Beyond Reviews: Predicting Recipe Ratings Using Recipe Attributes and Transformers

By Mohammed Elzubeir and Alexander Caichen

Abstract

This study explores the viability of predicting online recipe ratings using only intrinsic recipe attributes – instructions and nutritional data without relying on user reviews. Leveraging Sentence-BERT embeddings and structured metadata, we evaluate performance across linear regression, random forest, and neural network models. Our results show that while transformer-enhanced features improve prediction accuracy, random forests outperform neural networks in consistency and ease of deployment. However, predictions tend to reflect general trends rather than precise ratings, especially for low-rated recipes, due to data scarcity and noisy embeddings. The findings highlight both the promise and limitations of review-independent rating prediction in recipe recommendation systems.

Introduction

The task of predicting recipe ratings using only the intrinsic attributes of a recipe, such as ingredients, cooking instructions, nutritional content, and preparation time, is both practically relevant and theoretically intriguing. Online recipe platforms, boasting vast numbers of culinary contributions, typically rely on user-generated reviews to surface high-quality recipes. However, the overdependence on reviews poses significant challenges. Firstly, new or obscure recipes without a critical mass of user interactions are often overlooked, irrespective of their intrinsic quality. Secondly, reviews may exhibit biases influenced by external factors such as author popularity, cultural preferences, or the prominence of particular cuisines. Predicting ratings without leveraging user reviews can address these biases, providing a fairer and more consistent evaluation mechanism.

Advances in natural language processing, particularly transformer-based language models and large language models (LLMs), offer promising avenues for overcoming previous limitations. These models excel at capturing semantic nuances and complex interactions within textual data, making them uniquely suited to comprehending the intricacies of recipe instructions and ingredients. Additionally, transformer-based architectures have recently shown considerable potential in integrating structured numerical and textual inputs, potentially surpassing traditional methods relying solely on numerical or categorical data [7].

Our research attempts to address these challenges by exploring the viability of predicting ratings using recipe instruction embeddings generated through cutting-edge transformer-based techniques, specifically Sentence-BERT (SBERT). By circumventing user reviews and leveraging advanced neural architectures, our study aims not only to improve prediction accuracy but also to contribute to broader efforts in reducing biases and enhancing the fairness and discoverability of high-quality recipes. This paper delineates our methodological approach, empirical results, and the implications of these findings for the field of computational culinary analytics.

Background

Early Approaches (2011–2013)

Initial efforts to predict recipe ratings primarily leveraged structured recipe attributes, such as ingredients, cooking methods, and nutritional information. Geleijnse et al. (2011) [1] developed a personalized recommendation system using structured features like key ingredients, noting the feasibility of content-based recommendations despite limited data. Ueda et al. (2011) [14] employed user preferences from historical cooking behaviors to personalize recommendations, highlighting the early promise of structured data.

Contrasting findings emerged shortly thereafter. Teng et al. (2012) [11] demonstrated the predictive power of ingredient networks, achieving 79% accuracy using structured features alone. Conversely, Yu et al. (2013) [15] found that textual reviews significantly outperformed structured attributes like ingredients or instructions for rating predictions, sparking a debate regarding the sufficiency of recipe content alone.

Expansion into Reviews and Collaborative Filtering (2014–2018)

Acknowledging the strength of textual data, Liu et al. (2014) [8] presented a two-stage model to analyze sentiment in reviews, achieving a 65.6% accuracy on classification tasks. This highlighted the nuanced information contained within user reviews that structured attributes alone often miss.

Meanwhile, traditional machine learning methods, including collaborative filtering and hybrid models, gained popularity. Jain et al. [4] explored feature engineering with numerical recipe attributes (e.g., cooking time, number of ingredients), achieving moderate success. They identified contextual influences like recipe age and complexity as important predictors of ratings, but acknowledged inherent dataset biases towards positive ratings.

Rise of Deep Learning and Multimodal Models (2019–2021)

Deep learning models introduced advanced techniques for extracting latent representations from both structured and unstructured data. Sanjo and Katsurai (2017) [10] combined visual and semantic embeddings through convolutional neural networks (CNNs), showing image features significantly improved prediction outcomes. This indicated the benefit of multimodal approaches integrating structured and visual data.

Hensley (2019) [3] investigated transformer-based methods, specifically BERT, comparing it with traditional ingredient embeddings. Surprisingly, one-hot encoding of ingredients outperformed BERT embeddings, suggesting careful feature engineering might sometimes surpass advanced models lacking proper domain adaptation.

Modern Transformer-Based and Graph Approaches (2021–Present)

Recent advancements have increasingly relied on transformer-based architectures and large language models (LLMs), capable of integrating structured numerical data and textual descriptions. Li and

McAuley (2020) [7] highlighted the growing potential of transformer models in recipe-related tasks, given their ability to capture intricate ingredient interactions and recipe semantics.

Graph neural networks (GNNs) like Tian et al.'s Hierarchical Graph Attention Network (HGAT) model (2022) [12] demonstrated superior performance by explicitly modeling relationships among users, recipes, and ingredients. HGAT's sophisticated representation showed marked improvements over traditional collaborative filtering, underscoring the strength of relational modeling in enhancing generalizability, particularly in cold-start scenarios.

Comparative Analysis and Ongoing Challenges

Evaluating model performance across studies reveals discrepancies rooted in varying data scales, evaluation metrics, and methodologies. Early studies often reported accuracy, while recent works favor metrics like RMSE and MAE due to skewed ratings data [4]. Transformer-based models show promise in capturing complex ingredient and instruction relationships but demand considerable data and careful feature adaptation, as indicated by mixed results from Hensley [3].

Predicting recipe ratings using structured attributes and transformers is promising yet challenging. Historical research demonstrates evolving methodologies, from structured attribute reliance to sophisticated transformer and graph-based multimodal models. Despite progress, ongoing issues of data bias, model integration, and generalization persist, offering fertile ground for further exploration and improvement in computational recipe analysis.

Methods

Preprocessing

For this study we use the "recipes" dataset from Kaggle's "Food.com - Recipes and Reviews", which contains instructions, nutrition, image links, cooking/preparation time, review data, and other features from 522517 different recipes. From this dataset we use recipe "metadata", including total cooking time, calories, fat, saturated fat, cholesterol, sodium, carbohydrate, fiber, sugar, and protein, as the first predictor features. Several related studies [11, 15] have incorporated nutritional data into recipe rating prediction, whether as part of an ingredient encoding or as direct input features, achieving moderate gains in precision. Intuitively, recipes high in unhealthy nutrients such as sugar tend to be rated lower. We use a transformer-encoded recipe as the second feature. Finally, the target variable is aggregate recipe rating, ranging from 1-5 in intervals of 0.5 (1, 1.5, 2, … 4.5, 5), regardless of the number of user reviews.

Data preprocessing included basic reformatting (Ex: cleaning up recipe instructions in the format of R vectors) and removing recipes with only a single review. Although single-review recipes make up about 38.9% of reviewed recipes, they introduce strong individual bias. Multiple reviews, even two, provide more reliable aggregate ratings and more accurately reflect general sentiment, critical for training a model to predict how most people would rate a recipe.

Removing single-review and no-review recipes leaves us with a dataset with 167628 samples. More specifically we are left with a dataset with the following rating distributions:

Rating	1	1.5	2	2.5	3	3.5	4	4.5	5
Count	225	76	328	673	2755	3978	16063	34330	109201

Based on Fig. 1, if 2-review recipes are removed too, there will be less than 100 remaining recipes for rating groups 1-2. These groups are essential for training the model to predict lower ratings. To maintain a sufficiently diverse sample population, recipes with such low review counts will be retained.

To prevent prediction skew from distributional bias, we sample exactly 2500 recipes from each rating group. These are then split between the training/validation/testing sets using an 8:1:1 ratio. For rating groups with less than 2500 recipes (1-2.5) the entire group will still be split 8:1:1, with the 80% slice undergoing replacement sampling to obtain sufficient data for the training set. The number 2500 is chosen because it's low enough to accommodate the available data for mid-high rating groups (3-5) while high enough to support model generalization. Overall, exactly 2000 recipes per rating are allotted to the training set, making the total training set size 18000. Ratings 1/1.5/2/2.5 have 23, 8, 33, and 68 samples respectively for each of their validation/testing sets.

Model Prediction

We will be performing regression using our data and thus be using Mean Squared Error (MSE) between predicted and expected ratings as the loss metric. While creating an ordinal classifier is also a valid choice, given how rating values are discrete (1, 1.5, etc.), our goal is to predict ratings close to true ratings rather than exact. Ratings reflect subjective opinions and our model approximates a general public consensus, so a regression model predicting a rating of 3.4 when the true rating is 3.5 is better than a classifier predicting 3 or 4. Additionally, accuracy, unlike MSE, treats all misclassifications equally and offers no hint of how far a prediction is from the true value, making it difficult to judge prediction accuracy. Drawing inspiration from Jain et al. [4], we will use a MSE of 0.805, based on always predicting the average rating of the validation/testing sets, as the baseline. This MSE will be compared to the validation and testing MSEs to provide additional insights on model performance.

We proceed to compare the performance of four groups of predictors: metadata, SBERT ingredient embedding, trained SBERT ingredient embedding, and trained SBERT embedding appended to metadata, across four models: ridge regression, linear regression, random forest, and neural network. We compare the MSEs from using different predictors to determine the effectiveness of instruction embeddings on rating predictions. Different models are used because rating predictions via instruction embeddings is relatively uncharted territory.

SBERT embeddings are chosen over the [CLS] token from normal BERT embeddings for supposedly having better semantic representation at a sentence level, which is important when we want to represent an entire recipe using an embedding. We use Hugging Face's all-MiniLM-L6-v2 sentence transformer model to generate SBERT embeddings. For the "trained SBERT embeddings" we fine-tuned the aforementioned model for 12 epochs using our training set and BatchHardTripletLoss (for efficiency) as the loss function This reduced the triplet loss, which measures how embeddings with similar labels are closer and those with different labels are farther apart, from 4.501 to 4.468.

Linear regression is chosen as one of the models since multiple studies, such as one by Sarah Hensley [3], pointed out linear regression provided the best performance when predicting ratings. Ridge is included alongside linear regression to detect overfitting or collinearity influencing the linear regression model.

Model Specifics

- Each random forest has 200 trees.
- Metadata is standardized before combining with SBERT embeddings to prevent large values (Ex: 1500 calories vs embedding values ranging from -1–1) from skewing model predictions.
- Each neural net is trained for 300 epochs at a learning rate of 0.00003. The one exception is the metadata-only network that is trained at a learning rate of 0.00005.
- For more detailed neural net structures please refer to Fig. 2a-c.

Results and Discussion

Please see Fig. 3 for detailed experimental results.

MSEs from rating prediction using metadata were higher than those from using SBERT embeddings, suggesting that instruction content contributes more to ratings than nutritional or other numerical metadata. This is reasonable, as not everyone prioritizes health, but all users of a recipe read the instructions. Furthermore, embeddings generated by the fine-tuned SBERT yielded lower MSEs than those from the original SBERT model, indicating that the embeddings encode rating-relevant information and that this information can be enhanced through training. Finally, adding metadata to the trained SBERT embeddings further reduced MSEs slightly, suggesting that while instructions carry the most weight, metadata provides complementary information that improves prediction performance.

Comparing model performance, validation and test MSEs are lower for ridge regression than for linear regression, while the opposite trend is observed in training MSEs. This suggests that linear regression overfits certain features in the embeddings, and that ridge regression mitigates this through regularization. Although this overfitting may be due to multicollinearity, something ridge regression is designed to address, we do not have enough data to make a definitive conclusion. Nevertheless, these results open the possibility that instruction embeddings contain either superfluous or noisy information.

Model validation/test MSEs for each feature group follow the ranking: linear regression > ridge regression > neural network > random forest. Except for the metadata feature group, MSE decreases significantly from ridge regression to neural networks (by approximately 0.5), then slightly decreases by about 0.05 for random forests. The better performance of neural networks and random forests compared to ridge regression when using SBERT embeddings suggests a highly nonlinear relationship between embedding data and rating. The consistent, slight advantage random forests have over neural networks provides further evidence that the embedding data contains noisy information, which random forests easily ignore but neural networks cannot.

For the validation and testing sets, only the random forest models trained on embeddings achieved MSE values below the 0.805 baseline. This indicates that, for most models, rating predictions performed worse than simply predicting the average rating for every recipe. Examining the prediction distributions (Fig. 4-7) for each model, excluding random forests, confirms this pattern: the predictions form bell-shaped curves centered around the average rating of the validation and testing sets (3.814). A closer look at the prediction distribution by actual rating (Fig. 8a) explains the elevated MSEs: the broad spread of predictions means that some predictions differ from true ratings by as much as 2 rating points (Ex: predicting 4 when the actual rating is 2), which significantly increases MSE. In contrast, the random forest (Fig. 8b) avoids this issue by making predictions that cluster tightly around the mean, with occasional small shifts that reduce error. Notably, the final neural network trained with both embeddings and metadata (Fig. 8c), which achieved MSEs closest to the random forest, displayed a similar distribution shape: a tower around the average rating. Overall these results imply instruction embeddings do not contain a strong enough predictive signal or they are too noisy, preventing the models from performing significantly better than the baseline. As a side note, we suspect the apparent accuracy for rating 1.5 is a consequence of overfitting, likely due to oversampling a small set of approximately 50 recipes nearly 2,000 times in the training set.

All these pieces of evidence point to the issue of noisy embeddings. Theoretically speaking noisy embeddings is plausible given the diversity of writing styles and the fact most recipes only have single digit numbers of reviews (Fig. 1), forcing models to learn potentially conflicting information (similar embeddings yield dramatically different ratings). Considering this, models may be able to perform better than baseline if provided with more recipes rated by a larger number of users. The increased diversity in opinions from more reviews would help “normalize” rating opinions, allowing the network to discover more consistent patterns between embeddings and ratings.

Conclusion

Overall, the random forest model is recommended for predicting recipe ratings using instruction embeddings, given its consistently low error and minimal need for tuning. Fine-tuned SBERT embeddings have shown to improve predictive capabilities, highlighting the potential for such features. However, instruction embeddings are not yet reliable for accurately predicting ratings, currently offering only marginal gains over simply predicting the average rating. It is also recommended that other features, such as recipe metadata, be used to supplement the embedding data to increase prediction accuracy.

A critical obstacle identified in this study is the lack of sufficient and reliable data for low-rated recipes. Multi-reviewed ratings provide more robust insights than single-review ratings. However, due to online popularity dynamics, lower-rated recipes rarely accumulate substantial reviews. Users predominantly focus on highly-rated recipes, deepening the disparity in available data across rating levels. Future research should therefore emphasize systematic data collection strategies, potentially employing volunteer reviews to obtain balanced and comprehensive datasets, though personal bias may be more apparent using such methods.

Additionally, exploring alternative modeling approaches, such as integrating ingredient-specific predictions, and accessing datasets with continuous rather than discrete rating scales, could significantly advance prediction capabilities. Continued exploration of richer datasets is essential for further progress in the subfield of the literature and in the practical realm of achieving more accurate recipe rating predictions.

Sources:

1. Geleijnse, Gijs & Overbeek, Thé & van der Veeken, Nick & Willemsen, Martijn. (2010). *Extracting Vegetable Information from Recipes to Facilitate Health-Aware Choices.*
2. Harvey et al. (2013). *Cooking with Computers: Recipe recommendation* (personalized using ratings)
3. Hensley, Sarah (2019). *CS229 Project – Recipe Rating Prediction (Natural Language)*.
4. Jain et al. (2018). *Recipe for Success: Predicting Recipe Rating*.
5. Khan et al. (2021). *An Intelligent Approach for Food Recipe Rating Prediction Using Machine Learning*.
6. Li et al. (2021). *SHARE: A System for Hierarchical Assistive Recipe Editing*.
7. Li & McAuley (2020). *Recipes for Success: Data Science in the Home Kitchen* (Harvard Data Sci. Review).
8. Liu et al. (2014). *"My Curiosity was Satisfied, but not in a Good Way": Predicting User Ratings for Online Recipes*.
9. Majumder, Li, Ni, & McAuley (2019). *Generating Personalized Recipes from Historical User Preferences*. (cited in Hensley 2019)
10. Sanjo & Katsurai (2017). *Recipe Popularity Prediction with Deep Visual-Semantic Fusion*.
11. Teng, Lin, & Adamic (2012). *Recipe recommendation using ingredient networks*.
12. Tian et al. (2022). *Recipe Recommendation with Hierarchical Graph Attention Network*.
13. Van Pinxteren, Geleijnse, & Kamsteeg (2011). *Deriving a recipe similarity measure for recommending healthful meals*.
14. Ueda, Takahata, & Nakajima (2011). *User's food preference extraction for cooking recipe recommendation*.
15. Yu et al. (2013). *Do good recipes need butter? Predicting user ratings of online recipes*.

Appendix

Figure 1: Recipe distributions by Rating after removing all single-review recipes.

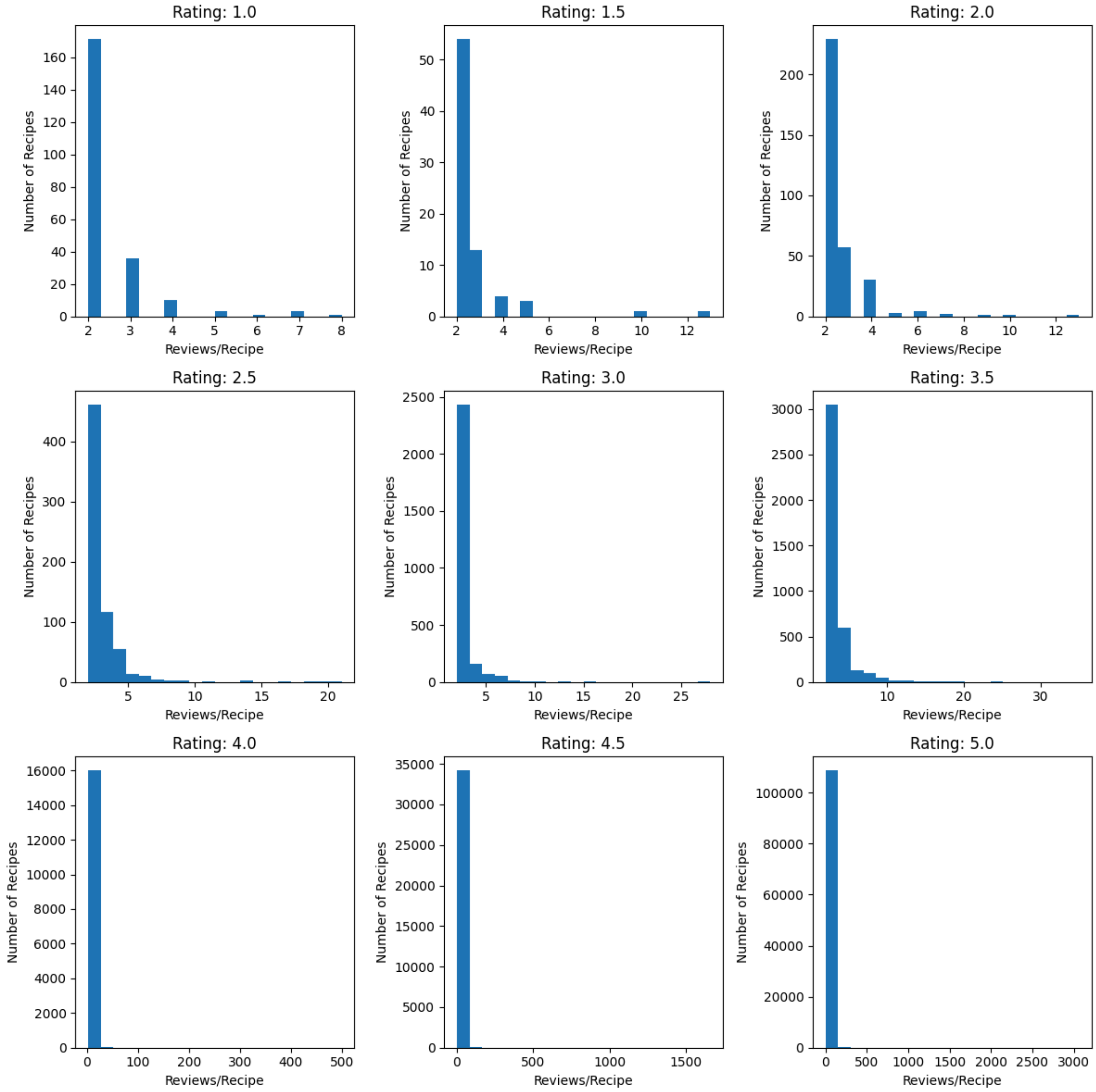


Figure 2a: Neural net for metadata.

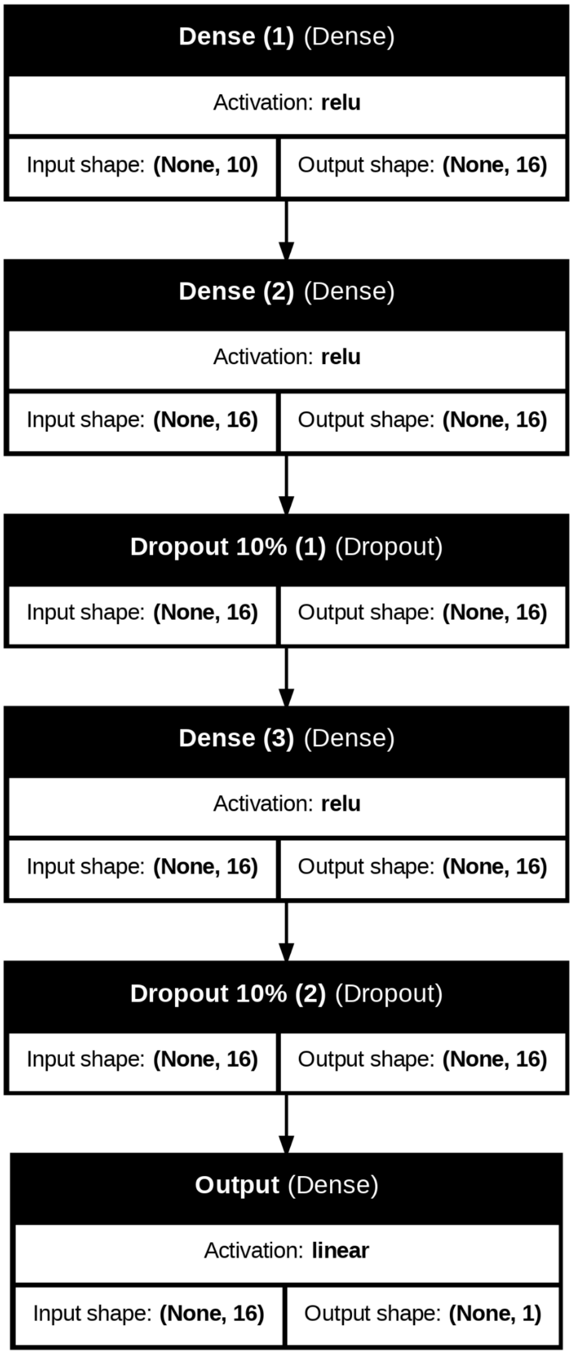


Figure 2b: Neural net for trained/untrained SBERT embeddings.

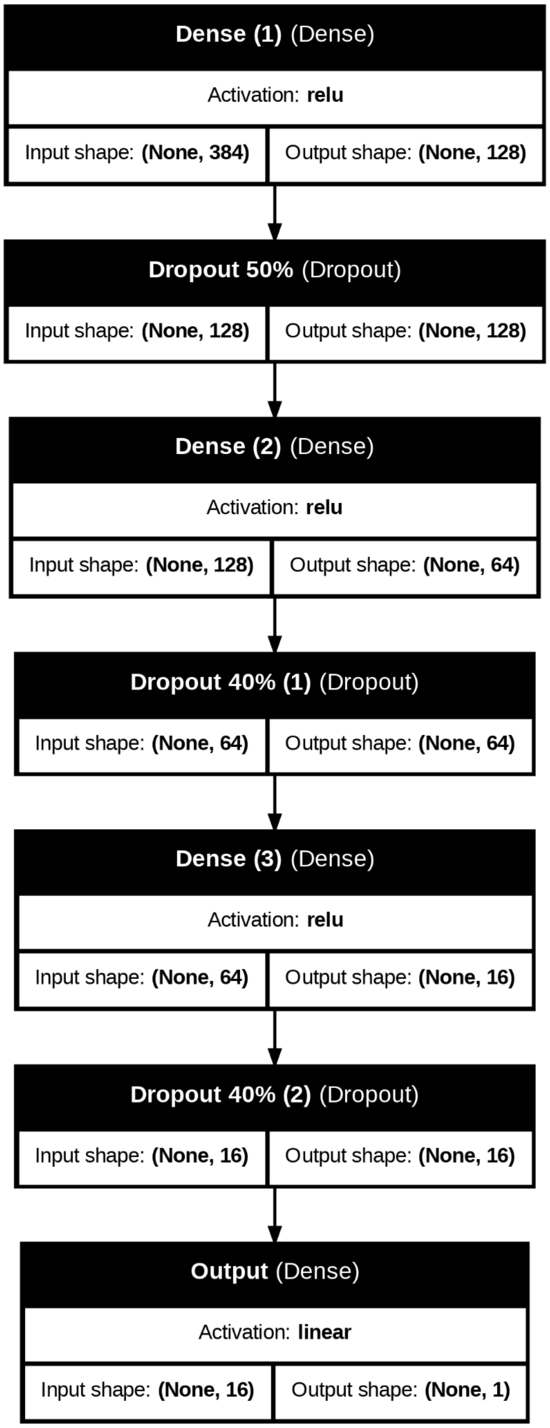


Figure 2c: Neural net for metadata + trained SBERT embeddings.

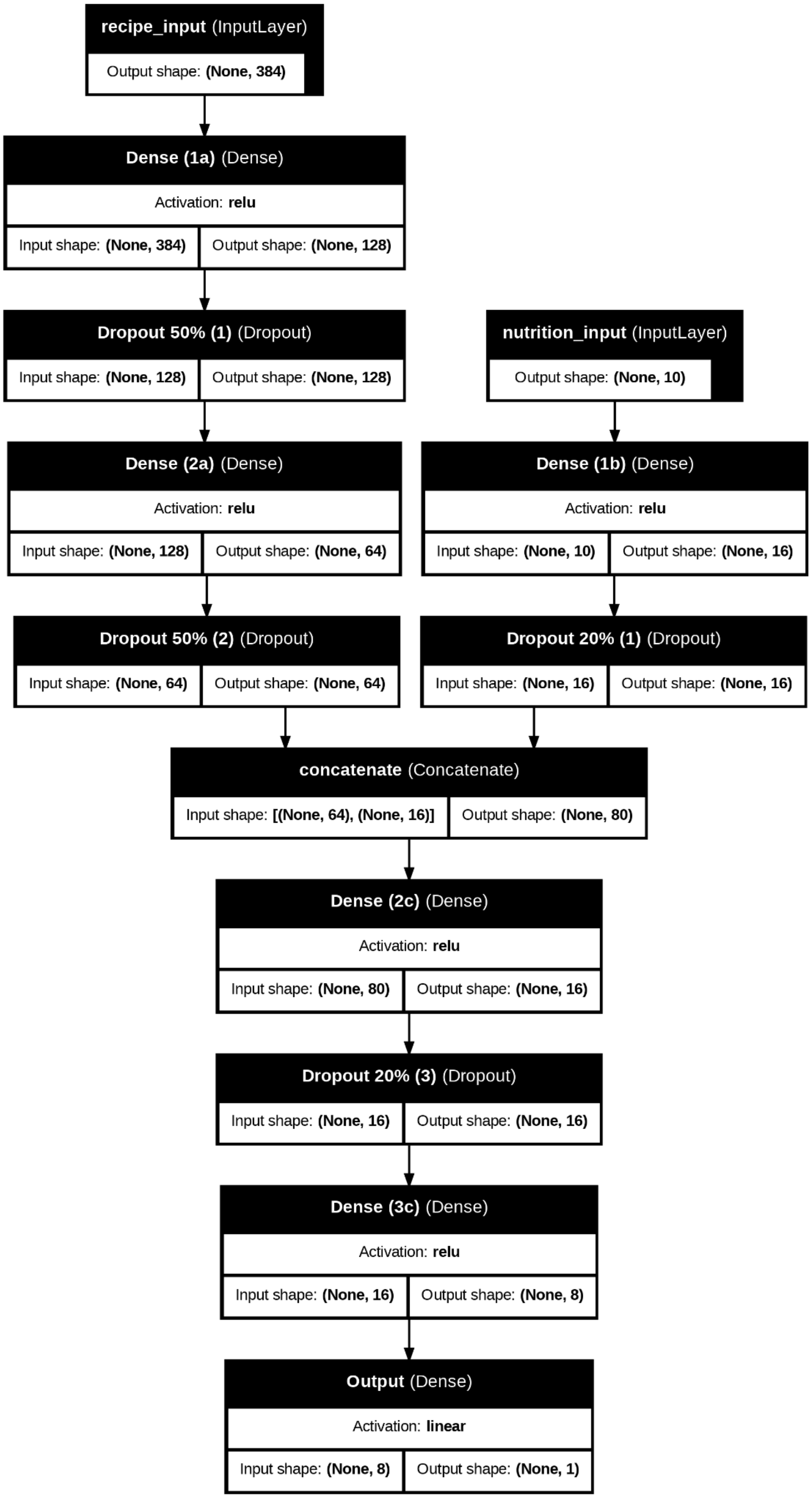


Figure 3: MSE results for all experiments.

Features	Model type	Training MSE	Validation MSE	Test MSE
Metadata	Ridge Regress	1.658	1.467	1.460
	Linear Regress	1.658	1.467	1.460
	Rand Forest	0.050	0.839	0.824
	Neural Net	1.594	1.427	1.416
Untrained SBERT Embeddings	Ridge Regress	1.323	1.407	1.368
	Linear Regress	1.297	1.447	1.417
	Rand Forest	0.046	0.801	0.808
	Neural Net	0.529	0.859	0.868
Trained SBERT Embeddings	Ridge Regress	1.145	1.309	1.264
	Linear Regress	1.142	1.332	1.279
	Rand Forest	0.045	0.789	0.788
	Neural Net	0.388	0.848	0.836
Trained SBERT Embeddings + Metadata	Ridge Regress	1.127	1.240	1.260
	Linear Regress	1.122	1.264	1.283
	Rand Forest	0.045	0.783	0.782
	Neural Net	0.360	0.825	0.815

Figure 4: Histograms of predicted compared to expected results for models trained on metadata.

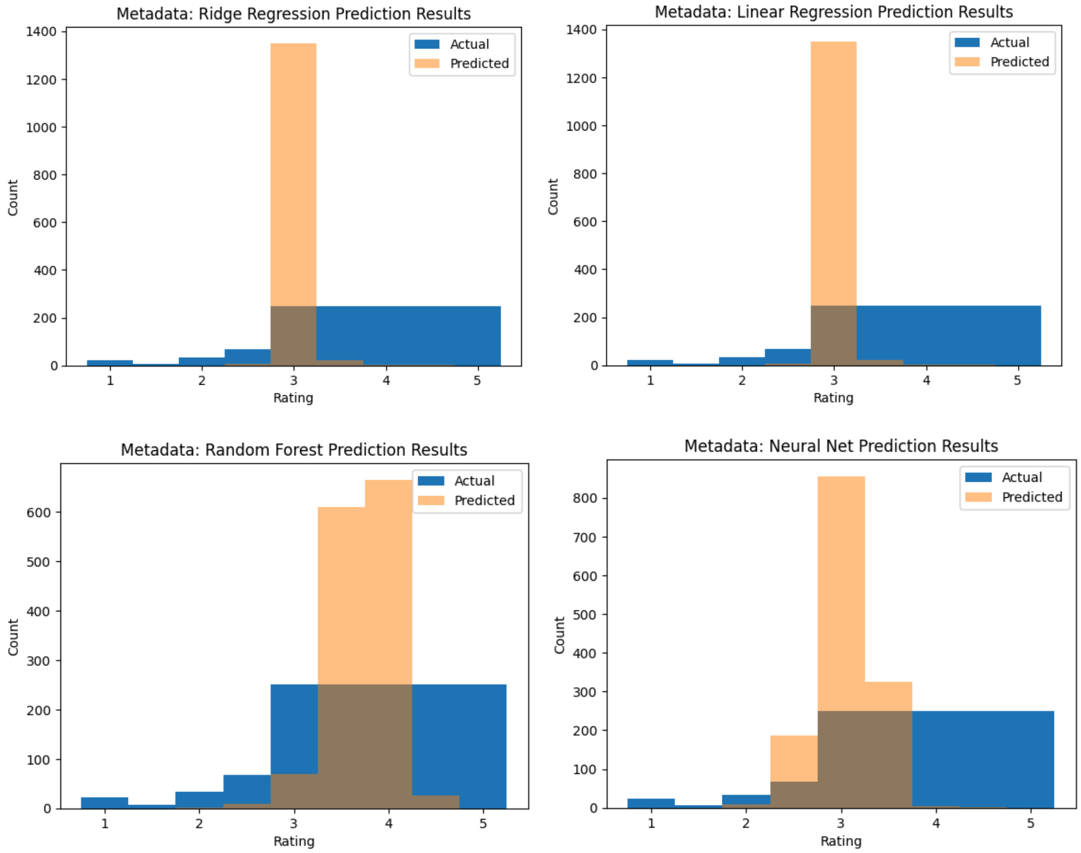
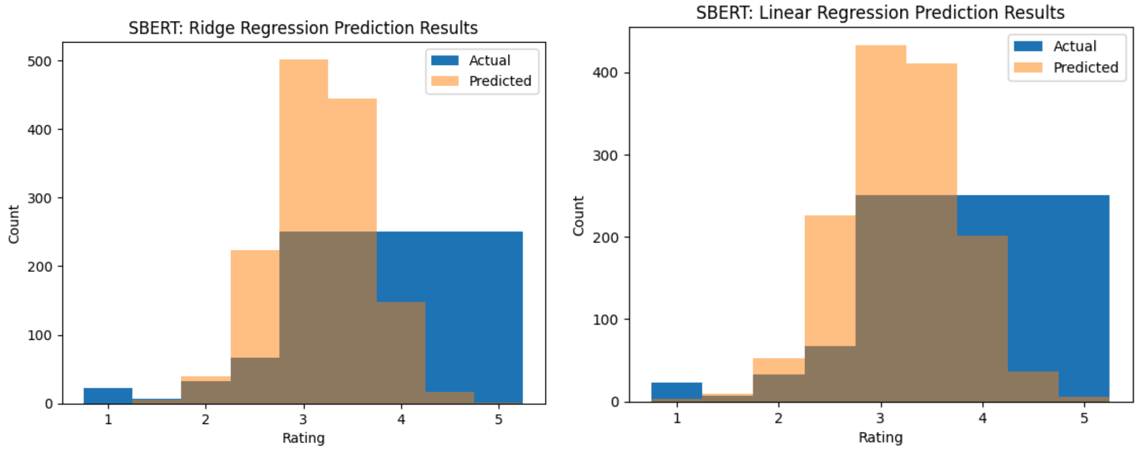


Figure 5: Histograms of predicted compared to expected results for models trained on SBERT embeddings.



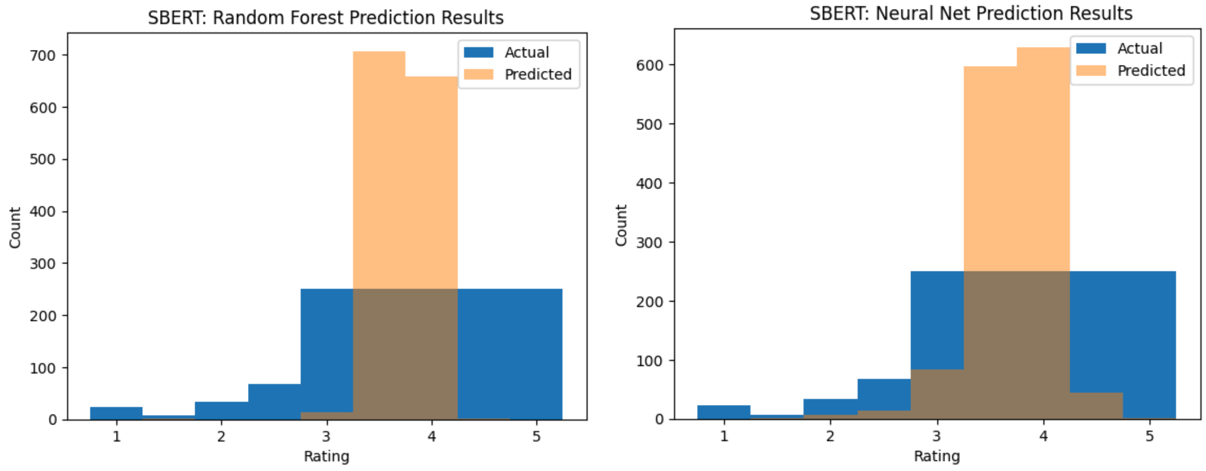


Figure 6: Histograms of predicted compared to expected results for models trained on trained SBERT embeddings.

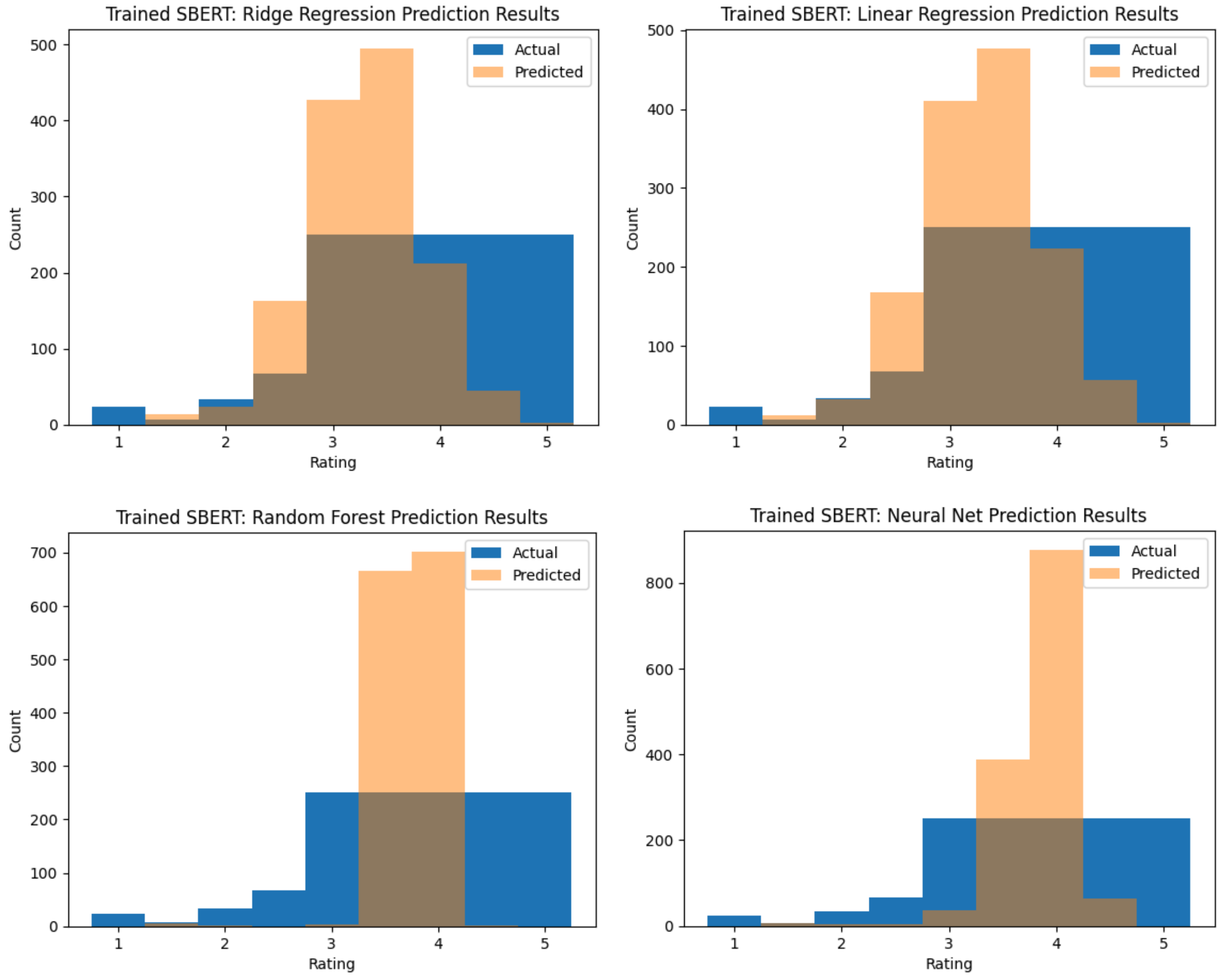


Figure 7: Histograms of predicted compared to expected results for models trained on trained SBERT embeddings appended with metadata.

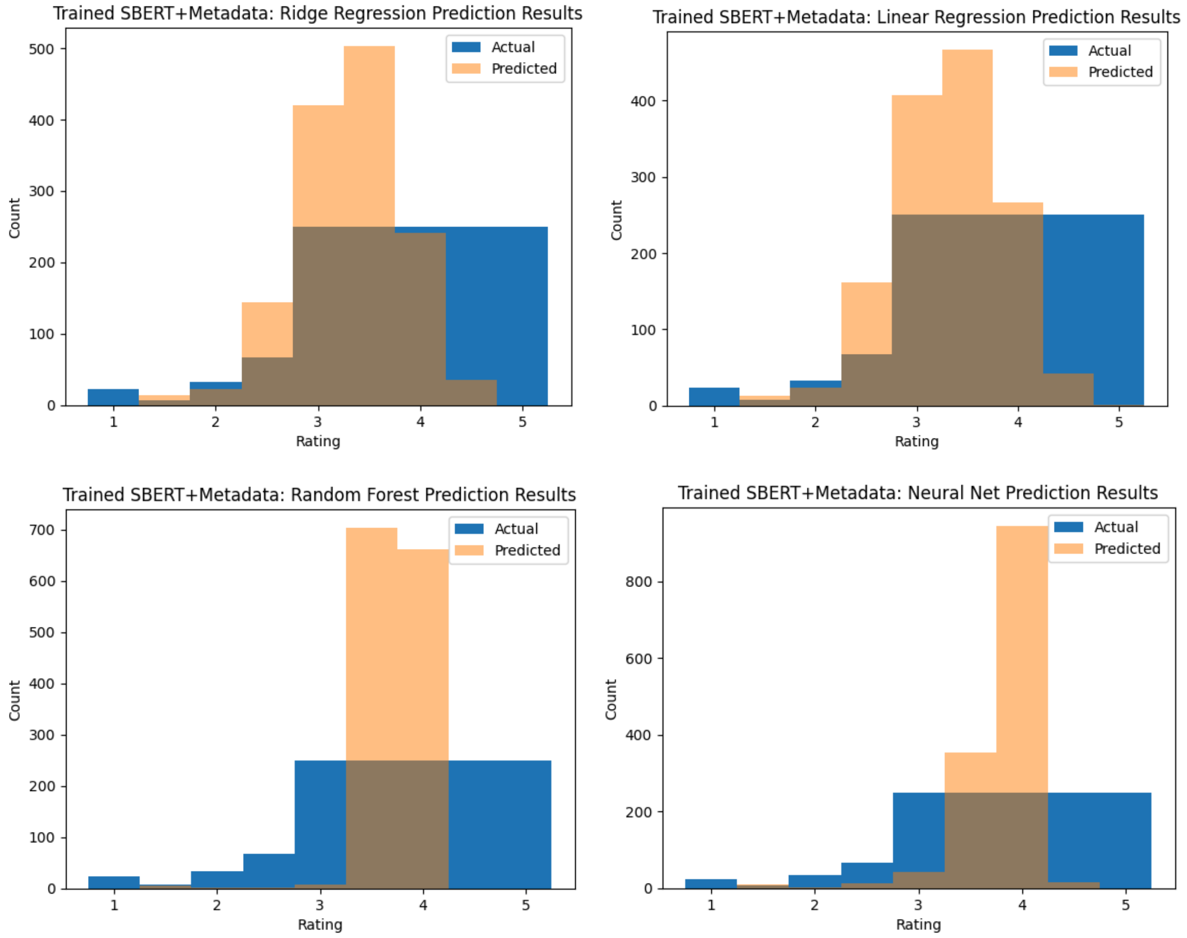


Fig 8.a: Ridge regression predictions separated by actual Ratings (for SBERT+Metadata)

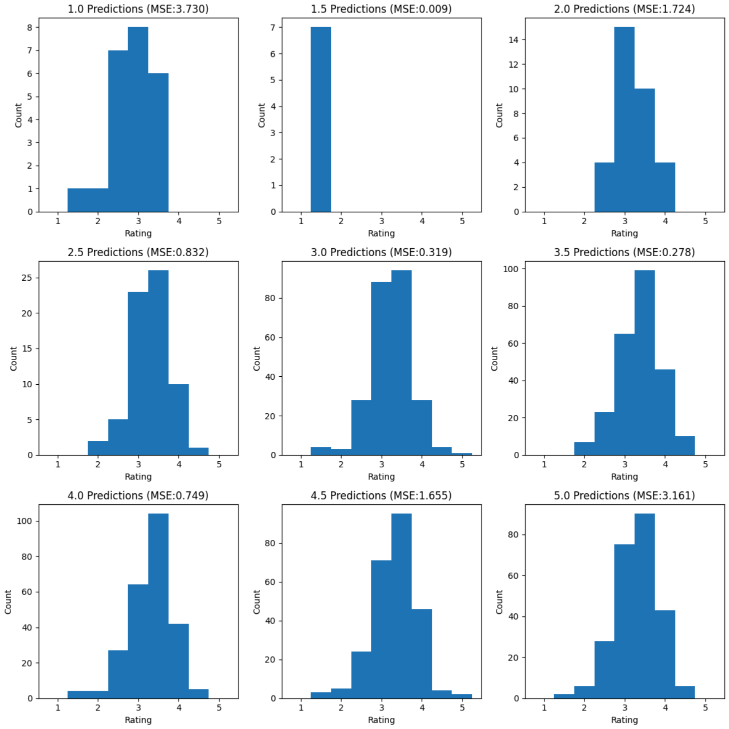


Fig 8.b: Random forest predictions separated by actual Ratings (for SBERT+Metadata)

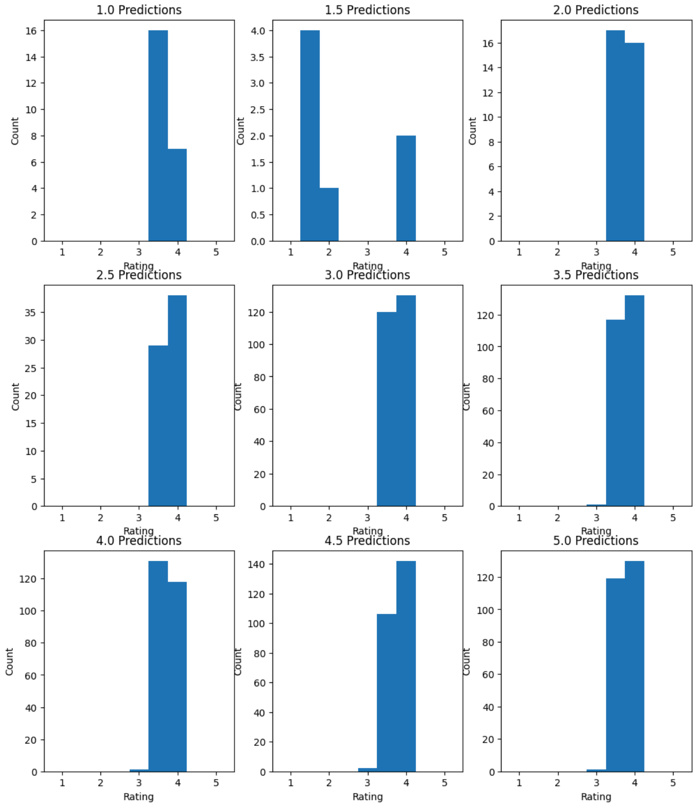
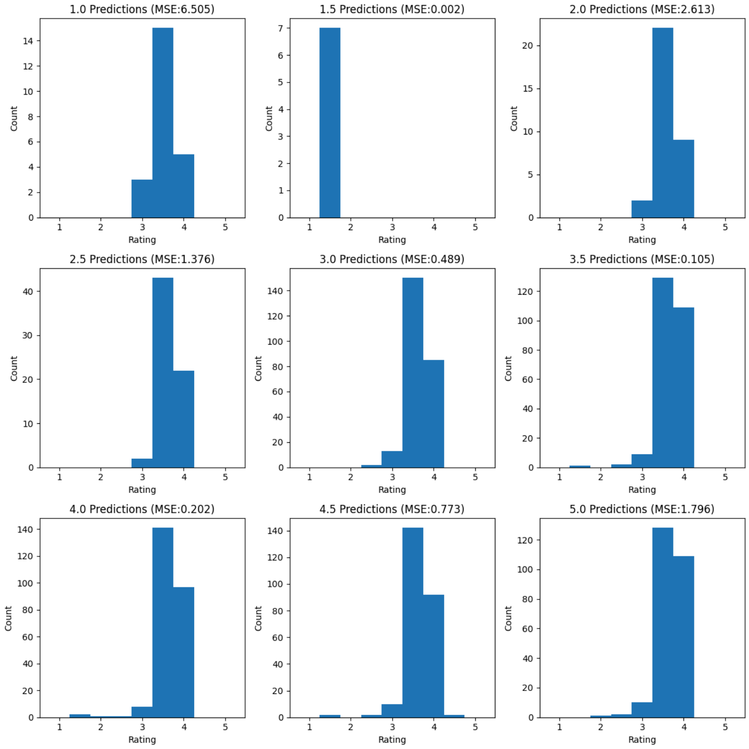


Fig 8.c: Neural net predictions separated by actual Ratings (for SBERT+Metadata)



Authors' Contributions

Alexander Caichen: Data and neural net for metadata-trained predictions, data and neural net for SBERT+metadata trained models, images for paper, Methods, Results and Discussion, Conclusion

Mohammed Elzubeir: Data and neural net related to models trained on trained and untrained SBERT embeddings, Abstract, Introduction, Literature review