The University of Texas at Austin
Texas McCombs
MS Business Analytics
McCombs School of Business

tradeblock

Final Report

Tradeblock

Group 8

Trade Offer Scoring System

Business Analytics Capstone Project, MSBA

Apurva Audi, Ryan Lee, Joe Morris, Aakash Talathi, Jiaxi Wang
McCombs School of Business, University of Texas at Austin
MIS 382N: 2023 MSBA Capstone
Professor Robert Hammond & Professor Daniel Mitchell
April 30, 2023

Table of Contents

Executive Overview

Tradeblock is an eCommerce platform in which users can barter with each other in order to trade pairs of sneakers. Given the niche market that is geographically widespread, this app helps to collect shoe enthusiasts and collectors alike, giving them a community where they can trade and discuss various sneakers. As a newly started business Tradeblock currently does not have an internal trade valuation system, where they are able to determine the value of a trade as compared to market value or use a system to determine how likely a potential proposed trade is to be accepted. A current issue that the app is facing is the low acceptance rates of proposed trades. By our estimates, less than 2% of all proposed trades are being declined or countered. This leads to a negative user experience, in which one has to work increasingly hard to find a user that will accept your barter. Tradeblock aims to capture various predictors in order to be able to better suggest trades to users, increasing both the percentage of accepted trades and therefore the customer trade experience. Our group will provide Tradeblock with multiple models capturing this relationship as well as a study on the various indicators that are most important in predicting the status of a trade.

In the following report our group of analysts dives into the various factors captured within the trade profile in order to better understand how to best analyze and predict a future trade based on indicators. We also seek to understand the dollar disparity between trades, and how an extra dollar would affect the likelihood of a trade being accepted. These recommendations would greatly benefit the Tradeblock organization’s understanding of the users implicit trading habits and would prove beneficial in implementing suggestions and promotions alike.

Business Outcomes

Business Context

The sneaker industry is booming, and sneaker collectors are always on the lookout for the next rare pair of shoes to add to their collection. For many, acquiring new shoes can be an expensive and complicated endeavor as they rely on the secondary market to obtain their desired items. This is where Tradeblock comes in. Tradeblock is a social marketplace that allows sneaker collectors to connect and trade their sneakers without having to pay exorbitant resale prices. Tradeblock was designed to replicate the real-life experience of trading sneakers at a swap meet or convention but on a much larger scale. By bringing millions of unique collections into one place, Tradeblock aims to be the ultimate online destination for sneaker enthusiasts. The platform is looking to implement a trade offer fairness scoring system which is designed to improve the rate at which trades are accepted, as it provides transparency on the likelihood of a trade going through. This helps users of the platform to make better-informed decisions and increase the odds of a successful trade. Given the ever-growing popularity of sneaker collecting, Tradeblock’s platform has the potential to revolutionize the industry. With its emphasis on efficiency, transparency, and convenience, Tradeblock can provide an invaluable service to sneaker collectors of all levels. Not only does it make it easier and safer for them to acquire their desired shoes, but it also allows them to use their "kicks as currency" and avoid ridiculous resale prices. By providing a valuable service to a passionate and engaged audience, Tradeblock has the potential to become a leader in the sneaker trading industry.

Business Questions

This project is focused on launching a "trade offer scoring system" model to improve the success rate of trades on Tradeblock. In order to ensure the success of this project, there are several key business questions that need to be addressed:

What factors are most relevant to predicting successful trades?

The factors that lead to trade offers being accepted at a higher rate are important for Tradeblock to understand in order to improve its current accepted trade rate of less than 2%. To determine the factors that influence the acceptance of trades, data exploration, and analysis will be conducted to identify the relevant variables.

What is the relative "weight" of each factor in the model?

After the factors that influence the acceptance of trades have been identified, the next step is to determine their relative "weight" in the models. This will allow Tradeblock to determine which factors are most meaningful and impactful in predicting the acceptance of a trade offer.

How can Tradeblock use the trade fairness scoring system to improve the success rate of trades?

The trade fairness scoring system will be used to predict the likelihood of a trade being accepted based on relevant data and parameters about the offer. This will help Tradeblock improve its trade acceptance rate by ensuring that customers receiving a trade offer feel the market value of the offer is fair. Users proposing the trade offer will also be encouraged to make more generous offers to improve the likelihood of their offer being accepted.

How can Tradeblock measure the success of the project?

The success of the project can be measured by tracking the improvement of the acceptance rate after implementing the trade offer scoring system visualization into their app's user interface. This can be done by comparing the trade acceptance and rejection rates before and after implementation. Additionally, user feedback and messages between users can be used to better understand customer preferences and behavior, which will help Tradeblock to further optimize the trading experience. By addressing these business questions, an effective offer scoring system will be developed to improve the acceptance rate of trades, as well as create a more efficient and fair trading experience for users.

Business Results

The outcome of this project is to provide a trade offer fairness scoring system that will improve the acceptance rate of trade offers and the turnaround time to receive an effective offer. Specifically, we hope that by implementing our scoring system, Tradeblock will increase the offer acceptance rate by 15% and reduce the turnaround time to receive an effective offer by 2 days. These targets have been set in an effort to achieve a positive correlation between trade offer score and acceptance, as well as to improve offer ratings to significantly increase the acceptance rate. To achieve these goals, our team will extract data relevant to this project from Tradeblock's data warehouse using their Trade and User entity relationship diagrams (ERDs). The successful implementation of the trade fairness scoring system will result in Tradeblock maintaining its competitive edge in the global sneaker resale market and providing a more efficient sneaker trading platform for users. This will enable Tradeblock to capitalize on the projected growth of this market, which is expected to reach $30 billion by 2030. Furthermore, it will improve user engagement and retention, as well as boost profitability.

Explore Source Data

Source Data Available

The client provided us with a cloned version of their database for the project as of December 2022. The main features of the data that they made available were the shoes users own, what shoes users would like to own, how willing users are to trade shoes, the market value of each sneaker, historical trade data, and unstructured direct messages. Their data had two distinct sections: users and trades. The users segment had 24 tables with linking tables between each one. These tables held information about the users and their characteristics, such as the shoes they own, the sneakers on their wishlist, and other generic user settings data. The trades segment had 18 tables with linking tables in between them. The trades portion had information on trade proposals, statuses, shipping information, and other relative data for facilitating a trade between two users. After aggregating and engineering all this data, we can leverage it to train our offer scoring system.

Business Description

The data from our client represents the users who use the Tradeblock app. Tradeblock is a sneaker trading platform that allows individuals to use their sneakers as currency. When users sign up for an account they can create their virtual closet on the app and initiate trades with other users using these shoes as their assets. Tradeblock aggregates this closet data on their end to allow users to source the shoes they want from other users. These actions require Tradeblock to store information on the shoes an individual owns, such as the condition, size, and brand, as well as the user's wishlist of sneakers they are hopeful to trade for. Since the platform is a marketplace for trading shoes, they also have thousands of data points describing the trades that have been initiated, confirmed, and denied. This information can be utilized to better understand how people are using their platform. The data can help reveal information such as the preferences of customers and how "value" is perceived for a given shoe by a given user. Their hope is that they can leverage this existing data to create an offer scoring model that will lead to users offering more realistic trades that will be accepted more frequently.

Issues and Limitations

The dataset that was provided to us has a few issues and limitations that impacted our model development. Firstly, there is an imbalance in the data due to the fact that a majority of the interactions were unsuccessful trades as seen in **Figure 1** below. This leads to a skewed distribution of the target variable, which would negatively affect the performance of the machine learning models we plan to use as a component of our trade-fairness scoring system. Additionally, due to the sheer number of interactions, querying the dataset takes much more time and can significantly slow our analysis and preparation of the data for modeling. The kernel has "died" in the Jupyter notebook many times while working, which requires the SQL query to be run again to continue working. The SQL query can take anywhere from 30 minutes to 1 hour to complete.

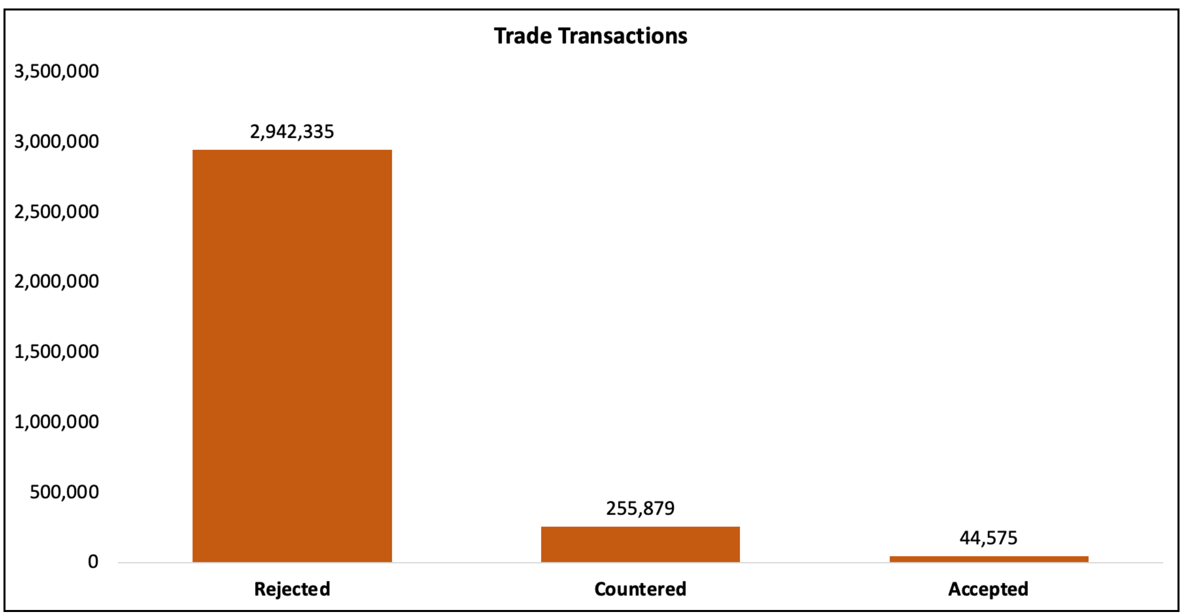

Figure 1: Trade Transaction Imbalanced in the Dataset

Furthermore, many of the market prices of the shoes are NULL in the dataset, which is critical when it comes to creating a trade-fairness scoring model. We plan on only keeping trade offers that have shoe prices for all shoes involved in the trade for both users. Our team believes this is the best way to approach this issue since Tradeblock has fixed the inconsistent recording of shoe market prices after they had cloned their database for this project. An updated version of the database with these market prices can be implemented in the future to improve the performance of the models.

Data Engineering

Data engineering and transformation activities were needed to prepare the data for analytics. Feature engineering was used to create a predictor that is the overall trade value of the shoes for each user. Our team decided that this was necessary for our model since the number of shoes in the trade offer can differ. For example, one trade proposal between users could be one shoe for one shoe, while another could be three shoes for one shoe. In order to have a consistent scoring function for every proposed trade, our team will calculate the overall trade value of the shoes for each user by taking the market price of each shoe offered by the user, discounting the price of each shoe based on its condition, and summing these values together to get an overall trade value for the shoes. For example, a pair of shoes in perfect condition with a market price of $200 would not be discounted since it is still in perfect condition, while a used pair of shoes with a few scuff marks would be discounted by some factor. The cash that is added on top of the trade by each user involved in the interaction will also be included in this value.

Additionally, an average trade willingness value will be included. Tradeblock has three classes for trade willingness: “Unwilling to trade”, “Open to trade”, and “Willing to trade.” Since the personal value of a pair of shoes may differ from user to user, we decided not to include this as a multiplier or discount factor for the trade value of the shoe, but instead take the average trade willingness as a separate predictor. Our hope is that this predictor will capture the overall personal value that users have with the shoes being given up in the trade proposal.

The inclusion of the original box for each shoe and its condition will also be predictors in our model. We will not include this in the overall trade value of the shoes because the shoes themselves are the key items of the trade and the value that users place on the inclusion of the original box and its condition will vary. Serious collectors may highly value receiving the shoes in the original box in good condition, while casual collectors really just want the shoes themselves and the original box would not affect the value of the shoes for them. Similar to overall trade value, the inclusion of the original box is discounted based on the condition of the original box. The average of these values will be taken for each trade proposal because we do not want to give users with more original boxes in poor condition a greater value than a single box in perfect condition.

The predictors stated above will be separated for each user to capture information about what each user is offering in the trade proposal. All of these steps were taken to maintain a consistent scoring function across trades with different numbers of shoes involved. The scoring function output will then be used as the input of a linking function to predict the probability that the trade offer will be accepted by the other user.

Analysis

To perform our classification modeling task, we have adopted a comprehensive preprocessing pipeline to ensure that the raw data extracted from Tradeblock’s data warehouse could be transformed into a format suitable for model training and testing. The following sections of this report focus on preprocessing, dealing with the large class imbalance in the data, and evaluating the performance of several machine learning algorithms for predicting the likelihood of trade offer acceptance.

Step Number	Step	Description
1	User-Specific Variable Differentiation	Differentiate columns between the user making the trade proposal (*user_x*) and the user receiving the trade proposal (*user_y*) by adding an "x" or "y" suffix to indicate which user is associated with the column.
2	Filtering and Cleaning for Unique Interactions	Filter out rows with no market price available for the shoes. Create a unique identifier for each interaction between users to ensure that *user_x* information is not presented as *user_y* and vice versa.
3	Variable Data Type Transformation	Transform market prices of the shoes and the amount of cash added to the offer to the data type "float."
4	Calculate the Difference Between Trade Values	Calculate the total value of the shoes that each user is offering in a trade interaction. Add the amount of cash each user added to their trade offer to the total value of their shoes. Subtract the total sum of *user_y's* offer from the total sum of *user_x's* offer to get the difference in trade value.
5	Create a Numerical Target Variable	Convert "*latestInteraction*" to string. Convert strings "rejected" and "countered" to 0. Convert strings "accepted" to 1.
6	Filter Out Bad Offers	Remove rows in the data where the difference in trade values was greater than 2000 and less than -2000.
7	Convert Interaction-Specific Data into Single Row	Group interactions by their interaction ID and create a new data frame with unique values for "*latestInteraction*", "*trade_val_diff*", and "*wishlist_matches*". The new data frame also includes mean values for each user's "*tradeWillingness*", "*hasBox*", "*hasAccessories*", and "*size_match*".

Table 1: Preprocessing Steps used for Model Analysis

Using the pipeline shown in **Table 1**, we preprocessed the raw data from querying Tradeblock's data warehouse into a format ready for training and testing our models. These steps were necessary to ensure accurate and meaningful analysis. Step 1 helps to clearly distinguish which user is associated with which column. *User_x* is the user making the trade proposal, while *user_y* is the user receiving the trade proposal and making the final decision to reject or accept a trade. Step 2 prevents inaccurate data and duplicate information from being included in the analysis. Step 4 allows us to find the difference in trade value between the two users. *User_y*'s total trade value is subtracted from *user_x*'s total trade value in order to make this predictor positive if *user_y* was gaining value from the trade and negative if *user_y* was losing value from the trade. Step 5 creates a numerical target variable for our models to train and test on. Step 6 ensures that the data is not skewed by extreme outliers. Finally, step 7 creates a dataframe where each unique interaction is a single row and the information about the interaction is summarized. This final step was necessary in order to maintain the same number of predictors for each interaction since the number of shoes offered by each user can vary between interactions.

Predictor Variables

The predictors used to train and test our models are below in **Table 2**. These predictors are divided into two categories: user-specific and single-value. User-specific predictors capture information that is specific to the shoes each user is offering, such as their willingness to trade the shoes, whether the shoes come with their original box and accessories, and the condition of the shoes. Single value predictors capture information about all the shoes involved in an interaction, such as the difference in total trade value, matching of shoe sizes, and wishlist matches.

Predictor Variable	**Description**	**User-Specific or Single Value**
Trade Value Difference	Difference in total trade value between *user_x* and *user_y*	Single Value
Trade Willingness	Willingness of user to trade a shoe in their inventory	User-Specific
Has Box	Indicates if shoes have their original box	User-Specific
Has Accessories	Indicates if the shoes in the trade are the same size	User-Specific
Size Match	Indicates if the shoes in the trade are the same exact size	Single Value
Wishlist Matches	Number of shoes offered by *user_x* that are on *user_y's* wishlist	Single Value

Table 2: Predictor Variables used in Machine Learning Models

The user-specific predictors are used to help us better understand information about the shoes each user is offering and their characteristics. The single value predictors provide us with a holistic view of the interaction, giving us insight into the overall dynamics of the interaction. By combining these two types of predictors, we can create models that accurately predict the outcome of a proposed trade.

Handling Class Imbalance

Before using the techniques for handling the class imbalance in our dataset, we first split our data into a training and testing set. Splitting the data into a training and testing set before class balancing techniques are applied is important because it prevents data leakage. Data leakage occurs when information from the test set is used to create or modify the training set. This can lead to overly optimistic results on the test set since the model has already seen the data it's being tested on. By splitting the data first, we remove this issue and ensure that the model is being evaluated on unseen data. 20% of the data was used for testing and 80% of the data was used for training.

SMOTE (Synthetic Minority Over-sampling Technique) and Random Under-sampling are two of the most commonly used techniques for dealing with class imbalance in datasets. In this case, our dataset has a major class imbalance with 3,198,214 rows in the rejected class and only 44,575 in the accepted class. Applying SMOTE to this dataset would involve oversampling the minority class of accepted rows by creating synthetic data points. This process is done by finding the k-nearest neighbors for each minority class data point and randomly selecting one of the neighbors to create a new, synthetic point for the minority class. By creating these additional synthetic data points, the minority class is artificially increased so that the dataset is balanced. Random Under-sampling, on the other hand, works by randomly selecting a subset of the majority class in order to reduce its size and balance the dataset. This process can be done by randomly selecting examples from the majority class or by randomly selecting a subset of the entire dataset.

In order to determine which technique was best for handling the class imbalance in our dataset, we applied SMOTE and RandomUnderSampler to our training set then fit the models to both training sets separately. For our sampling strategy, we used SMOTE with a ratio of 20%, meaning that 20% of the training set was the minority or "accepted trade" class. Additionally, we used Random Under-sampling to reduce the amount of the rejected class to be two times the amount of the accepted class. The ROC curve for the fitted models of both techniques can be seen in **Figure 2** and **Figure 3** on the next page.

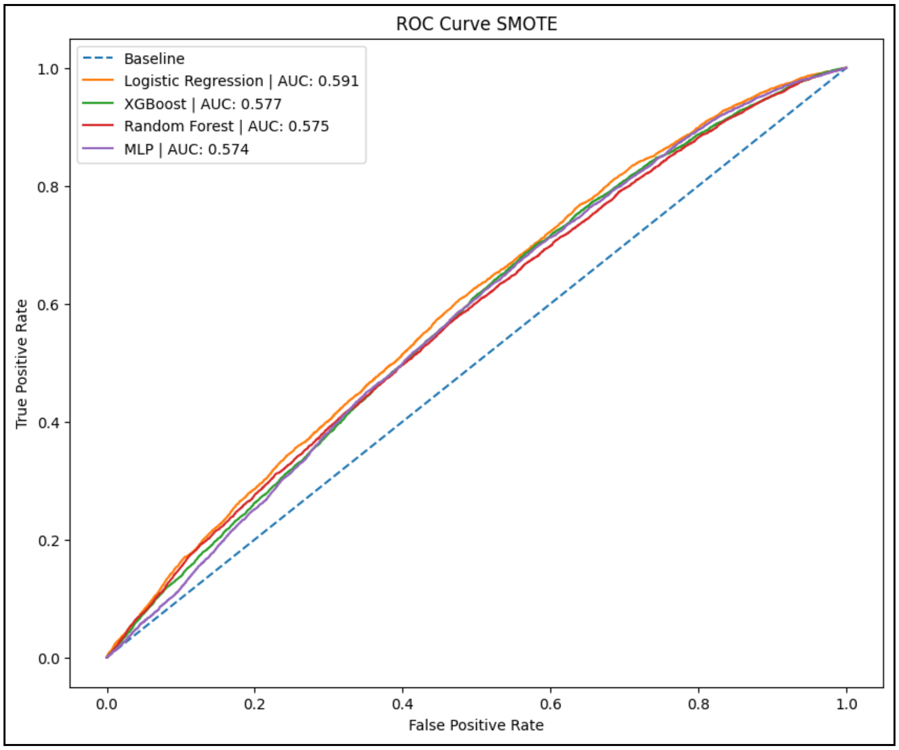

Figure 2: ROC Curve Using SMOTE

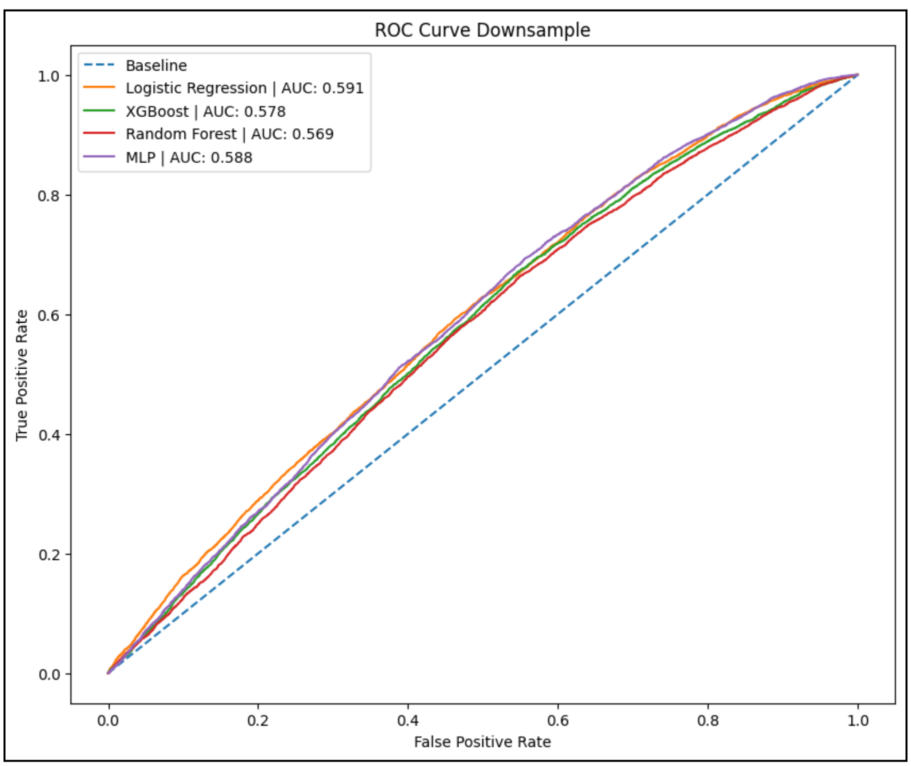

Figure 3: ROC Curve Using Downsampling

Downsampling was the better technique for handling the class imbalance in our dataset based on the performance of the different models we tried. Downsampling led to higher AUC scores for three out of the four models tested. Logistic regression and XGBoost had slightly higher AUC scores when downsampling was applied compared to SMOTE, while Random Forest and MLP had significantly higher AUC scores when downsampling was applied. This suggests that downsampling was more effective than SMOTE in improving the performance of our models. A reason for this may be that downsampling does not introduce any additional noise into the training data. SMOTE is known to introduce noise into the data in the form of duplicate or near-duplicate observations, which can lead to overfitting.

Hyperparameter Tuning

Our team tested four different models: Logistic Regression, MLP Classifier, XGBoost and Random Forest. We used a randomized search approach to train and test the model using cross-validation to find the best model and hyperparameters. Logistic Regression is a simple, interpretable model that can give a good indication of the most important features in the dataset, while also being able to provide a probability output. MLP Classifier is a powerful deep learning model that can capture complex non-linear patterns in the data, allowing it to make more accurate predictions. XGBoost and Random Forest are both ensemble methods that can be used to improve the performance of a single model by combining the predictions of multiple models. However, they are different in the way that they combine the predictions. XGBoost combines the predictions of multiple models using a gradient boosting algorithm, while Random Forest combines the predictions of multiple decision trees using a voting system. XGBoost is more powerful than Random Forest in that it can identify complex non-linear patterns in the data and can use those patterns to make more accurate predictions. However, Random Forest is less prone to overfitting than XGBoost, making it a better choice when dealing with large datasets.

Results

The best parameters, test accuracy, F1 score, and AUC-ROC using unseen exchange data are shown in **Table 3**.

Model	Test Accuracy	F1 Score	AUC ROC
Logistic Regression	0.971	0.510	0.591
Neural Network	0.896	0.490	0.588
XGBoost	0.831	0.470	0.579
Random Forest	0.791	0.460	0.569

Table 3: Best Predictive Models in Terms of Test Accuracy, F1 Score, and AUC ROC

Our three best performing models by way of these metrics were logistic regression, neural network, and XGBoost. After performing hyperparameter tuning in terms of test accuracy, f1 score and area under the roc curve, the logistic regression model was superior. The greatest advantages of using the logistic regression model is its computational efficiency and its extremely interpretable results.

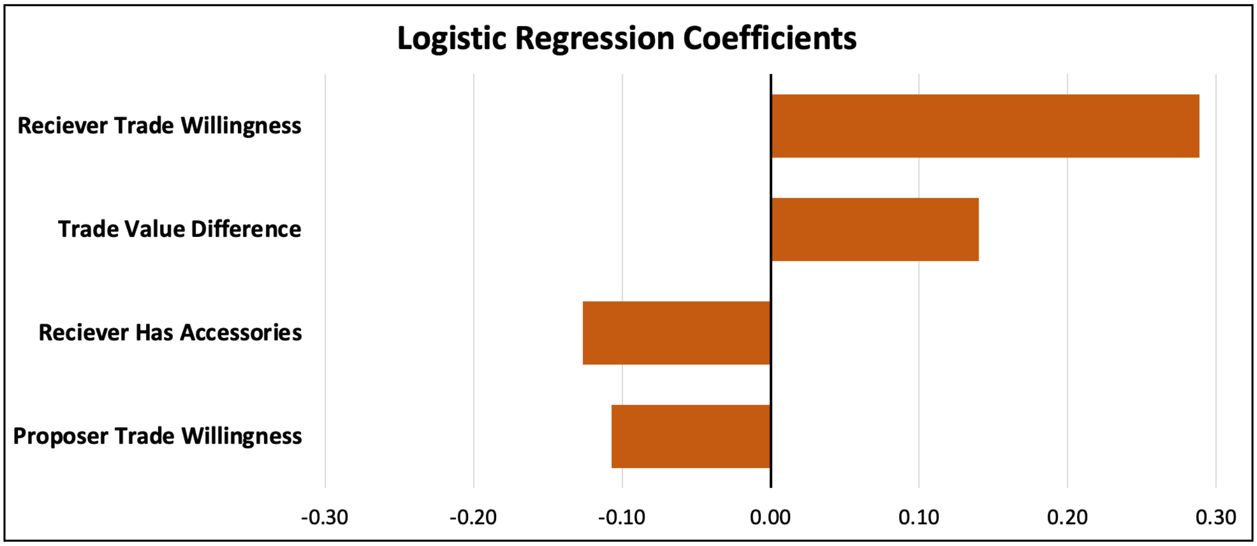

Figure 4: Standardized Logistic Regression Coefficients

Using the magnitude of the standardized logistic regression coefficients to better understand feature importance (**Figure 4**), we discovered that the receiver's trade willingness and the trade value difference were the most important predictors for determining the likelihood of a trade being accepted. This makes practical sense because the more tilted the value of the trade is in favor of the receiver and the more willing he or she is to trade, and the more likely a trade is to go through.

Recommendations, Operational Execution and Change Management

The Tradeblock Offer scoring capstone project aims to develop a metric to predict the outcome of shoe trades on the Tradeblock platform. The proposed solution involves creating an Offer Score section in the Review offer application window, which would require the creation of an API and a user interface redesign. The key predictors for trade outcomes identified were trade willingness, the difference in value, and the user's unique style and value preferences. However, the class imbalance between rejected and accepted trades presents a challenge in predicting the minority class. To address this issue, the project recommends using GPT models and prompt engineering to better understand the features of the shoes being traded. Additionally, the project suggests adding recommendations for shoes in a user's inventory that match another user's wishlist at the time of trade and understanding a user's style preferences to improve scoring.

Based on the findings and recommendations of the Tradeblock Offer scoring capstone project, we suggest that Tradeblock should implement the proposed scoring system and add the Offer Score section to the Review offer application window. The creation of an API and user interface redesign will be necessary to support the implementation of the metric. The creation of an API is a critical component of the Tradeblock offer scoring capstone project as it is necessary for the integration of the metric into the Tradeblock platform. An API, or application programming interface, is a set of protocols, routines, and tools for building software applications. In this case, the API will enable the exchange of data between the scoring system and the Tradeblock platform. To create the API, the development team will need to define the inputs and outputs of the scoring system. The inputs will likely include data related to the shoes being traded, such as brand, size, and condition, as well as data related to the user, such as trading history and style preferences. The output will be the offer score, which will provide a prediction of the likelihood of a successful trade. Once the inputs and outputs have been defined, the development team can create the API using a programming language such as Python. The API will need to be tested and validated before being integrated into the Tradeblock platform.

The Tradeblock offer scoring capstone project recommends the use of GPT models and prompt engineering to better understand the features of the shoes being traded. GPT, or Generative Pre-trained Transformer, is a type of deep learning model that has been shown to be effective in natural language processing tasks, such as language translation and text generation. In the context of the Tradeblock platform, GPT models can be used to analyze the text descriptions of shoes being traded and extract relevant features, such as brand, color, and style. This information can then be used as input to the scoring system, improving the accuracy of trade predictions. Prompt engineering refers to the process of designing prompts that guide the GPT model to focus on specific features or tasks. In the case of the Tradeblock Offer scoring capstone project, prompt engineering can be used to guide the GPT model to focus on relevant shoe features and improve the accuracy of the scoring system. Overall, we hope the use of GPT models and prompt engineering can help to improve the accuracy of the scoring system and enhance the overall user experience on the Tradeblock platform. However, the development team will need to carefully design and test the models to ensure that they are effective and reliable.

Furthermore, we recommend that Tradeblock incorporate recommendations for shoes in a user's inventory that match another user's wishlist at the time of a trade. This feature will help to increase user engagement and improve the overall user experience on the platform. Moreover, understanding a user's style preferences to improve scoring will be critical in enhancing the predictive power of the scoring system. We suggest that Tradeblock explore various techniques for collecting and analyzing user data to improve the accuracy of the scoring system.

To summarize, the Tradeblock offer scoring capstone project provides valuable insights into the development of a predictive metric for shoe trades on the Tradeblock platform. By implementing the recommendations outlined in this report, Tradeblock can improve the accuracy of trade predictions, enhance user engagement, and improve the overall user experience on the platform.

Conclusion

In conclusion, it was determined that the logistic regression model outperformed other tested models in terms of test accuracy, f1 score, and area under the receiver operating characteristic curve. This model was also extremely understandable, and helped us visualize the relationship between the predictors. As a takeaway from this report it is important to realize the advantages and limitations of such a model. Knowing the most important factors, in our case “willingness to trade” and “market price differential” allow us to explore possibilities on why trades are getting rejected, and what factors we can suggest in a trade recommender system to minimize rejected trades. However, it is important to note that due to individual preferences these are not perfect predictors and oftentimes two trades with the same predictors will have different results solely due to the individual's unique value function. In a bartering style trade, market value does not play as big of an impact in the total dollar amount exchanged from both sides. We found that while many accepted trades were near the zero sum differential mark, there were many counterexamples of goods being traded well above and below. That being said, this process is much better than the baseline in differentiating the predicted status of a trade. Therefore, we believe that this model is a great approach in order to suggest trades, and as the platform grows and more user data is available more personalized features can be added to hone in the model to account for these preferences.

References

Hills, Megan C. “How Collecting Sneakers Became a Multi-Billion-Dollar Industry.” *CNN*, Cable News Network, 5 July 2021, https://www.cnn.com/style/article/sneaker-market-high-value-auction/index.html.

Hoffman, Ken. “Machine Learning: How to Handle Class Imbalance.” *Medium*, Analytics Vidhya, 16 Feb. 2021, https://medium.com/analytics-vidhya/machine-learning-how-to-handle-class-imbalance-920e48c3e970.

Oleniacz, Laura. “More than Just Shoes: What You Need to Know about Sneakerhead Subculture.” *NC State News*, 3 Feb. 2021, https://news.ncsu.edu/2021/02/more-than-just-shoes-what-you-need-to-know-about-sneakerhead-sub-culture/.

Tradeblock. “Sneaker-Trading Start-up, TRADEBLOCK, Records Huge Growth Eight Months Post-Launch.” *PR Newswire: Press Release Distribution, Targeting, Monitoring and Marketing*, 11 Jan. 2022, https://www.prnewswire.com/news-releases/sneaker-trading-start-up-tradeblock-records-huge-growth-eight-months-post-launch-301457702.html.

Acknowledgements

The completion of this project could not have been possible without the participation and assistance of the Tradeblock team and University of Texas faculty members. We are thankful to the representatives from the Tradeblock team for their guidance. We would like to thank them for providing the necessary data and resources for this project. We would also like to express our appreciation to our faculty advisors Dan Mitchell and Robert Hammond for their encouragement and support which helped us a lot in completing this project.