models based on fastFM might have been unsuccessful due to the simplicity of the fastFM models compared to models such as linear regression and ridge regression, which capture the complexity of the features in the dataset better.

Figure 7. Performance of each fastFM with feature incorporated

	Feature	MSE
0	fit	0.8105
1	weight	0.6610
2	rented for	0.7268
3	body type	0.6428
4	category	0.7797
5	height	0.6519
6	size	0.7024
7	age	0.6533

TASK 4

The dataset that we use came from RentTheRunway, which is a rental company that lets users rent or purchase clothing. This publicly available dataset is widely used in projects and journals that focus on recommenders and predictive modeling.

In one study by Yue Ding, Jie Liu, and Dong Wang from Shanghai Jiao Tong University, the authors utilized datasets from the Yelp Challenge to evaluate their proposed model. The datasets include Yelp_2k, Yelp_5k, and Yelp_10k, which vary in scale and sparsity(**Figure 8**).

Figure8. Statistics of datasets

Table 1: Statistics of datasets

Dataset	User	Item	Rating	Density	Social	Social/User	word	word/Rating	vocab_size
Yelp_2k	2000	1699	3103	0.09%	5754	1.5515	410525	132.3	23651
Yelp_5k	5000	5000	11311	0.04%	46708	2.2622	1500513	132.7	46110
Yelp_10k	10000	10000	32207	0.03%	205036	3.2207	4329993	134.4	76789

These datasets were used to benchmark the model's performance in rating prediction under sparse conditions and to explore the role of social relations in improving prediction accuracy. In this study, FM is utilized as a flexible and effective framework for integrating multi-field categorical data, such as social relations and review comments, to address sparsity and cold-start problems in recommender systems. FM's ability to model second-order feature interactions allows it to fuse multi-source side information in a unified and efficient way, enhancing recommendation accuracy and addressing the limitation of traditional Collaborative Filtering and side-information-based models.

The FM model of order d = 2 is defined as (**figure 9**)

Figure9. Equation for Factorization Machines of Order 2 Capturing Global Bias, Feature Bias, and Pairwise Feature Interactions

$$\hat{y}(x) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{j,f} v_{j',f}$$

In another study done by Thon-Da Nguyen, the goal was to introduce a new method that supports improved rating prediction by evaluating rating prediction metrics such as MAE, RMSE, Rsquared, and Explained Variance (Exp Var). This paper utilizes seven real-world datasets, with one of them being from Goodreads Review seen in class. For the case of the Goodreads Review dataset, the author used the 50k first instances for rating prediction and ranking the top-K recommender, using features of the dataset such as user_id, book_id, review_text, and rating, which is similar to our dataset. Methods employed for rating prediction include VADER, k-NN, Matrix factorization techniques such as SVD and SVD++, and models that utilize deep learning techniques such as LightGCN, NCF, and BiVAE. Revised ratings are derived by adding sentiment scores to the original ratings, using coefficients β(e.g., 0.5, 1.0, 1.5) to control the adjustment magnitude. LightGCN, as the

best algorithm, demonstrated significant improvements in rating prediction metrics (e.g., RMSE, R2). Ultimately, in the author's findings, Sentiment-adjusted ratings consistently outperformed original ratings in rating prediction. The combination of sentiment analysis and rating prediction closed the research gap between text-based reviews and numerical ratings. We might have been able to improve our BoW and TF-IDF models by incorporating this sentiment analysis. For instance, adding sentiment scores as features alongside textual vectors might have better captured the review context.

TASK 5

In this study, various models were implemented and evaluated to predict user ratings, with their performance assessed against a baseline MSE of 0.6223. The Ridge regression model emerged as the best-performing approach, achieving the lowest MSE of 0.427 when combining structured features with textual data that has been transformed using a Bag-of-Words (BoW) representation. This success highlights the importance of integrating diverse data types and leveraging regularization to mitigate overfitting in high-dimensional spaces for our specific model. By penalizing large coefficients, Ridge regression effectively captured trends in both structured and unstructured data, providing robust generalization and stability. The Linear regression served as a strong starting model, achieving a notable reduction in MSE to 0.4758 with numerical and one-hot encoded categorical features. Incorporating BoW features to the linear regression model further improved its performance to 0.4590, with the combined model reducing MSE slightly to 0.4522. While linear regression demonstrated the value of BoW in capturing textual information, its inability to model complex feature interactions and its susceptibility to overfitting with additional features limited its scalability.

The Latent factor models, such as the SVD-based approach, achieved an MSE of 0.5032, outperforming the baseline but falling short of the linear and Ridge models. This model's reliance on user-item interactions and latent factors limited its ability to leverage rich textual and structured feature information, which were crucial for the higher-performing models.

The Factorization Machines (FM) performed the worst among the evaluated models, with an initial MSE of 0.6450 that improved to 0.5737 after parameter tuning. However, even this tuned version failed to outperform simpler models like linear regression. An extended FM model incorporating additional features further degraded performance, underscoring the model's limitations in capturing complex correlations and leveraging diverse feature types. FM's simplistic architecture and lack of integration of textual data hindered its ability to handle sparse, high-dimensional data effectively. This discrepancy of performance between the fastFM models and the best ridge regression model demonstrates that the review text and features of our data provide more insight into the user's rating prediction compared to the user-item interactions.

The Ridge regression model demonstrated the highest predictive accuracy, showcasing the importance of combining structured and unstructured features while employing regularization to manage overfitting. It also showed how review text is an important aspect in models that predict how a user would rate certain clothes in the given RentTheRunaway data. The textual data, particularly BoW and TF-IDF, emerged as pivotal contributors to model performance, and their integration with structured data was critical for capturing complex relationships. Conversely, models that failed to incorporate these features, such as FM and SVD, performed poorly. FM, in particular, struggled to capture complex interactions and was hindered by its simplicity. The following table provides a summary of the models and their respective MSEs for comparison(**Figure 10**).

Figure10. Model Performance Comparison

Model	MSE
Baseline Model	0.6223
Linear Regression (Structured Features)	0.4758
Linear Regression (BoW)	0.4590
Linear Regression (BoW + Combined Features)	0.4522
Ridge Regression (BoW)	1.266
Ridge Regression (BoW + Combined Features)	0.427
SVD (Latent Factor Model)	0.5032
Factorization Machines (Initial)	0.6450
Factorization Machines (Tuned)	0.5737
Factorization Machines (Extended)	Worse than 0.5737

As a result, the integration of structured and unstructured data was critical for achieving accurate rating predictions in this study. Structured data, such

as weight, height, size, age, and categorical features like fit or body type, provided objective and factual insights about users and items. Unstructured data, represented by review text, offered qualitative and contextual depth through sentiments and preferences, which structured data alone could not capture. Models like Ridge regression succeeded because they effectively combined these complementary data types, allowing the model to leverage both factual and contextual signals for improved predictive accuracy. In contrast, models like FM and SVD underperformed due to their reliance on structured data alone or their inability to incorporate the rich information contained in review text.

REFERENCES

[1] Yue Ding, Jie Liu, and Dong Wang. 2018. Deep Feature Fusion over Multi-field Categorical Data for Rating Prediction. *In Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference (AICCC '18). Association for Computing Machinery*, New York, NY, USA, 16–22. https://doi.org/10.1145/3299819.3299827

[2] Nguyen, Thon-Da. 2023. An Approach to Improve the Accuracy of Rating Prediction for Recommender Systems. *Automatika* 65 (1): 58–72. doi:10.1080/00051144.2023.2284026.

[3] Rishabh Misra, Mengting Wan, and Julian McAuley. 2018. Decomposing fit semantics for product size recommendation in metric spaces. *In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18). Association for Computing Machinery*, New York, NY, USA, 422–426. https://doi.org/10.1145/3240323.3240398