E-commerce Product Optimization

Lauren Ally and Sean Merkle

DS4420, Professor Eric Gerber

GitHub Link: https://github.com/laurenally28/EcommerceProductOptimization

Introduction and Literature Review

Understanding and forecasting product performance in e-commerce has become increasingly important as online platforms grow more data-driven. The online market faces lots of competition and constantly changing consumer preferences, making it very important for vendors and sites to effectively price their products and estimate future sales to stand out and meet evolving demands. Vendors often struggle with determining how to price their products best and estimate future sales. This could be especially useful for vendors when trying to optimize pricing. Our project looks to address this by applying various machine learning methods—which is a powerful tool for understanding and anticipating future trends in e-commerce, to predict key performance indicators. These insights can help vendors test strategies before implementation, potentially reducing risk and increasing revenue. Furthermore, by understanding how many units of a product to order as well as how to price it, we can also explore ways to cut costs to increase overall revenue, aligning with the critical need for effective inventory management in e-commerce. Our project contributes to the growing trend of using data-driven approaches to solve key problems in online retail, focusing on predicting the number of units bought in the last month.

Past research has extensively explored the use of predictive analytics in e-commerce to address challenges similar to those we aim to tackle in this project. For instance, Jakkula (2023) highlights the application of time series analysis and models like ARIMA and LSTM for sales forecasting, a core objective of our work that seeks to predict the units sold for each product. Similarly, predictive analytics has been crucial in inventory management, with machine learning algorithms such as decision trees and neural networks being used to forecast demand and optimize stock levels (Jakkula, 2023). While these studies provide valuable insights into the application of various methods for prediction in the broader e-commerce sector, our project distinguishes itself by focusing on the application of a Multilayer Perceptron Neural Network and Bayesian Modeling to a specific dataset of products on Amazon USA. This allows us to examine the nuances of predicting the number of units sold per month and potentially identifying strategies to optimize inventory and production of goods.

Various experts also emphasize the importance of data quality and integration to the success of predictive models regarding e-commerce. Many times, vendors struggle to obtain

quality data due to the heavy isolation of data systems throughout their supply chain (Abigail, 2025). Alongside this, many external factors can hinder the accuracy and precision of predictive models. This can include things like economic fluctuations, geopolitical issues, and constantly changing consumer preferences. Overall, we do think many organizations can benefit from predictive analytics in e-commerce, but must acknowledge the variables that can hinder the performance of models due to data quality issues and uncontrollable external factors.

ML Methodology and Data Collection

For this project, we chose to work with an e-commerce pricing dataset sourced from Kaggle. While exploring dataset options, it became clear that whether or not we wanted to focus on price optimization or predicting units sold, we would need a description, price, and units sold. Intuitively, it makes sense that these three features would have a stronger relationship than any others. So, in short, we chose this dataset because it was the only relevant one we could find with these three key features. We were also lucky that this is the one because it is US-based data, has ~1.7 million rows, and includes a category column as well which can be used for understanding secular trends as well.

We started by preprocessing the data. With the dataset being extremely long, we dropped any rows with missing values. Next, we converted the *isBestSeller* column to 0/1 for numeric modeling. Then, we had to deal with the key non-numeric features being *category* and *title*. For *category*, we one hot encoded by separating it into binary columns. As for *title*, we used TF-IDF vectorization, which transforms text into numeric features based on word importance across the dataset. This allowed our model to capture the more important descriptive terms in the title, such as product types or important attributes. Finally, we used a scalar (StandardScalar) for all numeric features so that they are all on a consistent scale centered at 0 and a standard deviation of 1.

Our first method was manual (no-packages) MLP in Python. We used the *stars*, *reviews*, *isBestSeller*, *price*, *category*, and *title* features to predict the number of units that would be sold in a month. This consists of several steps. First, the forward passes through the data, which is followed by the calculation of the loss. Next, we take the gradients, which are the derivatives of W and b. And finally, we update the W and b values. At first, our network did a very bad job at

generalizing—returning an alarmingly low R-squared score. To combat this, we added a hidden layer and increased the size of these layers so that they could handle the large amount of data we were giving it. From a data perspective, we decided to use *SentenceTransformers* (SBERT) to better understand the relationships between words and their meanings within the context of the complete sentence. This applied specifically to the *title* column, which represented the name of the product. These changes as well as the inclusion of an L2 regularization term drastically increase our model's ability to generalize, increasing the R-squared score greatly.

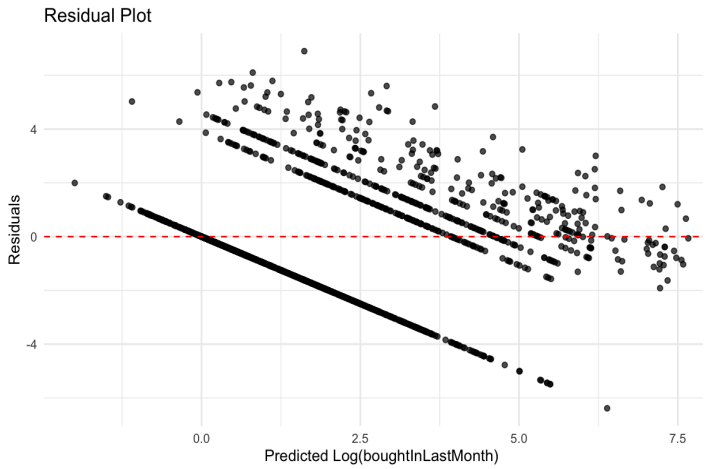
The second method we used was a Bayesian Linear Regression model in R using prebuilt packages. This was meant to give us a more probabilistic approach to this problem. We thought that this was important since this kind of framework accommodates random effects in the data naturally. Additionally, we thought about the fact that it could handle non-linear effects without resorting to approximate methods. Bayesian modeling is a meaningful approach for predicting the number of units that would be sold in a month since it allows us to incorporate uncertainty directly into our estimates and model real-world purchasing data behavior with more flexibility. In our dataset, sales counts are often skewed and influenced by several product-level features like price, customer reviews, and bestseller status. So, a Bayesian model allows us to express our prior beliefs about how these variables might influence the number of units sold in a month, and to update said beliefs. This is especially useful in our e-commerce context, where variability is high and sample sizes may be uneven across all of the different product categories. By modeling the log-transformed target and estimating posterior distributions for each predictor, we can gain better insights than other, traditional models would be able to give us.

ML Results and Interpretation

After running and tuning both our Bayesian Modeling and our Multilayer Perceptron, we found that they both gave relatively moderate results.

Firstly, our Bayesian regression model had an R-squared value of approximately .50. This means that when the model is fit on the target, *boughtInLastMonth*, it explains about 50% of the variance—demonstrating that the predictors were able to capture a pretty good part of the underlying patterns in the data, but still left a lot unexplained. So, we interpreted this as a model with very moderate predictive power.

To continue to interpret this model's performance, exploring some error metrics will help us see just how varied our predictions were and just how far off we were from the observed values. When calculating these from the test set, the model yielded a root mean squared error (RMSE) of about 1.98 and a mean absolute error (MAE) of about 1.50 on the log scale (as we had to log transform our target variable due to skewed data). Once the errors are transformed back into a non-log scale (the original data scale), we can look at these errors as telling us that an individual prediction can be off by a factor of about 3 to 5—which is not small. The magnitude of this error shows us that the model can see and capture the overall trend, yet there is still a great amount of uncertainty when looking into an individual prediction. Looking at the Residual plot to the left, it illustrates the difference between observed and predicted log units sold. The points that lie below the red line are underpredicted (negative residuals), while those above are overpredicted. The downward trend implies systematic bias at different ranges of log sales, where higher predicted values tend to have increasingly negative residuals. With this bias, it is easy to see that the model tends to underestimate sales for higher-selling products and overestimate at lower-selling ranges.

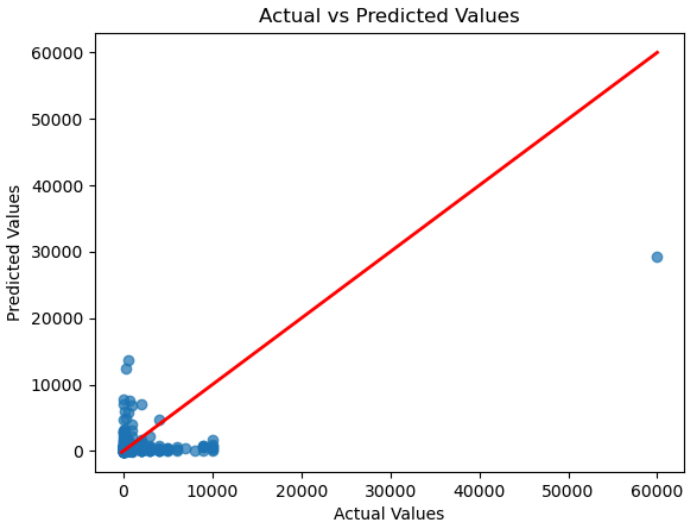


Interpreting our results in terms of coefficients, the model indicates strong positive relationships among some features (and specific categories) as well as some strong negative relationships among some features. For example, with the feature *isBestSeller*, products that are bestsellers are predicted to have sales that are much higher than those that are not. For the *price* feature, the coefficient is surprisingly close to 0, showing a very weak, if any, influence on the target variable. Additionally, different categories had a different influence on the target variable. Certain categories (e.g., Health & Household, Kitchen & Dining, Bedding) have markedly high positive coefficients (4–7 on the log scale), pointing to substantial increases in expected units sold relative to the baseline category. Certain categories (Health & Household, Kitchen & Dining, Bedding) have quite high positive coefficients (4–7 on the log scale), pointing to large increases in expected units sold relative to the baseline category column. The variability of the

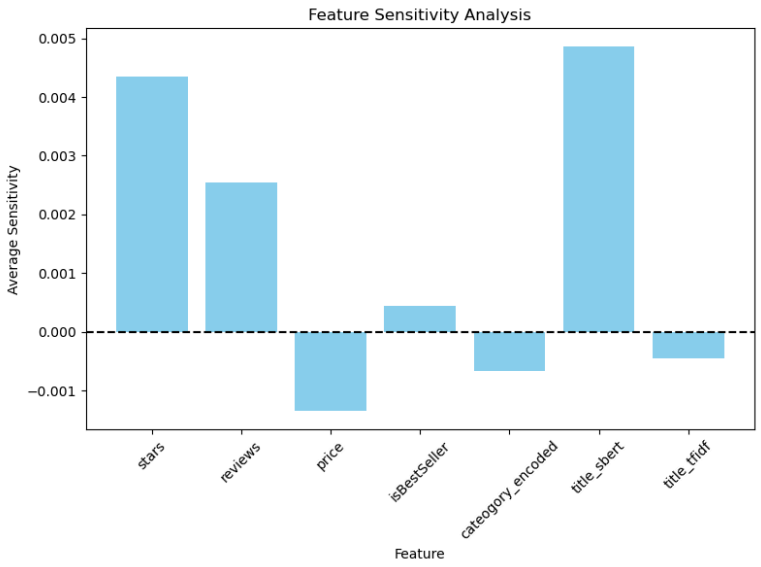
coefficients for each category is shown in the image below. There are many more categories in the dataset, but this image highlights how category type can have a complex effect on the number of units sold and therefore impacts our models. To summarize, the results of the Bayesian model allow us to further understand how the features impact performance and expose areas of improvement for future work.

categoryHairCareProducts	5.88
categoryHardware	0.94
categoryHeadphones&Earbuds	-0.48
categoryHealth&Household	6.92
categoryHealthCareProducts	4.56
categoryHeatingCooling&AirQuality	1.24
categoryHeavyDuty&CommercialVehicleEquipment	-0.29
categoryHomeAppliances	2.64
categoryHomeAudio&TheaterProducts	-0.71
categoryHomeDécorProducts	4.57
categoryHomeLighting&CeilingFans	-0.28
categoryHomeStorage&Organization	2.68
categoryHomeUseMedicalSupplies&Equipment	4.04
categoryHorseSupplies	0.21
categoryHouseholdCleaningSupplies	3.82
categoryHouseholdSupplies	6.75
categoryHydraulicsPneumatics&Plumbing	-0.38
categoryIndustrial&Scientific	4.69
categoryIndustrialAdhesivesSealants&Lubricants	-0.22

Looking more into the results of our manually implemented MLP, we can understand that similar to before, the model has moderate predictive power, but there is still lots of room for improvement. The model had an R squared value of .39—meaning that the model can explain 39% of the variance in the target variable. In terms of looking at error metrics, we saw the model achieve a Mean Squared Error (MSE) of about 1,021,124, a Root Mean Squared Error (RMSE) of about 1010.51, and a Mean Absolute Error (MAE) of approximately 279.72. So, although that model did alright, we believe that the great variability in predictions is driven by extreme values/outliers. Seen to the right, the Actual vs. Predicted Values Plot shows how our model did with predictions a bit more clearly. The red, diagonal line is what we see as “perfect” or “ideal”, and we can see some points fall around there. However, we can still see some dispersion and can see how some outliers were predicted not very well. This showed us that some tuning of the model and future steps would be necessary to improve the model. Looking more into the impact of coefficients, we wanted to see how specific features were influential in driving predictions.



Below, we can see the feature sensitivity plot we created to better understand this. The Feature Sensitivity Analysis provides valuable insights by highlighting variables that are influential to the predictions. Some findings we see are that the *stars* feature has the largest coefficient (0.076), showing that it had a significant positive correlation with units sold. Another finding we think we should call out is the fact that *price* had a low negative value, which directionally makes sense. However, it showed less impact than we thought. Moreover, we believe this is tied to the weak title analysis, leaving *price* with a lower effect without the item context. Overall, through performance metrics and various visualizations, there is a solid foundation in our current approach, yet further improvements can be pursued through many avenues which are outlined further in our future work section.



Conclusions and Future Work

To conclude, our study reveals that the Bayesian model and the multilayer perceptron (MLP) can extract some meaningful insights from our observed data, with strengths and obvious limitations. Our Bayesian approach captured key determinants of product performance, showing us moderate predictive power and error metrics demonstrating that predictions vary greatly. This method demonstrated well that the feature *isBestSeller* has a strong positive influence on our target variable, while the impact of price within our observed range remains minimal—which was incredibly surprising to us. On the other hand, the model rests on many assumptions—such as a log transformation to stabilize the variance in our target feature, the linear relationships between predictors and the transformed target, and reasonable priors. These assumptions, along with a few patterns in the residuals, show unexplained behavior/trends that we think could be due to external factors or nonlinear relationships.

Looking at the next model, our MLP approach explained around 39% of the variance in the data, which is a bit less than our Bayesian model which had an R-squared score of about .50. We believe that the MLP model benefited from our preprocessing. This included feature scaling, one-hot encoding, and regularization, as well as using TF-IDF vectorization and Sentence Transformers on the *title* feature. This processing on the *title* feature helped us try to capture the overall meaning of the title regardless of the exact wording (since each instance is one log string). Also, we thought that the MLP approach was highly sensitive to hyperparameter choices, which made it prone to overfitting or underfitting.

Looking ahead to future steps, we can see lots of promising paths for future work to enhance our analysis and hopefully overcome some of these limitations. First, we thought of integrating the prediction of an optimal price to try to help the predicted number of units sold—which could provide more actionable insights for vendors. One of the main limitations of our current models is the lack of features that explain the observed data. We think that the expansion of the feature set by integrating things like competitor pricing, seasonal trends, or even market sentiment data could help greatly. We believe that these additional predictors can better reflect real-world influences on product performance, supporting both the Bayesian and MLP models. After improving each model, we think the development of a user interface for a web application would be incredibly helpful to anyone working with inventory—particularly in e-commerce. We see this tool including interactive elements like sliders, allowing vendors to tailor inventory and pricing strategies in real-time.

All in all, while our dual-method analysis validates the effectiveness of both approaches in capturing essential but broad trends, it also emphasizes the need for further refinement. By extending our model to incorporate a richer set of features, leveraging detailed product descriptions, and introducing interactive tools for vendors, we hope to bridge the gap between data-driven insights and practical application—which has always been our goal.

Works Cited

Abigail, M. (2025, February 25). Optimizing E-commerce Supply Chains with Predictive Analytics [Article]. ResearchGate. https://www.researchgate.net/publication/389357716

Jakkula, A. R. (2023). Predictive Analytics in E-Commerce: Maximizing Business Outcomes. Journal of Marketing & Supply Chain Management, 2. https://doi.org/10.47363/JMSCM/2023(2)158