What Do People Say: Product Review Analysis with Pattern

1. Motivation and Problem Description

eCommerce is gaining dominance in the global retail market. Due to the lack of in-person interactions on virtual marketplaces, customer reviews become one of the most important channels for customers to communicate with each other and to provide feedback to retailers.

On the one hand, customers rely on reviews from past customers to have a peek at product quality and to make purchasing decisions accordingly. On the other hand, understanding customer reviews is also crucial to the success of online retailers. For instance, retailers can improve product quality or change marketing strategies based on customer preferences and suggestions. Retailers may also rely on review sentiment to predict sales volumes in the immediate future and to determine the quantity of inventories to acquire.

In this project, we work with Pattern, an eCommerce accelerator, to create a natural language processing (NLP) model based on customer reviews to predict product future performance. In particular, we are interested in conducting both short term predictions (e.g. from month to month) as well as long term predictions (e.g. success of a newly launched product after one year). On the one hand, short term predictions can help retailers make sales projections and manage their inventories. On the other hand, long term predictions will inform the retailers if it is worth continuing a newly launched product or how they could change the product to boost its success.

We further separate each of the two prediction horizons into two subtasks.

1. We aim to accurately predict products' future sales (both short term and long term) performance based on customer reviews. In particular, we would like to gauge the effectiveness of using review texts, in addition to review metadata, in sales prediction.
2. Gain insights on what keywords or topics are predictive of popular products

At a high level, the short-term prediction is to predict next month's median sales using historical data, such as past sales and historical reviews. The long-term prediction is to predict whether a product will ever reach the top 3,000 BSR during the one-year period one year after its launch, using data from the first three months after a product's launch.

For subtask 1, we implement separate models using only review metadata and using only review texts. We compare the performance of these two classes of models to gain insights on the advantages of either class. We also construct an ensemble model of the two classes of models and evaluate the amount of improvement, if any, from using

review texts in addition to review metadata.

For subtask 2, we hope to generate a dictionary of keywords or phrases that are predictive of sales performance that are intuitive to and are informative for online retailers in terms of future sales strategy.

2. Datasets and Resources

2.1. Datasets

For this project, we use two main datasets - Best Seller Rank (BSR) history dataset and Review history dataset.

The BSR history dataset is a history of Best Seller Rank for Amazon products within the Vitamins and Dietary Supplements category, ranging from July 2017 to July 2021. The data is scraped by a third-party service provider, Keepa. BSR is a performance evaluation calculated by Amazon using the product's current sales volume as well as the product's historical sales. Rank 1 in a category is the best selling product, 2 is the second best, etc. This dataset contains around 29 million rows and covers 9,991 unique products. The dataset's fields include ASIN (a unique identifier for the product), best seller rank, the average price of the product over the past 180 days, and the date of observation. Figure 1 shows an example of the BSR over time.

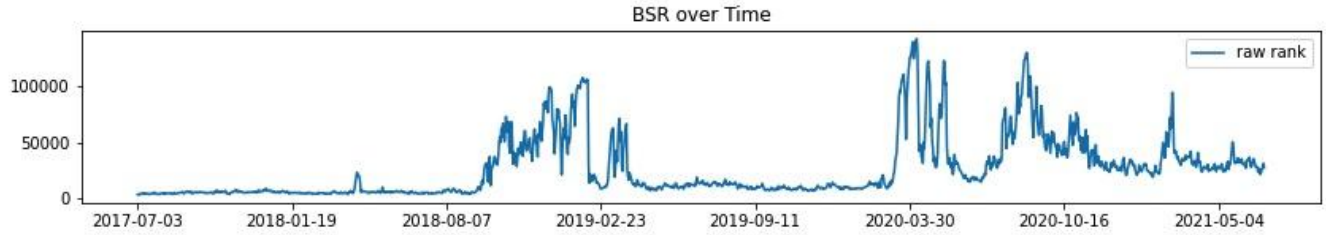

Figure 1: BSR over time for product B00005313T

The review history dataset is a collection of Amazon reviews for products within the Amazon Vitamins and Dietary Supplements category that are on sale at the time of scraping.[1] The earliest review is in January 2004, and the last review in this dataset is in July 2021. There are around 5 million reviews on 9,977 unique products. The dataset's fields include ASIN, product name, review title, review rating, review date, review upvotes, review comment count, and a binary field indicating whether the purchase is verified (an "Amazon Verified Purchase" review means Amazon has verified that the person writing the review purchased the product at Amazon and didn't receive the product at a deep discount). The two datasets cover 9,958 products in common.

We note two main limitations of the datasets. Firstly, the time span is not equal across

[1] Around fall 2021.

products. Keepa, the third party data provider, pays more attention to popular products on Amazon, scraping and updating their ranking information more often than unpopular products. Therefore, popular products are more likely to have frequent, continuous BSR information in our dataset than unpopular products, which tend to have long periods of missing BSR. As a result, when we restrict to products that have frequent reviews and BSR records, the final sample may not be representative of the true population of products and reviews on Amazon. In addition, the dates at which we know the product ranking may not match the dates at which reviews were added for said product, sometimes leading to a small overlapping period for us to work with.

Secondly, the ordinal nature of the BSR hides some important and relevant information we may care about as an online retailer. For instance, the sales volume of rank 1 may differ drastically from the sales volume of rank 2, although they only differ by 1 in terms of ranking. In addition, Amazon does not reveal how exactly they calculate the BSRs. The calculation seems to depend most heavily on daily sales, with little weights on historical sales of the product. As a result, the ranks may suddenly spike or drop, adding noise to the BSR data.

To address the second limitation, Pattern further provides us with their estimated sales volumes corresponding to each BSR (as shown in Figure 2). The mapping is fitted based on actual sales volumes and ranks of their customer retailers, although the exact mapping function between BSR and sales volume is not revealed to us. A drawback is that the estimate for top products may not be accurate, since Pattern's customers rarely attain an extremely small rank (e.g. below 100). The estimates on these top products are essentially based purely on extrapolation of the fitted function.

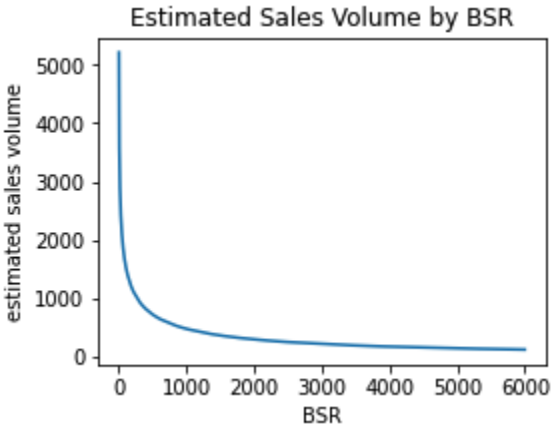

Figure 2: Estimated Sales Volume by BSR

2.2. Target Variable

The target variables are different for the two subtasks. For short-term prediction, we use the '*monthly median sales'* as the target variables. For long-term prediction, the target variable is the *'successful product*'. We define a product as successful if it has ever

reached top 3,000 in terms of Amazon's BSR during the one year period one year after launch. We will describe how these three quantities are generated in the data processing section.

2.3. Data Processing

2.3.1. Short-term predictions

We take the following steps to pre-process the estimated sales data: For ranks without an estimated sales volume (i.e. ranks larger than 454,302), we extrapolate the estimate using a line between 0.05 (which is the estimated sales volume for rank 454,302) and 0. Then we merge the estimated sales data with the original BSR data using rank, to give each rank a corresponding estimated sales volume. We then group the data by product-month and calculate the product's median sales volume for each month. We use the monthly median sales volume as the target variable in some of our models. We refer to this quantity as "*monthly median sales*" throughout the report.

After we split the data for training purposes, we have 2298 unique products containing 61199 observations in the training set and 851 unique products containing 21813 observations in the test set. The training and test sets are the same for all the short-term prediction models.

2.3.2. Long-term predictions

Based on the data processed in the previous steps, we take the following steps to generate new variables for the long-term prediction task: We remove all products whose first review date is earlier than their first BSR date. We then define a product's launch date as its first BSR date. Then, we calculate the optimal (minimum) BSR for a product over the subsequent one-year period after the first launching year. The time period of one year is chosen to minimize the effects due to seasonal changes (which affect the market of vitamins a lot). We then use this optimal BSR to generate our new target variable, the '*successful product*'. We define a product as successful if it has ever reached top 3,000 in terms of Amazon's BSR during the one year period after the first launching year. For every product, we will predict if it will be classified as 1, a successful product, or 0, an unsuccessful product. We pick 3,000 for two reasons. First, it is intuitively a good rank—being one of the top 3,000 products in Amazon's Health and Household category (which contains millions of products) is an unambiguous achievement. Second, using this threshold, about 18% of products in our datasets will be categorized as successful. The threshold is a nice balance between a real success and a balanced classification. We then generate features by aggregating the reviews data and calculating some statistics on the BSR data for the first three months after a product's launch. By doing so, we are able to observe how the initial review and the initial BSR of a product after its launch affects its long-term BSR.

After we split the data for training purposes, we have 2,768 unique products in the training set and 923 unique products in the test set. The training and test sets are the

same for all the long-term prediction models.

3. Models

For both tasks presented in the introduction, i.e. short-term and long-term predictions, we have developed two different types of regression models

1. models based on review metadata, and
2. models based on review texts.

The idea behind this approach is to look at the predictive power of the text as an incremental value-add on the model without any text. We do this by ensembling our two models. This helps us with the interpretability of the models (our second goal), while also giving valuable modeling insights into our prediction task.

3.1. Non-Text Models

The first set of models we run consist of models based on review metadata, by which we mean all information related to a product apart from the text of its reviews. More specifically we input features related to the past sales performances of the product, i.e. BSR related features, as well as the past ratings of the product.

In the short-term predictions, we use all review metadata of a product that was posted in or before this month to predict the median sales of the next month. For long-term predictions, we consider all of the review metadata within three months after the launch of a product to predict its success in a year.

In terms of models, we use a linear regression model, an XGB regressor and an RF regressor for short-term prediction. For long-term prediction, we use a logistic regression model, an XGB classifier and an RF classifier.

3.2. Text-Based Models

The second set of models we run is the text-based model, which only takes review texts as features. These models have no access to review metadata such as the rating or purchase verification; they are also unaware of any historical sales performance. The rationale is that a component text-based model should be able to pick up, to some extent, the general sentiment of each review from the raw text.

In particular, we use all reviews under a product that is posted in or before this month for the short-term prediction of median sales of the next month; for long-term prediction of successful product, we look at all reviews within three months after the launch of a new product.

We first use a bag-of-words model for its two major merits. First, it is easily trainable and provides a baseline performance that can be compared with a more complex transformer based model. Second, a bag-of-words model is highly interpretable. For instance, we can

simply look at the phrases that are associated with positive or negative coefficients in a linear regression, and gain some insights on what topics are associated with high or low sales performance.

The predictors are the (weighted) frequency of the 500 most common phrases in the training corpus. For text processing, we experiment with the bag-of-words model, where we simply count the occurence of each of these 500 phrases, as well as the term frequency-inverse document frequency (TF-IDF) model, where the weighted frequency of each phrase is calculated from the term frequency and inverse document frequency. The TF-IDF model has the advantage of adjusting for the fact that some words are used more commonly in general. We then use a regularized linear (logistic) regression model to predict the monthly median sales of the following month (succes in 1 year).

The set of hyper-parameters we experiment with includes: simple bag-of-words model vs. TF-IDF model, unigram vs. bigrams vs. trigrams, L1 vs. L2 regularization, and the different penalty strengths.

Despite the interpretability of the bag-of-words model, it is often too simplistic to capture the holistic meaning of a full text sequence. On this front, a transformer based model is the state of the art. We, therefore, also employ transformer models, specifically tiny BERT, to try and accomplish the same task that was described earlier. The structure for our BERT-based model is described in Figure 3 below. We first pass all our reviews through the transformer model to generate individual embeddings for each review, then aggregate over all the embeddings to make the model resilient to changes in the number of reviews that we pass through the model, and finally we pass our aggregate review embedding through dense layers to create our prediction (regression or classification depending on the task).

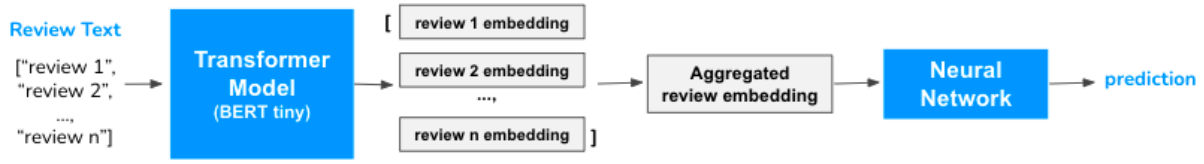

Figure 3: Model structure for BERT-based model

The set of hyper-parameters we experiment with for the BERT-based model includes: max sequence length, number of training epochs, number of dense layers in the neural network leading up to the individual review embeddings and the network post the aggregate review embedding, and latent embedding sizes.

3.3. Ensemble Models

Recall that a main goal of this project is to gauge the effectiveness of using review texts, in addition to review metadata, in sales prediction. This question is particularly interesting

and relevant in the long-term prediction scenario. For a new product, we would like to understand if the initial reviews reveal anything additional information about the product that is not captured by the initial sales performance. In other words, if we can assess the growth *potential* of a new product using the reviews in the first few months after product launch.

As a result, we implement an ensemble model that takes into account the predictions of four individual models: random forest, XGBoost, bag-of-words, and BERT.

The target variable of the ensemble is a binary variable of whether the product is successful. The predictor variables are the predicted probabilities from the four individual models. We experiment with two models, a simple logistics regression and a decision tree classifier. The ensemble model is intentionally kept simple to avoid overfitting, especially after an already exhaustive hyper-parameter tuning within each model class. In either case, we fit the model on the validation set[2] and make predictions on the test set to evaluate the performance of the ensemble models.

4. Results

For short-term predictions, we use R^2 and RMSE to evaluate the performance of regression models by comparing the predicted sales volume with the ground truth value.

For long-term predictions, we use confusion matrices, ROC and AUC to evaluate and tune the models, and use F1-score as the main evaluation criterion to compare the performance of different models. We can read precision and recall from the confusion matrix, where precision is the proportion of correctly identified success cases, and recall is the proportion of actual success cases that are correctly identified. We care about the precision of our models because we want our models to make accurate predictions about the future performance of every product. We also care about the recall of our models because we do not want our models to easily overestimate the future performance of a product and thus give a false signal to our customers. To get the best precision and recall at the same time, we use F1-score, which is the harmonic mean of precision and recall values for a classification problem. The above-mentioned metrics change with the changing threshold values in the binary classification problem. Thus, we generate AUC-ROC curves to easily visualize which threshold is giving us a better result. ROC curve is an evaluation metric for binary classification problems. It is a probability curve that plots the true positive rate against the false positive rate at various threshold values. The AUC is the measure of the ability of a classifier to distinguish between classes. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

---

[2] In particular, we fit the model on the validation set. The same validation set is used for cross-validation for all individual models. We choose not to fit the ensemble model on the training set, because the non-text-based models are overfit to the training set—they correctly predict almost 100% of observations; however, the text-based models, especially the BERT model, are not overfit to the training set. As a result, if we fit the ensemble model on the training set, the ensemble model will simply use the predictions from the non-text-based models.

4.1. Non-text Models

We tune the hyperparameters of the three non-text models and find the results shown in table 1. The results of the non-text models are extremely good for the short-term predictions. This is intuitively clear as current sales performances are a determining factor for short-term sales performance predictions, i.e. the current success of a product will be a clear predictor for its success on a short time-scale. This intuition will become clearer when analyzing the models in the following section.

Models	Hyperparameter	R^2 score
Linear regression	———	0.938
Xgboost	learning rate = 0.05 n estimators = 100	0.946
Random forest	max depth = nodes expanded until leaves pure n estimators = 500	0.939

Table 1: short-term prediction results on test set for hyperparameter tuned non-text models.

For the long-term success prediction, we tune the hyperparameters of the three classifiers and find the results shown in table 2. The results are surprisingly high for the long time-scale over which the models make predictions.

Model	Best hyperparameter	F1 score
Logistic Regression	———	0.0121
Xgboost	learning_rate=0.2 n_estim=100	0.371
Random Forest	max_depth=5 n_estim=200	0.338

Table 2: long-term prediction results on test set for hyperparameter tuned non-text models with past-performance and rating features.

It is intuitively clear that for long-term predictions current and past sales performances will be less predictive than in the short-term. Based on this intuition, we also consider for comparison the same three models using rating-based features only. The scores of these models with a reduced number of features are shown in table 3. These results are almost comparable with the results of the models with all features in the input in table 2. It is

thus clear that for long-term predictions the rating-related features are important.

Model	Best hyperparameter	F1-score
Logistic Regression	———	0.0345
Xgboost	learning_rate=0.2 n_estim=500	0.318
Random Forest	max depth = nodes expanded until leaves pure n_estim=500	0.245

Table 3: long-term prediction results on test set for hyperparameter tuned non-text models with rating features exclusively.

4.2. Text-Based Models

For the short-term task, our best BoW model achieves an R^2 score of 0.14 with a bigram TF-IDF model with a LASSO regression using a penalty strength of 0.1. The R^2 score is quite high, especially given that we are only using a simple linear model on a very sparse model matrix. On the other hand, our best BERT-based model has an R^2 score of 0.164, which is approximately a 10% improvement on our best bag of words model, but is far from the performance of the models that use momentum data as features. This brings us to the conclusion that momentum data is far more important than review data for short-term predictions.

An interesting, but expected trend we see when comparing the BERT-based model's performance with the bag of words model for the short term task is that the transformer model does better at getting the tail predictions right. This is seen in the bin scatter plot below.

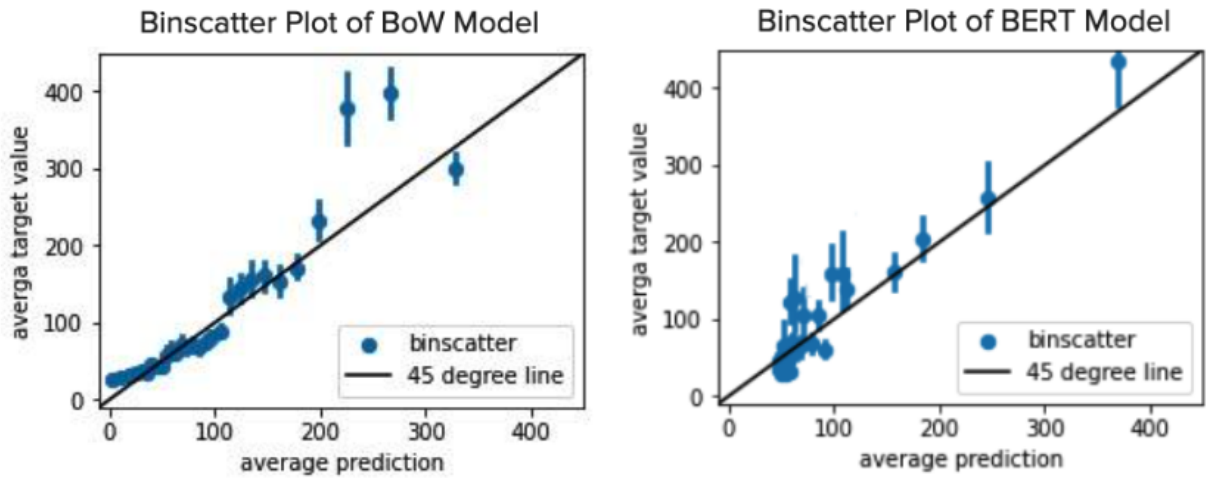

Figure 4**:** BoW vs. BERT model performance on short-term task

The story is slightly different for the long-term prediction task—text-based models, especially the BERT model, begin to outperform non-text, historical sales based models.

Using the BoW model, the highest F1-score is at 0.34, achieved using a TF-IDF model with unigram and L2 penalty at a strength of 0.5. Figure 5 tabulates the F1-score on a hold-out validation set across different model/hyper-parameter combinations.

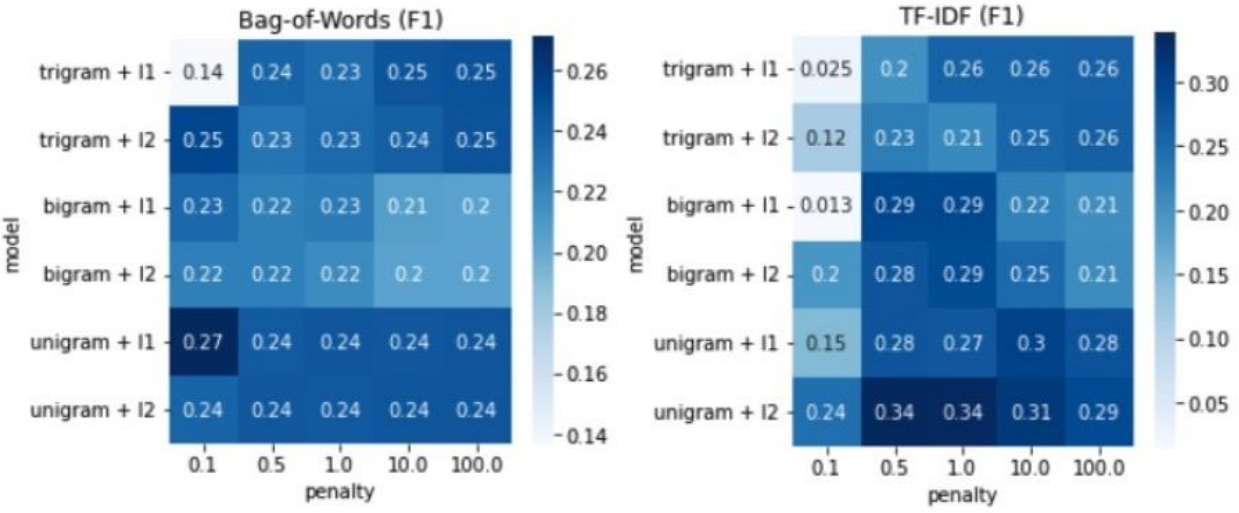

Figure 5: F-1 scores of BoW for the long-term prediction

We make two major observations. First, trigram models overall perform the worst. This is potentially due to lack of variation in trigram features. Note that we are using only the first three months of reviews for long term predictions, which results in a small training corpus. In particular, the 500 most common trigrams collected from the training corpus

usually have low frequency with a median of 3. Second, unigram models overall perform the best. One potential reason is that the variation in frequency of unigrams is much higher, making the model capable of explaining the variation in target variables. Another potential reason is that simple word counts can already capture a lot of meaning or predictive power of the reviews.

Turning to the BERT-based model. Figure 6 shows the performance across different model choices:

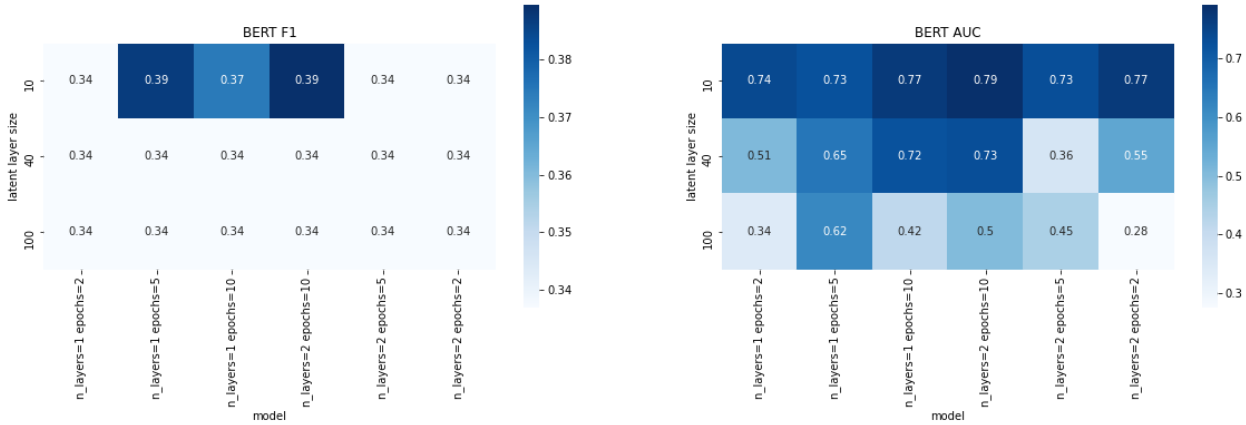

Figure 6: BERT-based model performance for the long term task

We see from the results above that the F1-score is fairly stable, and is not very sensitive to the model choices we make. The AUC, however, does get affected significantly by the model choices. We see that a small embedding size (of 10 elements) works better - this is perhaps because a smaller number of trainable parameters reduces the variance of the model, not allowing it to overfit on the train data.

We also see that the best model is one that is trained on 10 epochs and has 2 dense layers for each FFNN structure. The 2 dense layers working better than 1 is probably indicating that having more layers allows for us to reduce the bias of the model and counteract the rigidity introduced by the small embedding size. The 10 epochs is an artifact of the model structure we adopted, where we take a mean across all the embeddings to make one prediction - this means each review embedding doesn't have its own feedback loop, but instead has to share one with other reviews in that data point. This is perhaps the reason why we need to regress on each data point 10 times for us to get the best result.

In Figure 7 below, we see the performance of the BERT-based model as a function of the max sequence length hyper-parameter.

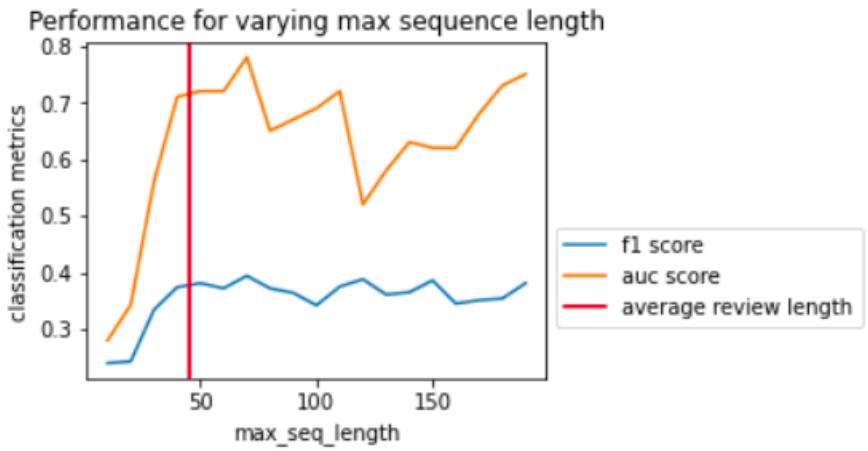

Figure 7: Transformer model performance w.r.t. max sequence length

We see in the above plot that the performance of the model is poor when the max sequence length is very small - this is because we are, perhaps, getting rid of important information and truncating reviews to be too small. We also see that the performance overall sees a plateau once the max sequence length exceeds the mean review length. This is an indication that the model doesn't need to read the entire review to get the information it requires for the prediction, and after a certain threshold, the longer review contains a similar signal to the start of the review, so we don't need to read the whole review to make a prediction. This can also be taken a step further, and can be interpreted as the threshold to which most customers will stop reading a review before making a purchase decision.

4.3. Ensemble Models

The ensemble models exhibit a noticeable improvement in prediction performance, suggesting that review text is conducive to long-term sales prediction in addition to review metadata and historical sales performance.

In table 4 below, we compare the performance of our individual models on a test set along different performance metrics.[3]

[3] Note that all models are tuned to maximize F1-score—all other metrics, AUC, precision, and recall, are computed using the set of hyper-parameters that attains the highest F1-score.

Models	F1	Accuracy	AUC	Precision	Recall
Ensemble (logistics regression)	0.437	0.835	0.805	0.557	0.360
Ensemble (decision tree)	0.536	0.839	0.786	0.548	0.524
BERT	0.438	0.872	0.755	1.00	0.281
Bag-of-Words	0.321	0.762	0.670	0.325	0.317
XGBoost	0.388	0.829	0.758	0.532	0.305
Random forest	0.402	0.832	0.759	0.547	0.317

Table 4: Performance of ensemble models v.s. individual models

In addition, in Figure 8, we plot the ROC curve of the ensemble predictions as well as the individual model predictions.

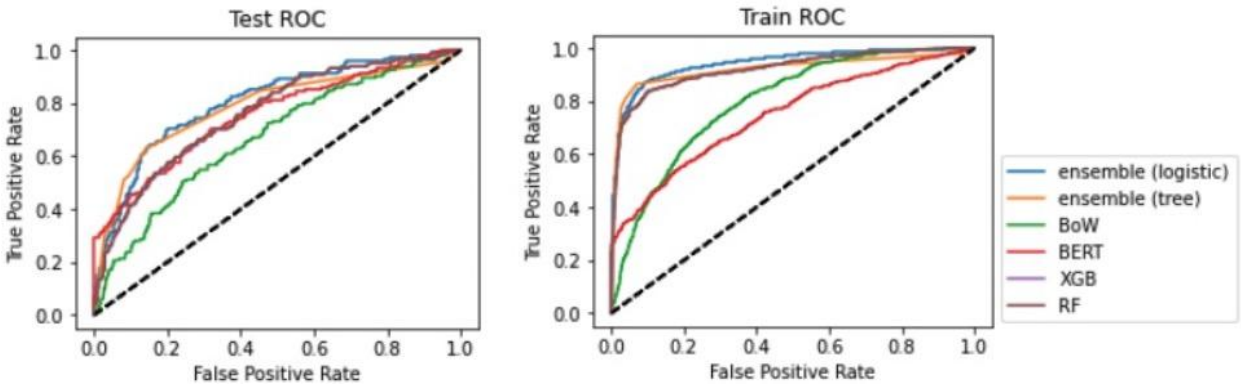

Figure 8: ROC curve of ensemble v.s. individual models

Overall, the ensemble models outperform all individual models. The only exception is for precision score: while the BERT model has a precision of 1,[4] the ensemble model achieves a much lower precision. In particular, the BERT model makes 46 positive predictions, all of which are correct. As a comparison, 86 out of 152 and 59 out of 106 positive predictions are correct from the decision tree and logistic regression ensemble model, respectively.

The increase in performance of the ensemble model suggests that the non-text-based models and the text-based models have different advantages. Especially, using review texts helps model predictions even if we have historical sales data. Specifically, F1-score increases quite substantially from 0.4 in a random forest non-text model to over 0.5 in the

[4] BERT is especially stringent on positive predictions. The false negative predictions, however, do not show a clear pattern in terms of the actual underlying BSR. The false negative products have minimum BSR almost uniformly from 0 to 3000 (which is the cutoff of a successful product).

decision tree ensemble model.

It is also hard to distinguish which of the two ensemble models is better—the decision tree results in higher F1 while the logistic regression results in higher AUC. In fact, depending on the relevant performance metric, neither ensemble model might be ideal to make the final prediction. To see this, recall that the BERT model is 100% correct on positive predictions. However, neither ensemble model sticks fully to the positive predictions from BERT. For example, the logistic regression ensemble model incorrectly predicts 24 of the 46 positive observations to be negative, and the decision tree ensemble model incorrectly predicts 2 of the 46 positive observations to be negative. In some real life applications, it might be detrimental for retailers to miss even a single successful product, in which case, an ensemble model that takes all positive predictions from the BERT model (and potentially some positive predictions from other individual models) might be more beneficial.

5. Interpretation: What makes a successful product?

5.1. Short-term Success: Momentum

For short-term predictions, the future sales performances are almost entirely predicted by current and past performances. In particular, the non-text models outperform the text-based models by a large margin. For that reason, we only consider the non-text models for short-term predictions. Their most predictive features are the mean performances of the products over the given month as can be observed in figure 9 for the XGB model. This feature is by far the most important feature for the XGB model, which indicates that it is making predictions almost entirely based on past performance.

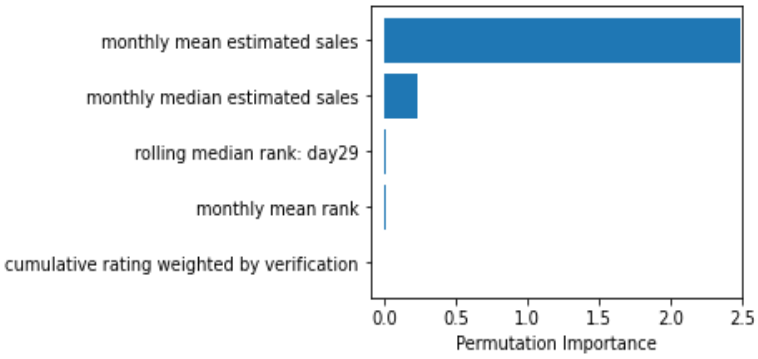

Figure 9: XGB regression model features importance plot. The monthly mean estimated sales is by far the most relevant feature.

The partial dependence plot for the monthly mean performance of the products shown in figure 10 indicates that the XGB model is making predictions based on the principle of momentum, i.e. a low monthly mean performance will result in a lower predicted sales performance in the next month. This observation is expected intuitively. Indeed, the sales volume of a product fluctuates a lot on a timescale of approximately a day due to the

changes in price and availability. On the other hand, the average sales performance of a product over a month does not fluctuate a lot. It thus makes sense that the model shows a linear dependence on the sales volume of the previous month.

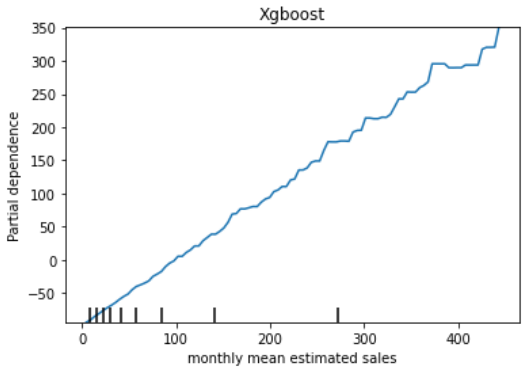

Figure 10: XGB partial dependence plot for monthly mean performance. The relationship between the target variable and the monthly mean performance is linear indicating that the model is predicting based on the principle of momentum.

5.2. Long-term Success: Number of reviews

The number of reviews is an important feature for the long-term prediction of the non-text model. This supports the intuition mentioned under the long-term prediction results of the non-text model, that is: in comparison to short-term prediction, not only past performance but also review metadata are important for long-term prediction.

In particular, the number of verified reviews is a predictive feature from which we can gain some interesting insights into the sales and marketing strategies of e-commerce vendors on Amazon. Indeed, the feature importance histogram for the long-term prediction of the non-text models is well distributed in comparison to the short-term one, as can be seen on figure 11. This implies that the model needs several features to make predictions. Moreover, features unrelated to past performance are also relevant: the number of verified reviews is the fifth most important feature.

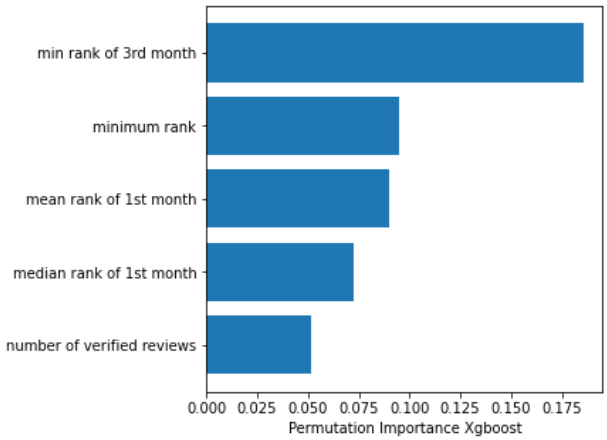

Figure 11: Feature importance of XGB model for long-term prediction. The histogram is well distributed over several features showing that the model is learning a more complex behavior compared to the short-term prediction model. The number of verified reviews is unrelated to past performance but is still the fifth most relevant feature.

The partial dependence plot of the number of verifier reviews for the XGB model which is shown on figure 12 is particularly interesting. Indeed, the dependence on the number of verified reviews is at first counter-intuitive. Normally we would expect that the more verified reviews the more successful the product as it would show that the product is getting a lot of traction. This is in fact the behavior we see for the total number of reviews, i.e. verified and not-verified, as discussed below. However, it is known that a lot of e-commerce vendors create fake verified reviews during the first few months after the launch of their products to create an artificial traction for their product. We interpret the trend shown on figure 12 to represent this phenomenon, i.e. the more verified reviews the more suspicious it is that they are faked and hence predictive of an unsuccessful product.

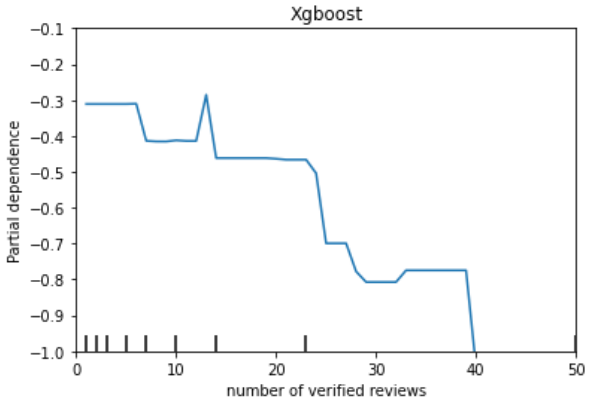

Figure 12: Partial dependence plot of the number of verified reviews for the long-term predictions of the XGB model. The RF model for long-term predictions

shows a similar partial dependence on this feature.

From plotting our best BERT-based model predictions against the number of reviews in figure 13, we see another indication of the number of reviews being a strong indicator of success. This is shown in the binscatter plot below.

An important thing to note about this result is that in the way we structure our BERT-based model, the model has no explicit indication about the number of reviews (because we take the mean of our embeddings to create an aggregate embedding rather than summing over that axis). This means that our model is not only able to infer the number of reviews, but also realize that more reviews means more sales, and therefore means a higher probability of success. Another thing that is important to clarify is that the plot here is unlike the plot for XGB where we were looking at the number of verified reviews, while here we look at the total number of reviews.

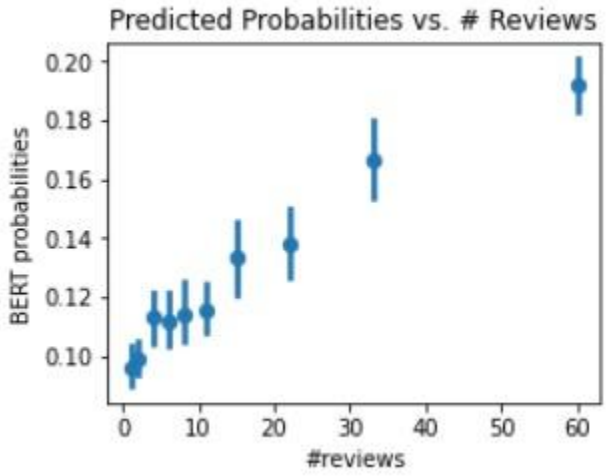

Figure 13: Binscatter plot of the best BERT-based models predicted probabilities for success vs the number of reviews.

5.3. Long-term Success: Review ratings

The review ratings give a similar insight into the data than the number of verified reviews discussed above. The mean review rating is one of the most important features for the long-term predictions of the RF model as shown on the feature importance plot in figure 14. Similarly to the XGB model, we infer from the feature importance plot that long-term predictions require more features than the short-term ones and that review related features are also relevant for this task.

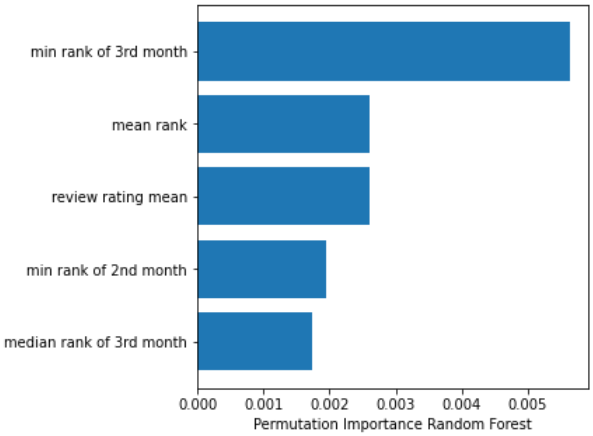

Figure 14: Feature importance plot for the long-term predictions of the RF model.

The partial dependence plot for the mean review rating is shown in figure 15. We observe a similar counterintuitive trend for this feature as for the number of verified reviews presented above, i.e. a very high mean review rating corresponds to a small probability of success. We believe that this result is also due to the fake reviews produced by vendors during the first few months after the launch of their product. Indeed, a mean review rating that is too high is not representative of real data but shows that it has been crafted. On the other hand, a mean review rating of around 4 corresponds to the highest probability of success as it is more representative of the distribution of real good reviews.

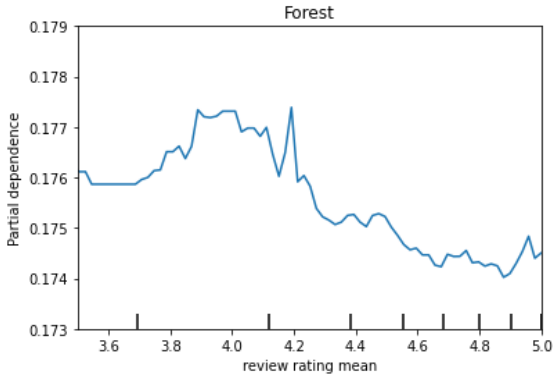

Figure 15: Partial dependence plot of mean review rating for the long-term predictions of the RF model.

5.4. Long-term Success: Average Length of Reviews

We also investigate the relation of the transformer model's predictions with average review length, and we see the interesting trend shown in figure 16. This trend shows that for very short or very long reviews, our transformer model is more likely to predict the product to be unsuccessful, while for reviews of length somewhere in the middle, it is more likely to predict them to be successful. This is perhaps because of a tradeoff we see with the length. Namely, when the reviews are too short, the transformer model

doesn't have enough signal in the text to make a confident prediction of the product being successful (it is worth noting here that the transformer was definitely one of the more conservative models). On the other hand, when reviews are very long, they are more likely to be negative reviews, because they include long comments about complaints the reviewers have. So, reviews in the middle (of average length around 40) are the ones that are most likely to result in a prediction of success because they have enough tokens for BERT to gain enough signal, and they are also more likely to be positive reviews compared to the longer reviews.

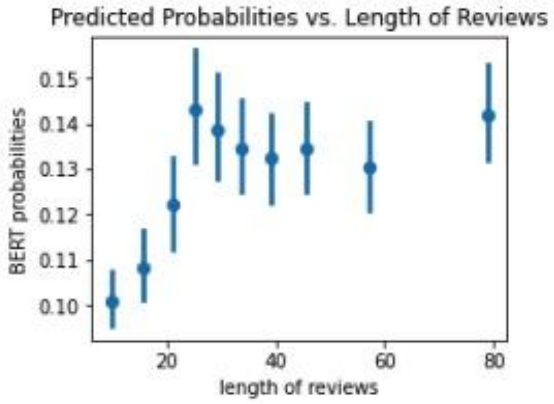

Figure 16: Binscatter plot of the best BERT-based models predicted probabilities for success vs the average length of reviews.

5.5. Long-term Success: Review Content

The BoW model provides some insights in terms of what phrases are associated with higher probability of being successful.

First, reviews that mention ingredients are more likely to make up a successful product. To see this, we examine the BoW-based regression models, and note that unigrams associated with the 50 most positive coefficients include the following ingredients: fiber, acv (apple cider vinegar), b12, fish, oil, turmeric, coffee, elderberry, and enzymes.[5] On the other hand, we see no ingredients among the 50 most negative coefficients. In general, it appears that objective description of the ingredients appears to be more important than subjective feeling of the product. In fact, we see both positive and negative coefficients associated with complimentary words—"improve," "perfect," and "amazing" are some of the most positive unigrams, whereas "relief," "helping," and "nice" are one of the most negative unigrams. This is quite intuitive, as some of the complimentary reviews could be fake and are posted by sellers. On the other hand, customers might be looking for certain effective ingredients in their desired vitamin products, and objective descriptions of ingredients may sound more trustworthy.

Second, reviews that convey a sense of healthiness are associated with long-term

[5] In decreasing order of coefficient size

success. In this case, we look at words that frequently appear in BERT's positive (v.s. negative) predictions. For example, "natural," "organic," and "healthy" are three common words in BERT's positive reviews, but do not, or rarely, appear in BERT's negative reviews. In addition, "kid" is also frequently appearing in positive reviews. The connotation appears to be that the product is safe for kids, which conveys another level of reassurance.

Third, frequent negations in reviews is a sign of dispreference. For example, "didn't" and "doesn't" are two words that frequently appear in negative reviews but not in positive reviews. Intuitively, these negations are associated with complaints: for instance, "this product doesn't work for me." Overall, the negative sentiment is associated with bad-quality products.