Assignment: 5.2 – Predictive Analytics Case Study

Theodore Koby-Hercsky

DSC630-T301 Predictive Analytics (2225-1)

Dr. Brett Werner

April 13, 2022

**Introduction**

In this case study the use of predictive analytics comes into play for Japan’s biggest community powered shopping app called Mercari. As this app is where people can sell and buy different used and new products that range from clothes to phones from all sorts of designers. While the object of this case study is to build a model that will use price prediction to automatically suggest the price the product should be sold for.

**What was the problem being solved?**

The problem Mercari wants to solve is to have the ability to suggest the correct price to the seller before they list their items. But this is going to be a difficult task being that the seller can sell any type of product and even multiple at once. While we can use the information provided by the seller for each product that consists of the id of the product, title of the product, the condition of the product, category of the product, brand of the product, if the shipping is paid by the seller or the buyer, and the full description of the product. As this information will help build the model to predict the correct price the item should be sold for.

**Why was this problem important to solve?**

This problem is important to Mercari and their customers and sellers all over Japan because by automatically predicting the price of what the product should be sold for will help its sellers put a fair price for their items. This problem is also important to the buyers as this helps make sure the buyer is getting a fair price for the product they are trying to buy.

**How was the data acquired?**

The data was acquired from the company Mercari that is Japan’s largest community powered shopping app. Which requires users to input information on the items they are trying to sell that was listed above. As the answers from these questions make up the data that will be used to predict the price of the items they are trying to list for sale.

**Methods and Results**

**What steps were taken to prepare the data?**

The first step that was taken to prepare the data was to check for missing values as the report stated 42.7% of items did not state the brand name, less than one percent is missing the category name, and only four out of all items are missing the item description. Next, they converted all textual categorical values to lower case which will allow data that is entered as “APPLE” and “Apple” to allow the model to be the same thing. The case study used exploratory data analysis to calculate statistics and create visualizations that show different patterns and relationships from the data.

**How was this problem solved?**

The problem was solved by using feature engineering and selecting features such as brand\_name, brand\_name\_given, item\_description\_id, and more. Next, they used the one-hot-encoding to convert categorical values to numerical values. As this will take a variable that has multiple values such as yes and no and creates two new variables that has one variable for yes and one for no. While another technique that is used is bag of words that represents the item description to numerical values that will create a unique column for each word in the vocabulary. Last they established a baseline before starting with modeling which is a guess against to compare with the results. Which will determine if the machine learning models do better than the guess which will determine if we need to try a different approach if the ML doesn’t beat the guess.

**What modeling techniques were used?**

As there are a ton of machine learning models that the case study could use so a good idea is to try several algorithms and determine which ones work the best. So, they decided to evaluate four different types of models that will cover the complexity of the spectrum. As they used linear regression, decision tree regression, support vector regression, and extreme gradient boosting regression. As these will be optimized using tuning on the hyperparameters. As model hyperparameters are basically settings that are created by the data scientist for the machine learning algorithm that is used before training the models. While a random search is a technique that was used to select hyperparameters that defines the grid and will randomly sample different combinations while performing a K-fold CV. As stated, “random search performs nearly as well as Grid search with a drastic reduction in run time” while grid search “exhaustively try out every single combination.” (towards) Another technique that was used is K-fold CV which is when a data set is split into K number sections that uses folds as a testing set. As they used a 5-fold cross validation that splits into 5 folds with the “first fold is used to test the model and the rest are used to train the model” with the “2nd fold is used as the testing set while the rest serve as the training set.” (towards) While this process will be used until all five folds are used for testing. Once the random search cv is complete the test error with the lowest average is selected and will be determined the best hyperparameter combination that will predict the prices.

**Why did the team choose the methods/models they did?**

The approach that was taken when picking models in this case study was to try out several algorithms to determine which one would work the best. As when working in machine learning we are still in the experimental field as it is impossible to know ahead of time which model would work the best. That is why they decided to use multiple hyperparameters as the best set will be different for every case study meaning they will need to try different ones on the data to find the best setting. As the one with the best setting will be used to set the price of the product.

**What metrics were used to evaluate the results? Why was this metric chosen?**

The y-train mean metric was used in this case study for the baseline prediction. While also reporting back the mean squared error with the use of y-test and y-pred. As the baseline results came back as 0.7497. As this metric was chosen to evaluate the data and create a baseline model.

**Conclusion**

**How were the results or model implemented?**

The models that this case study implemented are Linear regression, Support Vector regression, XGBoost regression, and a Decision Tree regression. The Linear regression was implemented by using the SGDRegressor with the loss equaling the squared loss with a random state of 42. Which had a Linear regression error of 0.5168. Next a Support Vector regression model was created that showed the best parameters for the x-train and y-train-log. While the Support Vector regression error came back as 0.5178 which is slightly higher than the Linear regression error. The third model created was the XGBoost regression that had a max depth of 2, 4, 7, and 10 and estimators of 5, 10, 25, 50, and 100. While the XGBoost regression error is 0.5975 that is now deprecated in favor of the regression squared error. The last case study created is a Decision Tree regression model that has a max depth of 5, 10, 20, 30, and 50 that showed the best parameters of max depth of 30 while the minimum sample split of 50. While the Decision Tree error came back at 0.582 which was higher than all except the XGBoost regression error which was 0.5975. As the RMSE Root Mean Squared Error keeps the differentiable property of MSE and square roots it to handle smaller errors. Which allows for better interpretation of the error being that the scale is the same as the random forest and is less prone to outliers. Overall, the RMSE tells us that a value closer to zero will indicate a perfect fit meaning the Linear regression error of 0.5168 will be the best model to use when predicting the cost of products.

**What were the actionable consequences of the case study?**

The conclusion of this case study is that the Linear regression error was the lowest of all models used that came in at 0.5168. As this showed us that this model would be the one closest to a perfect fit meaning it would be the best to use when predicting a price for the items being sold. A consequence that could come from this case study is that users that went with a Linear regression model just because the RMSE value was lower could be missing out on the Support Vector regression error which was just a little higher at 0.5178 which was just 0.0010 more. Meaning the use of the Support Vector regression could be almost as accurate as the Linear regression which could be just as useful as the other model.

**What did the team learn from the case study?**

This case study taught the team that the use of multiple models in a report is a great thing as it will allow you to see different outcomes that can help you choose the best course of action for your problem. As this case study was focusing on the ability to predict the price of an item that was posted on the Japan’s biggest community powered shopping app called Mercari. The team learned that XGBoost regression model was the furthest from a perfect fit meaning they would less likely want to use this model in the future for other predicting case studies for prices based on answered variables.

**How should or would the team approach the problem differently in the future?**

In the future the team could approach this problem differently by adding a heat map that would check the correlation between different variables from the Mercari app. As by seeing the correlation between categorical name and brand name or brand name and the price. Next, I would say the use of a ROC AUC curve would be beneficial as it would tell us the performance of the model with a visualization as it seems like this team likes to create visualizations so the use of an ROC curve would be a great fit. The last approach that I believe they could have included the use of a crosstab that could be calculated from the model performance for actual and prediction from a prediction train.

**References**

Lobo, Geoffrey. “A Data Science Case Study with Python: Mercari Price Prediction.” *Medium*, Towards Data Science, 12 June 2021, https://towardsdatascience.com/a-data-science-case-study-with-python-mercari-price-prediction-4e852d95654.

Lobo, Geoffrey. “Mercari-Price-Prediction/mercari\_price\_prediction.Ipynb at Master · Geofflobo12/Mercari-Price-Prediction.” *GitHub*, https://github.com/geofflobo12/Mercari-Price-Prediction/blob/master/Mercari\_Price\_Prediction.ipynb.