Predicting Shipping On-time Performance

Winston Zhong

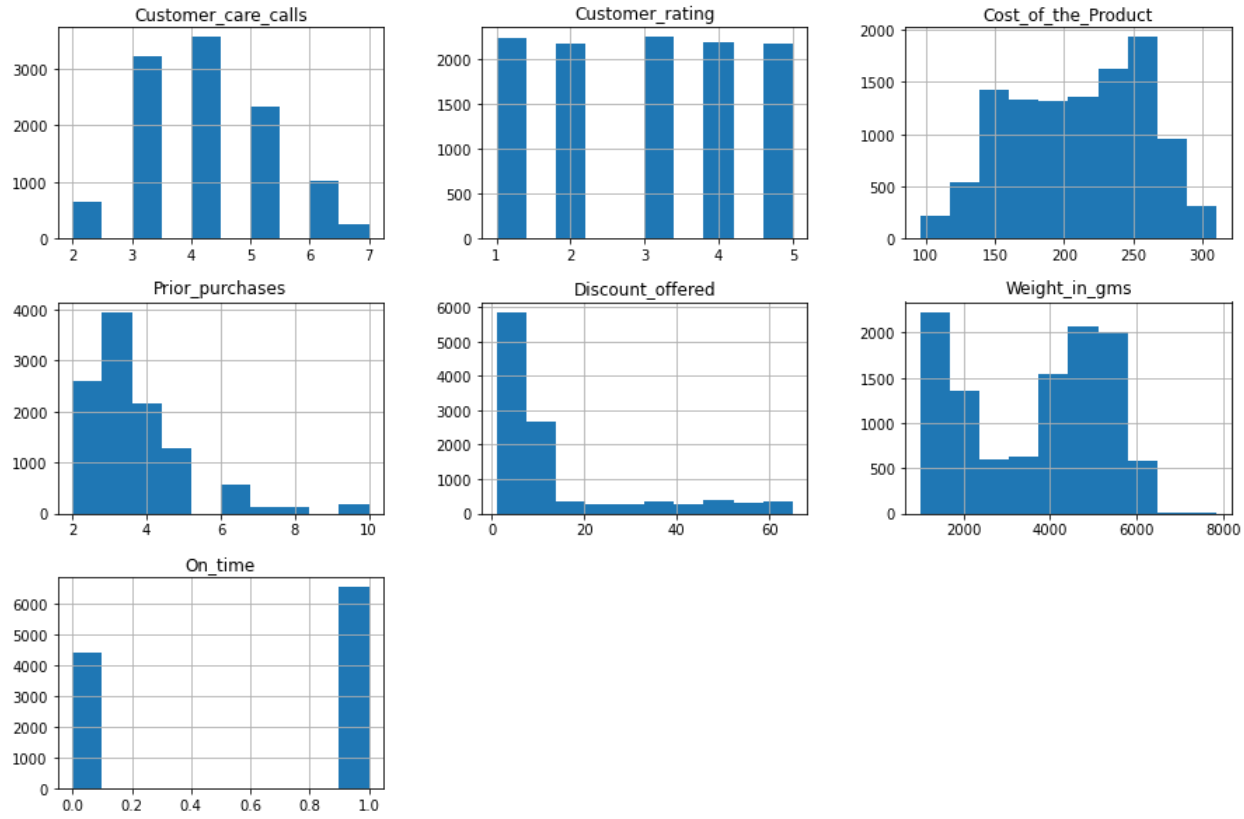
Problem Statement

E-commerce has become such a big part of almost every business that sells consumer goods. A lot of times customers are given an estimated delivery time for when their purchase will arrive, so it is very important for an e-commerce company to make sure their shipments arrive on time. This can depend on different factors, like the shipment's weight, mode of shipment, importance of the shipment, or something else not expected.

I was able to find a data set on Kaggle for an e-commerce company's shipping on-time performance along with other features on each shipment. Using this data set I wanted to find out how this e-commerce company can predict which shipments will be delivered on time and how to improve their on-time performance. To do this, I analyzed the data and developed a classification model that would predict the on-time performance of each shipment.

Data Wrangling

The data set had 12 columns: ID, Warehouse block, Mode of shipment, Customer care calls, Customer rating, Cost of the product, Prior purchases, Product importance, Gender, Discount offered, Weight in grams, and Reached on time. There were 10,999 entries and they were all either integers or strings. I removed the ID column because that didn't give us any additional information. Then I stripped any leading and trailing spaces from all the string columns. The data was already very clean so there was not much that needed to be done. I created histograms of the numeric features to see the distributions of each feature. These charts are shown below on Figure 1. For the



On_time column, 0 represents shipments delivered on time, so you can see there were more shipments that were not on time than shipments that were on time.

Figure 1

Exploratory Data Analysis

I first wanted to look at the percentage of shipments that were delivered on time based on the mode of shipment. The percentages I found were flight: 39.84%, ship: 40.24%, and road: 41.19%. The overall average was 40.33%, with road having the highest

on-time rate. This is consistent with what we saw on the histogram for the On_time column.

Next, I wanted to see what the average customer rating was for shipments that were on time versus shipments that were not on time. I found the average customer rating for shipments that were on time was 2.97 and the average customer rating for shipments that were not on time was 3.01. The difference is only about 0.04 so I wanted to do a permutation test to see if the two averages were significantly different from each other or not. Figure 2 and 3 below show the customer rating distributions for each outcome.

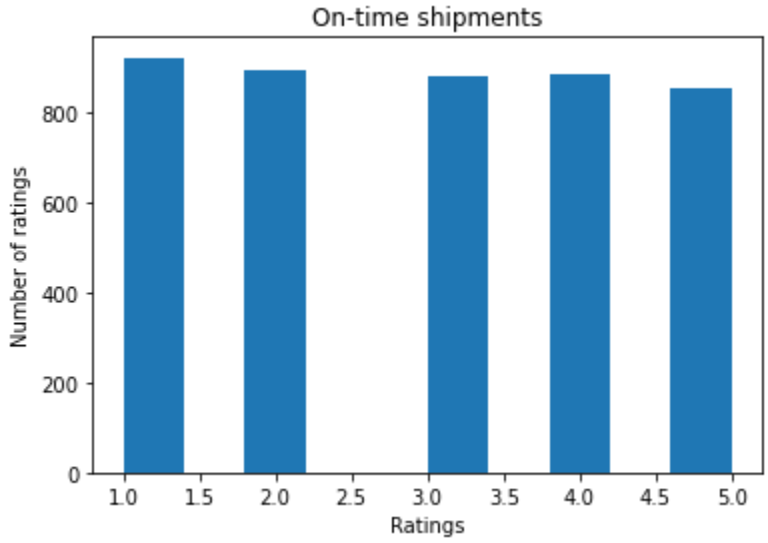


Figure 2

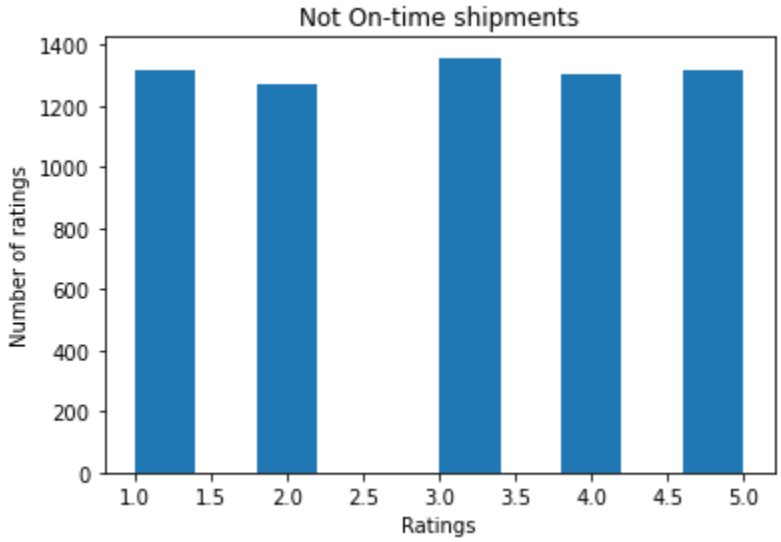


Figure 3

Null Hypothesis: The average customer rating for shipments on time is equal to the average customer rating for shipments not on time.
Alternative Hypothesis: The average customer rating for shipments on time is not equal to the average customer rating for shipments not on time.

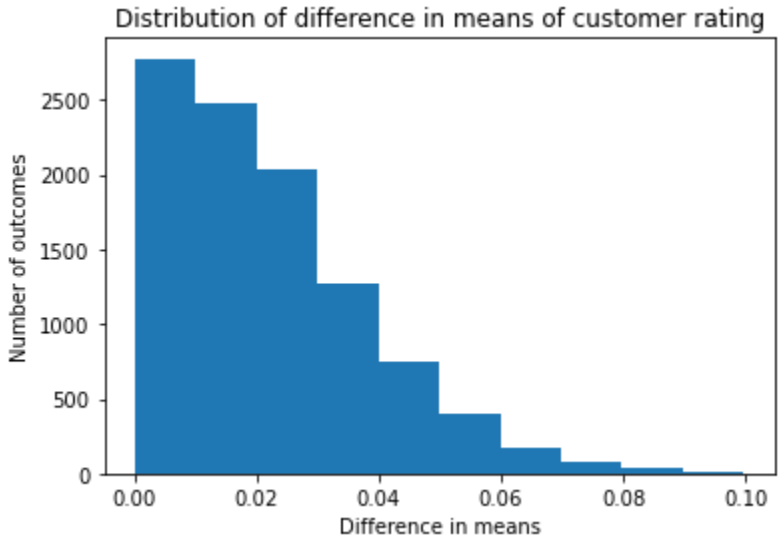


Figure 4

The above chart in Figure 4 shows the distribution of difference in means of customer rating between the two results. After performing the permutation test, we got a p-value of 0.1657, which means we fail to reject the null hypothesis. The average customer ratings between the two results are not significantly different.

Finally, I one-hot encoded the categorical columns in the table so that we would have numerical entries for all columns. After that I created the heatmap below in Figure 5 to show the correlation of each feature to each other. We can see that the Discount_offered feature has the highest correlation with our dependent variable, On_time.

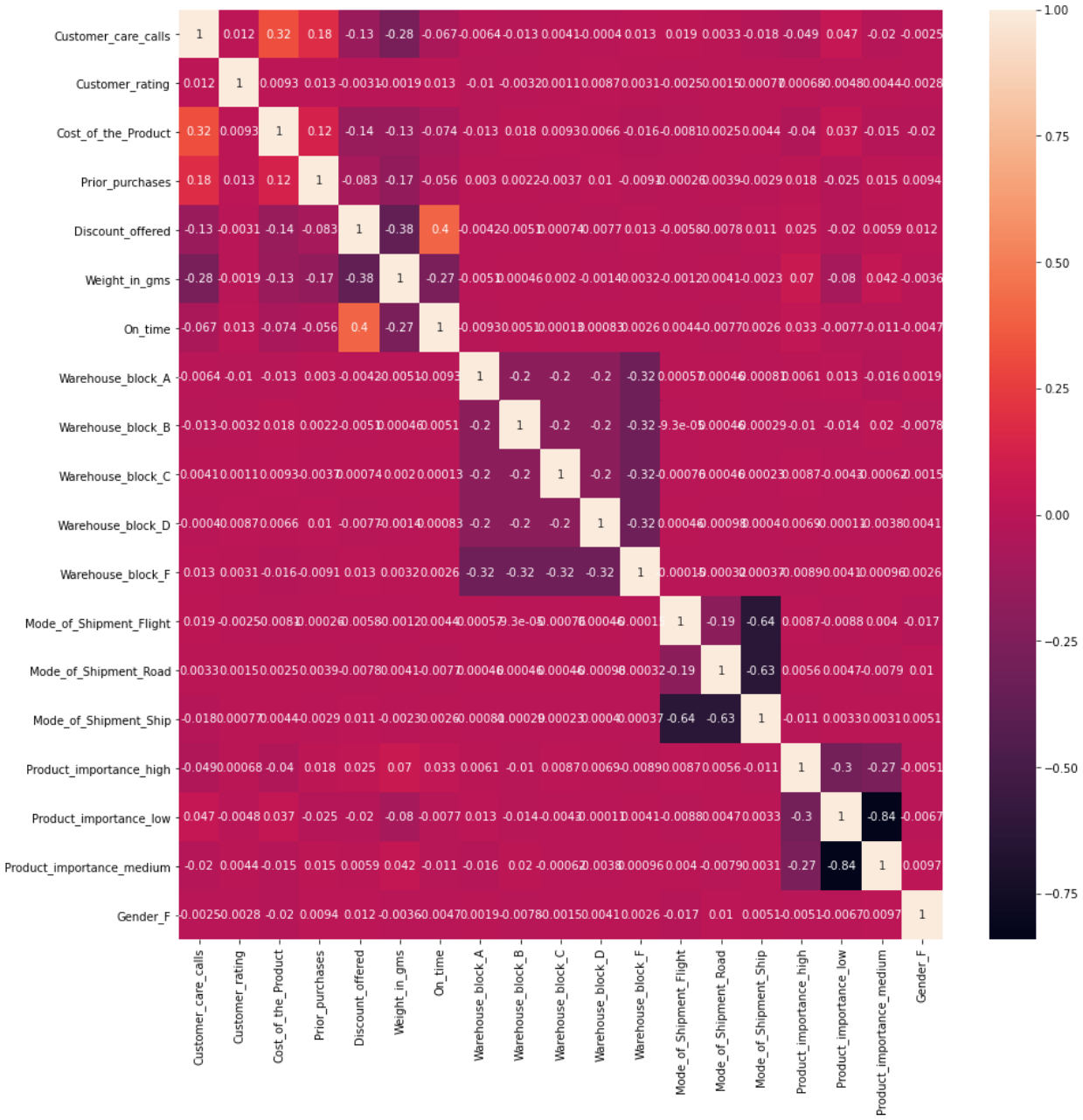


Figure 5

Pre-processing and Training Data Development

My data set is already in the form that I need it to be, so all I had to do here was a train test split with On_time as my y-columns. I did 80% training and 20% testing split. Then I used the standard scaler to scale my X data sets. Now my data is ready to be used for modeling.

Modeling

This is a binary classification problem, so I will be using the confusion matrix, accuracy score, precision, and recall to measure the fit of each model. For this problem, we want to minimize false positives, so that means when the model predicts a shipment will arrive on time but it actually does not. 0 indicates a shipment arrived on time so that means we want the highest precision for 0 we can get. I chose to test out four different modeling methods for this problem, logistic regression, k-nearest neighbors (KNN), gradient boosting, and decision tree.

Table 1 below shows the results from modeling using logistic regression. We can see here that the hyperparameter tuned model using the scaled data set gave us the best results. We'll compare this with the best results from the other modeling methods.

	Accuracy	Precision	False Positives
Default	0.63	0.53	456
Default Scaled	0.63	0.53	443
Tuned	0.63	0.53	444
Tuned Scaled	0.64	0.54	443

Table 1

Table 2 shows the results from modeling using KNN. Figure 6 shows us the accuracy of the KNN model as the number of nearest neighbors goes from 1 to 8. Accuracy for training is decreasing as K increases, and accuracy for testing is increasing as K increases. Figure 7 shows us the error rate of the KNN model for K equals 1 to 40. From this we were able to determine K=31 was the optimal number of nearest neighbors. But we see from the table that the default parameters with the non-scaled data set gives us better results. Next we'll look at the results from gradient boosting.

	Accuracy	Precision	False Positives
Default	0.66	0.57	434
Default Scaled	0.63	0.53	445
K=31	0.63	0.53	456

Table 2

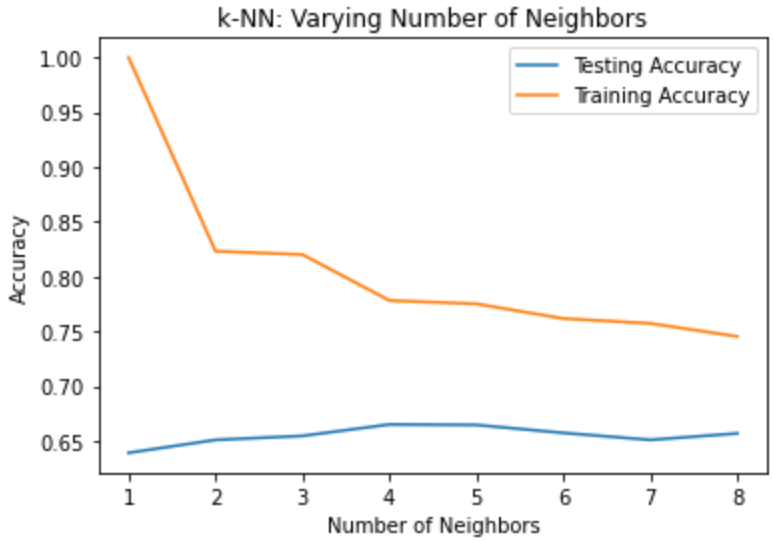


Figure 6

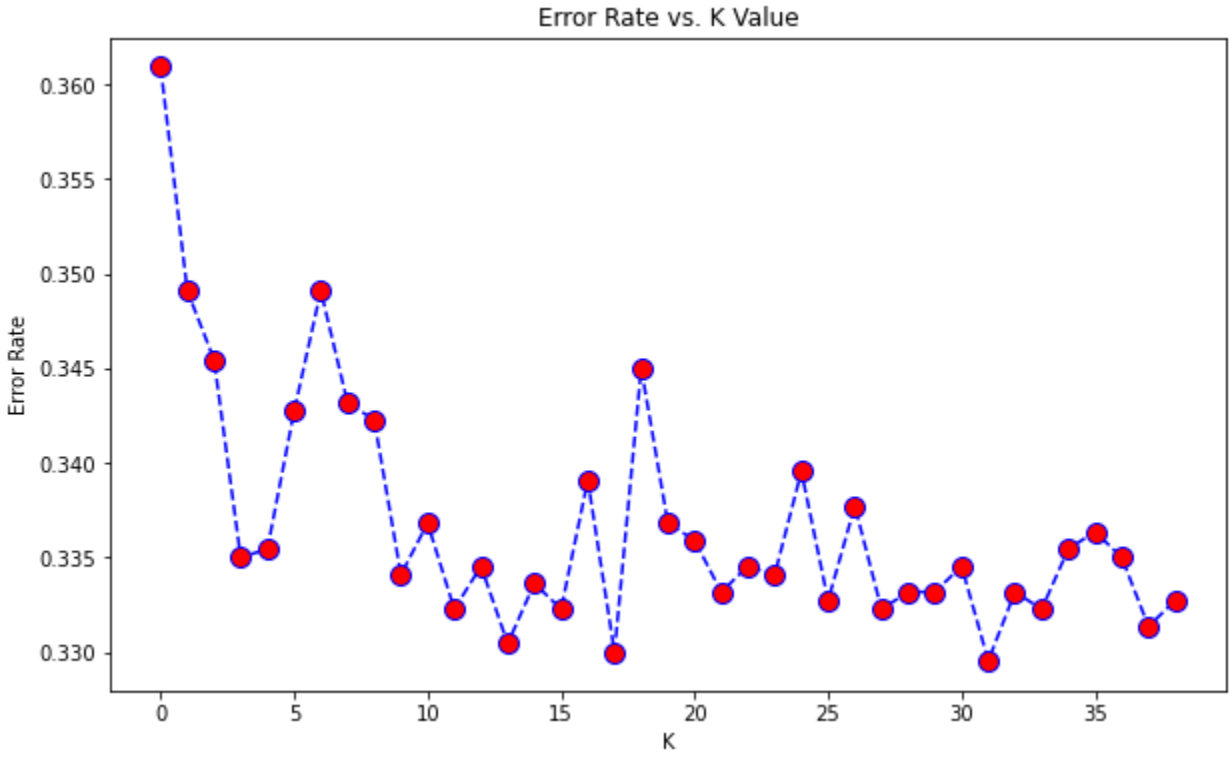


Figure 7

Table 3 shows us the results from modeling using gradient boosting. It gives us a slightly better accuracy score than KNN's default, but the precision is lower and there are more false positives. For that reason I am going to stick with KNN as the better model. Figure 8 below shows the most important features ranked from highest to least. Discount offered is shown as a much more important feature than any of the others.

	Accuracy	Precision	False Positives
Default	0.67	0.55	614
Default Scaled	0.67	0.55	614
Tuned	0.67	0.56	487
Tuned Scaled	0.67	0.55	661

Table 3

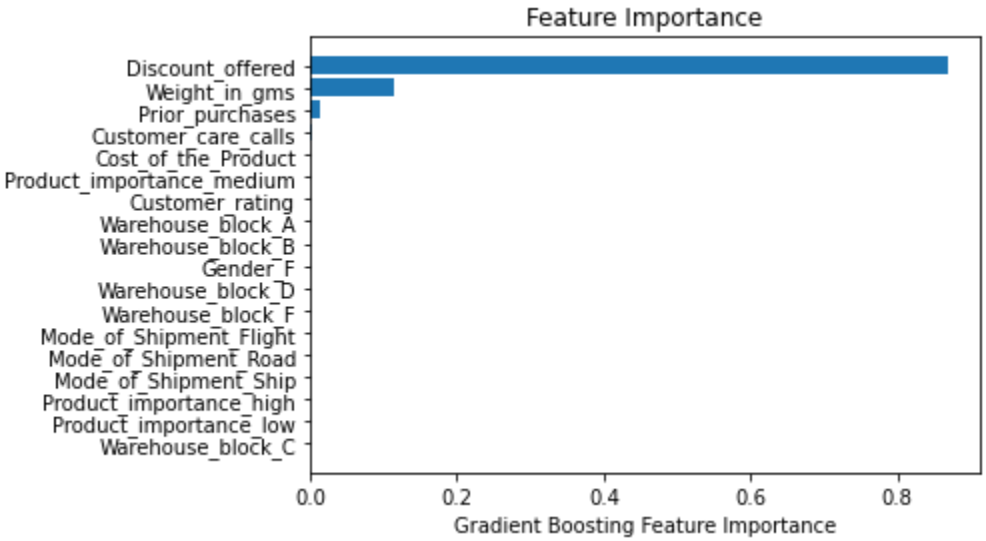


Figure 8

Finally, we have the results for the decision tree modeling in Table 4. We see the default gives us the least number of false positives, but the accuracy score and precision are not as good as what we got for KNN. We will move forward with KNN using default parameters and non-scaled data as our best model with an accuracy score of 0.66, precision of 0.57, and 434 false negatives.

	Accuracy	Precision	False Positives

Default	0.63	0.53	402
Default Scaled	0.63	0.53	402
Tuned	0.66	0.54	709
Tuned Scaled	0.66	0.54	709

Table 4

Conclusion

Using our best model, I used it to predict what percentage of shipments would arrive on time based on the importance. The results were 46.27% for high importance, 47.01% for medium importance, and 43.84% for low importance. I wanted to look at how we could improve the on-time performance for high importance shipments. I used my model to predict which warehouse block would result in a higher rate of shipments arriving on time. Figure 9 below shows the predicted on-time performance for each of the warehouse blocks. Warehouse block C has the highest predicted on-time rate while warehouse block D has the lowest predicted on-time rate. I then predicted that 20.51% of high importance shipments that were late and not in block C were placed in block D. With block C being predicted to outperform block D, I would recommend moving all high importance shipments from block D to block C to improve on-time performance for high importance shipments.

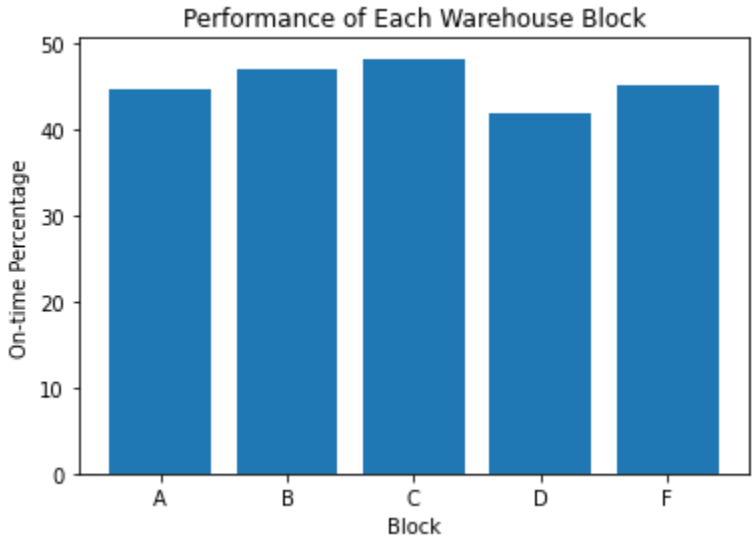


Figure 9

Next, I wanted to look at how the weight of the shipments affected on-time performance. I calculated the average predicted weights for each result and got 4,196.57 grams for shipments arriving on time and 3,170.35 grams for shipments arriving late. Figures 10 and 11 show the distribution of weights for each result. It seems that shipments weighing between 2,000 grams and 4,000 grams will almost never arrive on time as seen with the big gap in the on time weight distribution chart.

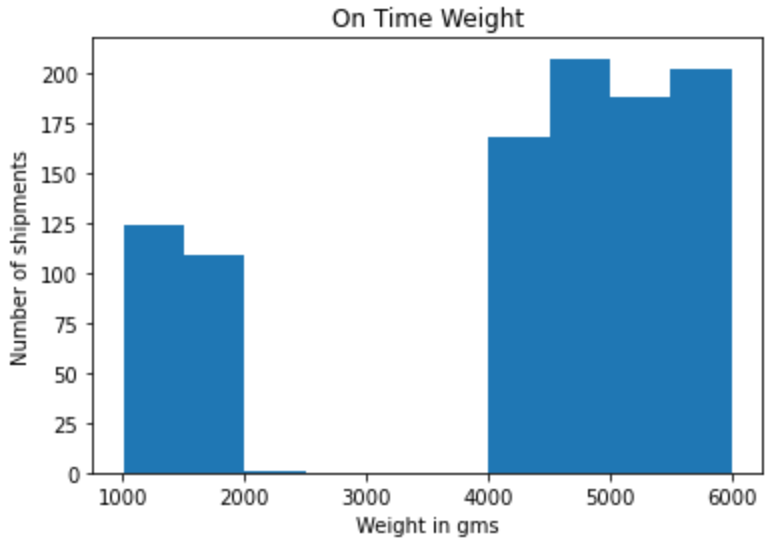


Figure 10

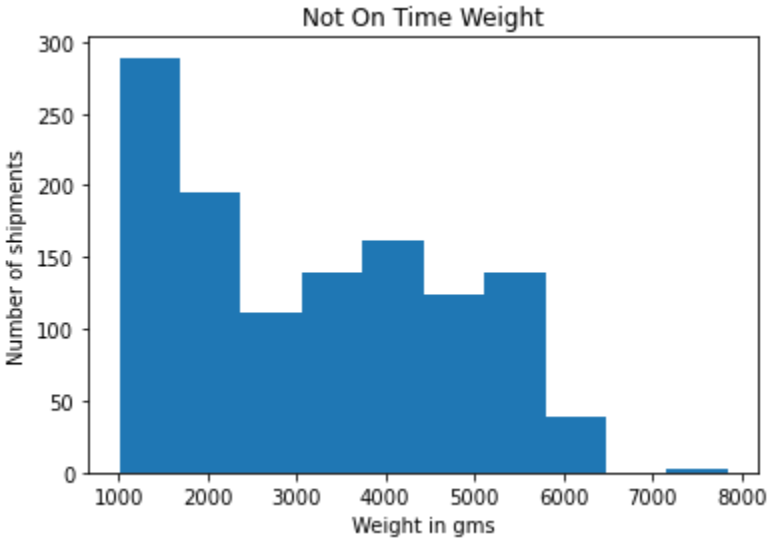


Figure 11

I took a similar approach to find out the effects of the cost of the product. Predicted average cost for on time shipments were $215.57 and predicted average cost for late shipments were $205.57. The distributions for each result are again shown on the figures below. The main difference here is there are more shipments that cost between $150 and $200 that are late than there are on time.

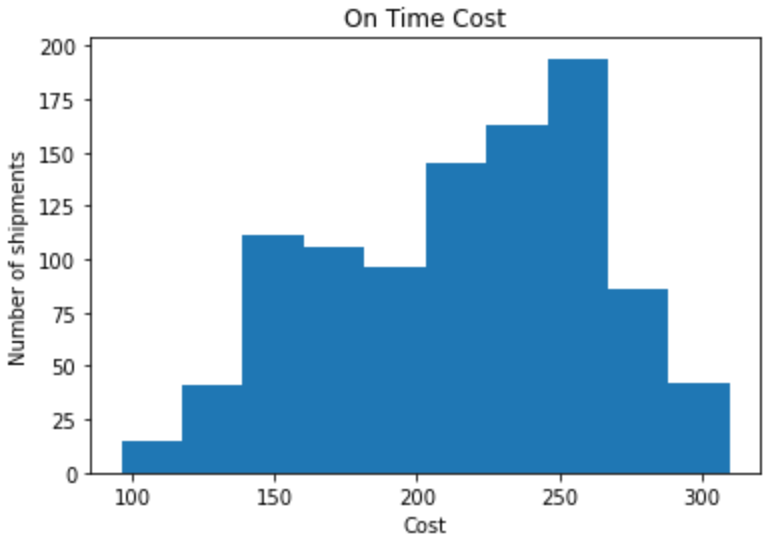


Figure 12

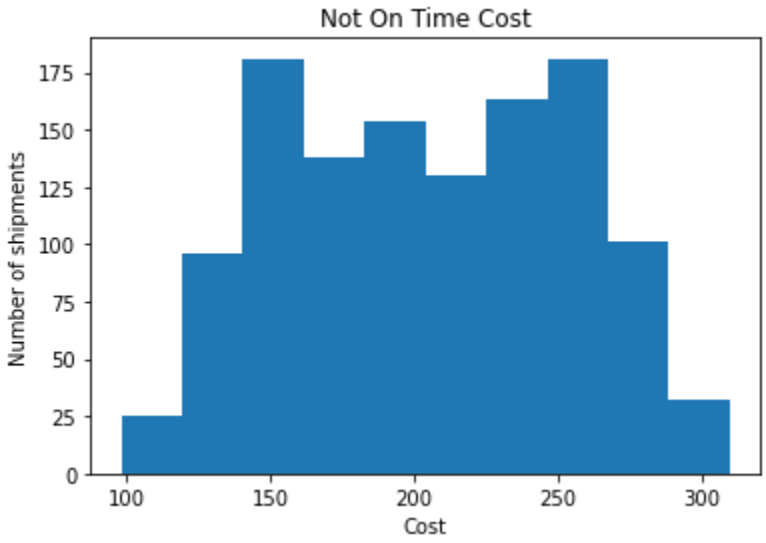


Figure 13

Finally, we saw earlier that the discount offered was a very important feature for our gradient boosting model, so I am going to see what kind of effect the KNN model

predicts it has. I found the predicted average discount for on time shipments is 5.49% and the predicted average discount for late shipments is 20.17%. Shipments that are late get almost four times the discount so this is definitely a big difference. And we can see from the distributions in Figures 14 and 15 that any shipment with a discount over 30% is predicted to never be on time. Any shipment with a discount over 12% is predicted to have a very low probability of arriving on time.

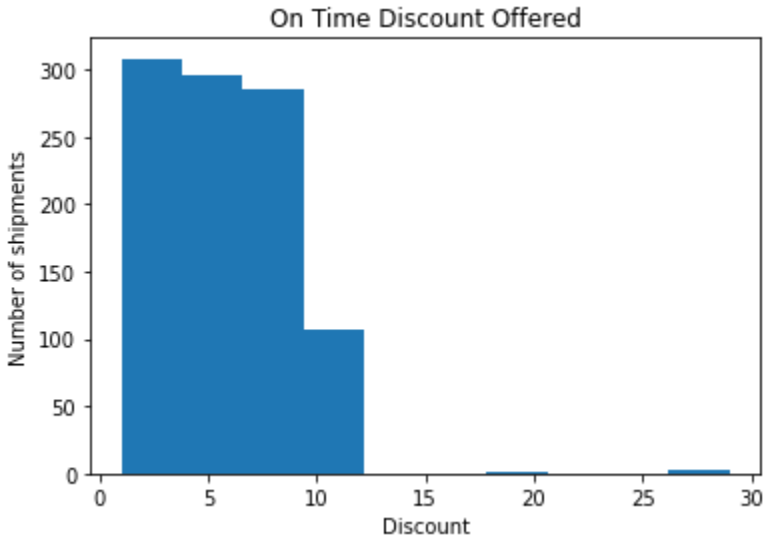


Figure 14

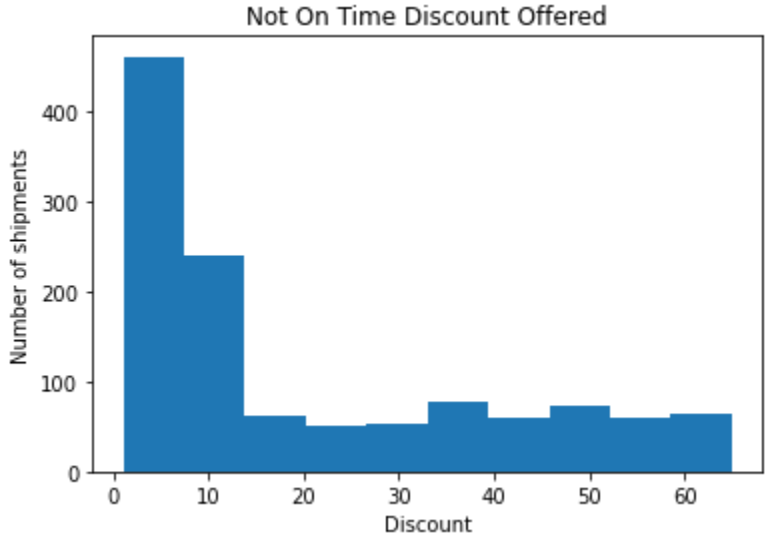


Figure 15

Further Research

Discounts offered surprisingly had the biggest effect on the on-time performance of this e-commerce company's shipping performance. This is definitely something to look into. Maybe it could be that the products with high discounts offered are products the company is looking to get rid of and have a lower importance. Other things that I think can improve this study is having more features like the distance from origin to destination and the type of products that are being shipped. Distance can have an effect on shipping performance for obvious reasons and maybe different types of products have to be shipped differently that could affect the on-time performance.