**CUSTOMER FEEDBACK ANALYSIS FOR RENT THE RUNWAY**

Yibo Feng

Contents

[1. Abstract 2](#_Toc57552644)

[**1.1** **Project Overview** 2](#_Toc57552645)

[**1.2** **Predictions** 2](#_Toc57552646)

[**1.3** **Inferences** 2](#_Toc57552647)

[**1.4** **Success Evaluation** 3](#_Toc57552648)

[2. Data Exploration 3](#_Toc57552649)

[**2.1** **Data Description** 3](#_Toc57552650)

[**2.2** **Data Preprocessing** 3](#_Toc57552651)

[**2.3** **Data Exploration** 4](#_Toc57552652)

[3. Methodology 5](#_Toc57552653)

[**3.1** **Sentiment Analysis** 5](#_Toc57552654)

[**3.2** **Clustering Analysis** 6](#_Toc57552655)

[**3.3** **Classification Analysis** 6](#_Toc57552656)

[4. Models 6](#_Toc57552657)

[**4.1** **Sentiment Analysis** 6](#_Toc57552658)

[**4.2** **Products with High Rating Analysis** 8](#_Toc57552659)

[**4.3** **Recommendation Analysis** 9](#_Toc57552660)

[**4.4** **Customer Rating Preference Analysis** 10](#_Toc57552661)

[**4.4.1 Decision Tree** 11](#_Toc57552662)

[**4.4.2 Logistic Regression** 11](#_Toc57552663)

[**4.4.3 Gradient-boosted Tree** 11](#_Toc57552664)

[**4.4.4 Conclusion** 11](#_Toc57552665)

[5. Conclusion 13](#_Toc57552666)

[6. Appendix 14](#_Toc57552667)

[**6.1** **References** 14](#_Toc57552668)

# **Abstract**

## **Project Overview**

Our main purpose of the project was to figure out how the clothes brand, Rent the Runway, should focus on utilizing the factors that could affect customer feedback and predicting which kinds of clothes should be promoted to a unique customer. To complete that object, the first thing we did was to extract the columns that can illustrate customer feedback. Then based on different data types of those columns, targeted data processing had been given. For example, a sentiment analysis was made to cope with column “review\_text” and column “review\_summary” which contain string data type. To predict customer’s rating feedback for a special product (numeric data type), we used linear regression algorithm to run a machine learning.

## **Predictions**

1. Predict the sentiment of reviews in each transaction for the purposes of finding out important factors influencing customers attitudes and preparing for the prediction of rating.
2. Predict the situation where customers are likely to give different ratings for the purpose of judging whether the customer would like to provide a high rating or not.
3. Predict what kinds of product will fit different customers for the purpose of recommending appropriate product to customer.
4. Predict what kinds product will get high ratings for the purpose of helping RentTheRunWay keep them stocked.

## **Inferences**

1. We planned to measure the prediction of customer’s sentiment in each transaction with some variables such as words in “review\_text” that weighs the most in illustrating the emotion of customers, and to predict “attitude” which directly shows feedback (positive or negative) from customers.
2. To analyze products with high ratings, we planned to use Random Forest to predict ratings that customers gave to the product, and found important features with high weighted scores according to the result of the model with the best maximum depth of the tree.
3. We were willing to see some related predictors work well in predicting what kinds of product would fit a special customer, such as “weights” and “heights”. We planned to use PCA to select important features and input them into K-means model to split transactions into several clusters. Then, we could find top features in each cluster to specify the products.
4. To predict what kinds of products that customer would like to give a rating with 10, we planned to compare the performance of decision tree, logistic regression and gradient-boosted tree with AUC to decide the best model. We could use this model to find out important features so that we were able to select specific types of products.

## **Success Evaluation**

1. For Sentiment Analysis, we got a sentiment analysis model with 88% accuracy, 10 most negative words (factors) and 10 positive words (factors). However, it was difficult to convert emoji to text when text processing with pyspark.
2. For Customer Rating Preference Analysis, we got different situations that contribute to different rating levels and selected important features.
3. For Recommendation Analysis, we got multiple background of customers, and successfully split products into three clusters to find most suitable category of clothes.
4. For Products with High Rating Analysis, we got specific kinds of products that contribute to getting high ratings using predictors in inference. We found top ten most important features after comparing models. However, the accuracy of classification models didn’t perform very well when predicting products with high ratings.

# **Data Exploration**

## **Data Description**

The dataset is collected from RentTheRunWay by Rishabh (2018) [1]. We downloaded it from *Kaggle*. The format of the raw data is .json file. The dataset provides some types of information, such as ratings and reviews from customers, fit feedback from customers after they buy clothes. Besides, the dataset also contains product measurements and category information. The data set includes 15 columns and 192544 rows. Each raw in the data records one transaction. There are 105508 customers who completed transactions and 5850 kinds of product are included.

## **Data Preprocessing**

From the data set, we found that “review\_date”, “corrupt\_recored”, “item\_id” and “user\_id” wouldn’t make sense in our analysis, so we removed them from the original dataset. As mentioned above, the format of height didn’t work well in the future. Therefore, we tried to extract numbers and transformed them into integer with units of centimeter. Besides, “weight” contained “lbs” in the value, so we removed “lbs” from the values and only kept its number. Then, we filled missing values of numeric variables with their means, such as “weights”, “heights” and “size”, and filled missing values of categorical variables with the most common values.

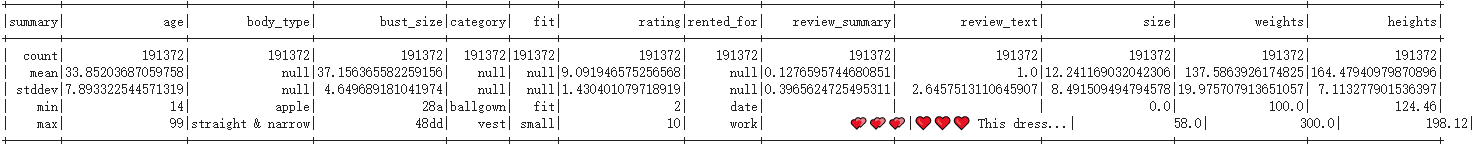
For sentiment analysis, we added a new column named “attitude” with logical values. We set rating that lower than or equal 5 to logical value 0, rating that greater than 5 to logical value 1.

For products with high rating analysis, we created a new column as the target prediction. We set 6 as threshold number that values in “rating” higher than 6 would be assigned as “high” while the others were “low”. And for customer rating preference analysis, we created a new column as the target prediction. We set 10 as threshold number to split ratings into 1 for “is 10” and 0 for “not 10”. For both of these tasks, we made a pipeline to map categorical variables into label indices and assembled these indices into vectors before applying in each model.

For recommendation analysis, we used a pipeline to perform PCA on data frame to decrease the dimension of features.

## **Data Exploration**

After the total preprocessing, there were 191372 records in the data frame. We did statistics analysis with the rest variables. From the summary, we can find that all of them contain null values. We can find that the minimum rating is 2 while the average rating is around 9, which means most customers prefer to give a good rating.



From the Figure 2.3(1), we can find that most types of customers are hourglass. This can be a clue to help the platform select their products. And from Figure 2.3(2), we can find that most customers prefer to give 10 ratings, which are high enough. According to Figure 2.3(3), we could find that most of customers thought the clothes fitted themselves, which means the size was suitable. This observation could be helpful in analysis.

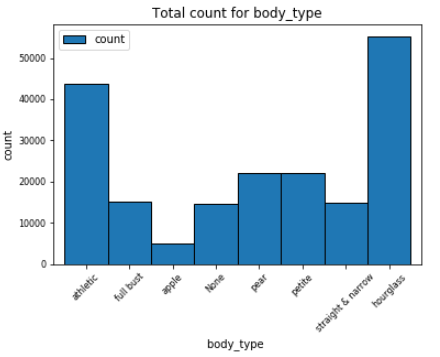
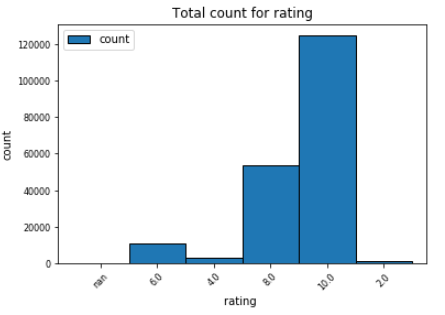
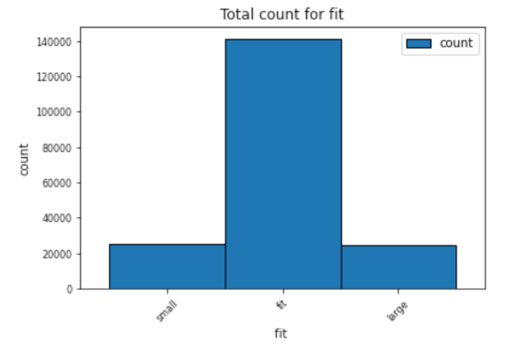
  

Figure 2.3(1) Total Count for Body\_type Figure 2.3(2) Total Count for Rating Figure 2.3(3) Total count for fit

From Figure 2.3(4), we found that most customers rented clothes for wedding. According to Figure 2.3(5), the count of category varied sharply. Most of rented clothes were dresses, which may result from the fact that most of customers rented clothes for wedding. From Figure 2.3(6), the large number of customers aged from 25 to 30 verified the thought before.

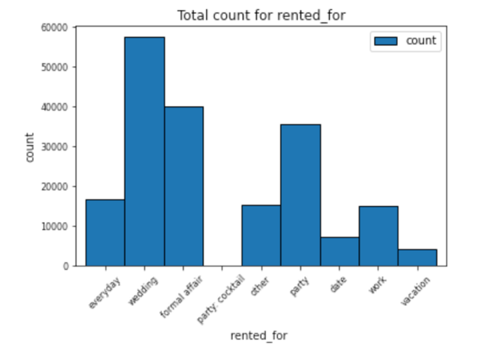
**

Figure 2.3(4) Total count for fit Figure 2.3(5) Total count for category Figure 2.3(6) Total count for Age

# **Methodology**

## **Sentiment Analysis**

## **Clustering Analysis**

## **Classification Analysis**

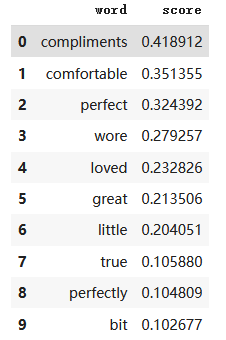
# **Models**

## **Sentiment Analysis**

We were focused on creating an ideal sentiment analysis model which was used to predict the sentiment of reviews in each transaction for the purposes of finding out important factors influencing customers attitudes. The model followed Logistic Regression. Having added the new column “Attitude”, we randomly split the dataset into 3 parts: training (60%), validation (30%) and testing (10%) for this part’s analysis. After a series of text processing on “review\_text” column, we finally got the output column “tf-idf”. Then we set the feature column “tf-idf” and the label column “Attitude” to the two logistic regression models.

For the grid search, which was used to find the regParam and the elasticNetParam of the best model, we ran a loop with a 3 \* 3 grid to orderly fit the training dataset. In the latest code execution, the best regParam was 0.01 and the best elasticNetParam was 0.2. With these two parameters, we successfully got the best model, with which the 10 most positive factors (words) and the 10 most negative factors (words) were figured out. (As two graphs are shown below)

The 10 Most Negative Factors (Words)

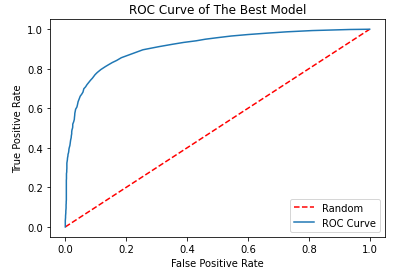
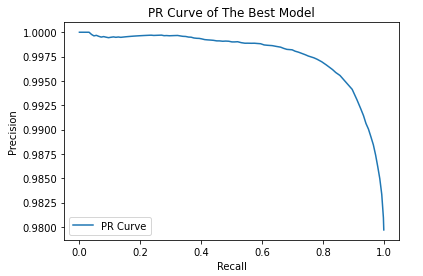


The 10 Most Positive Factors (Words)



Then the Binary Classification Evaluator was set to measure the ROC Area Under Curve (AUC) of two logistic regression models. The AUC of the original model was 0.787123, and that of the best model was 0.909823. The accuracy had improved a lot.

To further evaluate the model, the ROC Curve as well as the PR Curve were plotted. (shown below)

For the ROC curve, the closer the curve is to the upper left corner, the more positive cases take precedence over the negative ones, and the better the overall performance of the model will be. For the PR curve, we find that range of precision rate is very high. One of the possible reasons may be that words meanings in review\_text from customers have strong correlation with their attitudes. Which means most customers do not use ironies in their comments.

## **Products with High Rating Analysis**

The goal we implemented this model was to predict the situation where customers are likely to give different ratings for the purpose of judging whether the customer would like to provide a high rating or not. The model using in task was Random Forest classification.

The independent variables were “age”, “body\_type”, “bust\_size”, “category”, “fit”, “rented\_for”, “weights” and “heights”. The target variable was created variable named “new\_rating” with threshold of 6. We dropped “review\_summary” and “review\_text” since they were irrelevant with this task. We created a pipeline with StringIndexer to transform to label indices and encapsulated these indices in the column “total\_features”.

We split the transformed dataset into three parts including training (60%), test(30%) and validation(10%). Since the target column had three levels, we used f1 in “MulticlassClassificationEvaluator”, which is weighted measure of Precision and Recall. We tuned parameters in numTrees and maxDepth to improve performance because they were the most important. Validation was the testset in tuning parameter and the goal is to find parameter set that contributes to the highest f1 value. The maximum f1 score was 0.8877076632523566. Then, we used parameter set to build random forest model and transformed in test dataset. The f1 score here is 0.8851122573345579.

We extracted all features and sorted them by feature importance in descending order. As you can see from the chart(Figure 4.2(1)), “fit”, “rented\_for” and “age” are top 3 most significant features in this task. We didn’t find an efficient way to visualize the result of random forest model and therefore we chose one tree example(Figure 4.2(2)). 0.0,1.0 and 2.0 in prediction are separately “Low”, “High” and “Medium”. We chose it because it has all three different predictions. Based on the picture, most of prediction are high. When “fit” level is fit, “age” is higher than 29.5, “height” is higher than 189.23 and “rented\_for” is wedding or formal affair, the rating will be “Low”. When “weight” is lower than 111.5 lbs, “rented\_for” is date or wedding, “height” is higher than 171.45 and “fit” is small, the rating is predicted as “Medium”. From two observations, we thought “rented\_for”, “fit”, “weight” and “height” are crucial in predicting high rating.

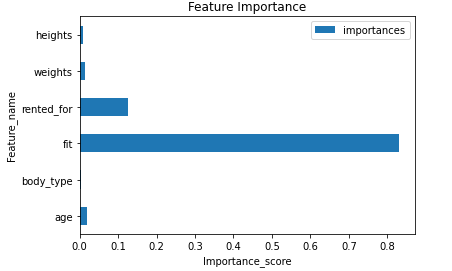
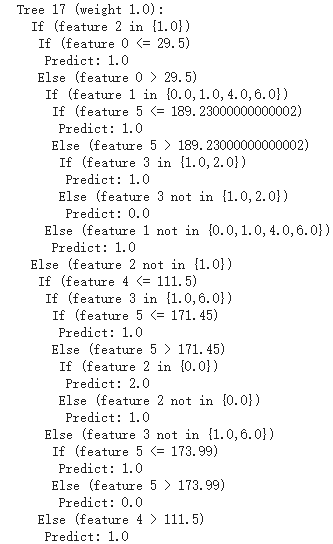
 

Figure 4.2(1) Feature Importance Figure 4.2(2) Tree Example

## **Recommendation Analysis**

The goal we implemented this model was to Predict what kinds of product will fit different customers for the purpose of recommending appropriate product to customer. The model using in this task was K-Means Clustering.

Since the goal in this task was to make recommendation for different customers, we decided to use transactions whose ratings were equal to 10. We combined “body\_type”, “bust\_size” and “category” as one column in which it should be presented as “body\_type-bust\_size-category”. The reason why we did it like this way because we wanted them to be represented as customer(body\_type and bust\_size) and product(category). We used a pipeline to perform Principal Component Analysis(PCA) on dataframe which includes “body\_type-bust\_size-category” and “review\_summary”.

Since it was a clustering problem, we used “ClusteringEvaluator” and evaluator metric is silhouette score. Based on Figure 4.3(1), we set K from 2 to 3 and found the highest silhouette score when k was 3. We then built a cluster result plot based on K-means model. From Figure 4.3(2), the cluster result was not bad, and some points were a little bit far away from their clusters.

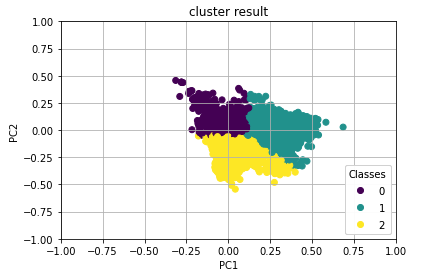
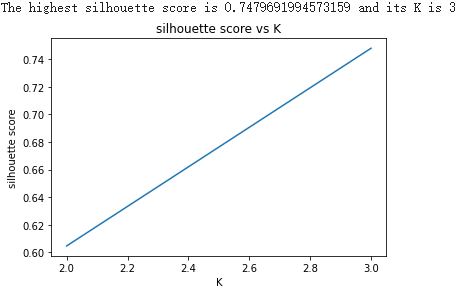


Figure 4.3(1) Silhouette Score Plot Figure 4.3(2) Cluster Result Plot

Since there were three different clusters, we thought it may result in three different customer results for recommendations. Therefore, we filtered “body\_type-bust\_size-category” based on three clusters and grouped by them and sorted in a descending order.

We then extracted top 3 observations in each cluster and made three plots from Figure 4.3(4) to Figure 4.3(6). In first cluster, it could conclude that when “body\_type” is athletic or hourglass and “bust\_size” is from 34b to 34d, we will recommend dress. In second cluster, when “body\_type” is hourglass and “bust\_size” is 44f, dress or grown will be a good choice. In third cluster, when “body\_type” is hourglass and “bust\_size” is 34b or 36c-d, dress is a good choice. In short, popular “body\_type” is hourglass and athletic and “bust\_size” is mainly from 34 to 36 or 44. At this time, people match such criteria will be recommended to buy dress or grown.

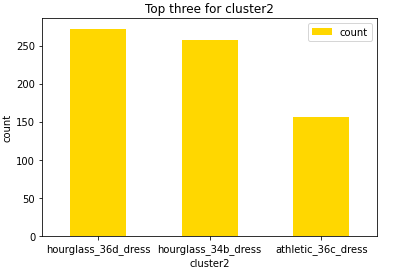
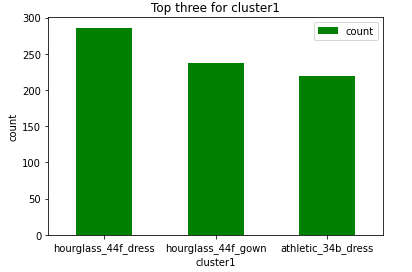
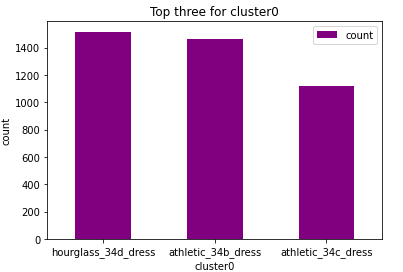


Figure 4.3(3) Top 3 for cluster 0 Figure 4.3(4) Top 3 for cluster 1 Figure 4.3(5) Top 3 for cluster 2

## **Customer Rating Preference Analysis**

For this task, we built three models to select important features that could influence the customer rating preference by comparing the score of each model. Models were used to predict whether a customer would like to give a rating of 10, which we set as 1. We randomly split the transformed data into two parts, 80% of which was training data while 20% was testing data. We used only five related variables that we inferred as important in analysis to build models, which are “body\_type\_Vector”, “category\_Vector”, “rented\_for\_Vector”, “size” and “age”.

### **4.4.1 Decision Tree**

As mentioned in data preprocessing, we created a pipeline to transform categorical variables into numerical values. We used transformed data frame to build decision tree classification model. We merged these five variables into a vector and scaled to unit variance, then worked as the input of the training model. Then, we used five-fold cross validation with grid search to select the best max depth, which is one of the most important parameters. We tuned “maxDepth” from 4 to 6, and found the best parameter is 6. To evaluate the model, we used ‘AUC’ because this was a classification problem. The AUC of test data of decision tree was 0.5013547208796848.

### **4.4.2 Logistic Regression**

Apart from vector assembler and standard scaling, we used Chi-Squared feature selection to select features whose p-value were below 0.05. Then, to avoid overfitting, we used three-fold validation with grid search to tune regularization parameter from [0.005,0.01] and the ElasticNet mixing parameter from [0.01, 0.05]. The best combination was regParam with 0.005 and elasticNetParam with 0.01. The AUC of test data of logistic regression was 0.5011663514671425.

### **4.4.3 Gradient-boosted Tree**

Same with decision tree, we used vector assembler to assemble inputting variables into a vector and scaled this vector. Then, we called GBTClassifier to build the model. We used five-fold cross validation with grid search to tune the max depth of the tree from 4 to 6. Then, we found the best parameter was 6 and the AUC was 0.5018142778414154.

### **4.4.4 Conclusion**

To compare the three models, we drew ROC curve and Precision/Recall curve to see the performance. We found three curves were almost coincided in Figure 4.4(1) while there was small difference in Figure 4.4(2). According to Jason, we should use Precision-Recall curves because changing category variables into numeric ones made it become a large class imbalance. (Jason Brownlee, 2018) [2] Therefore, we found Gradient-boosted Tree performed better.

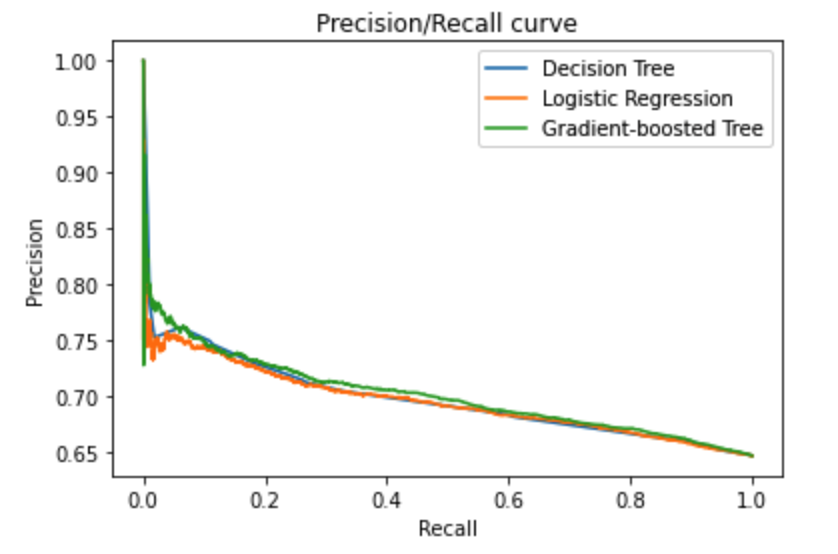
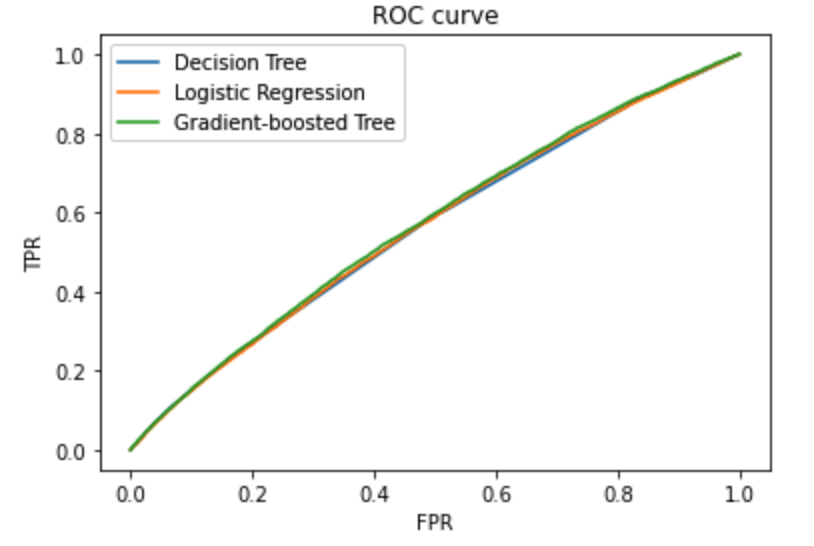


Figure 4.4(1) ROC curve Figure 4.4(2) Precision/Recall curve

Besides, by comparing the AUC of the models, we could support our choice.

**Model Comparison with AUC**

|  |  |
| --- | --- |
| **Model** | **AUC** |
| Gradient-boosted Tree | 0.501814 |
| Decision Tree | 0.501355 |
| Logistic Regression | 0.501166 |

Table 4.4(1) Models with AUC

Therefore, we used the result of Gradient-boost tree to select important features. We selected 10 most important features according to the weights.

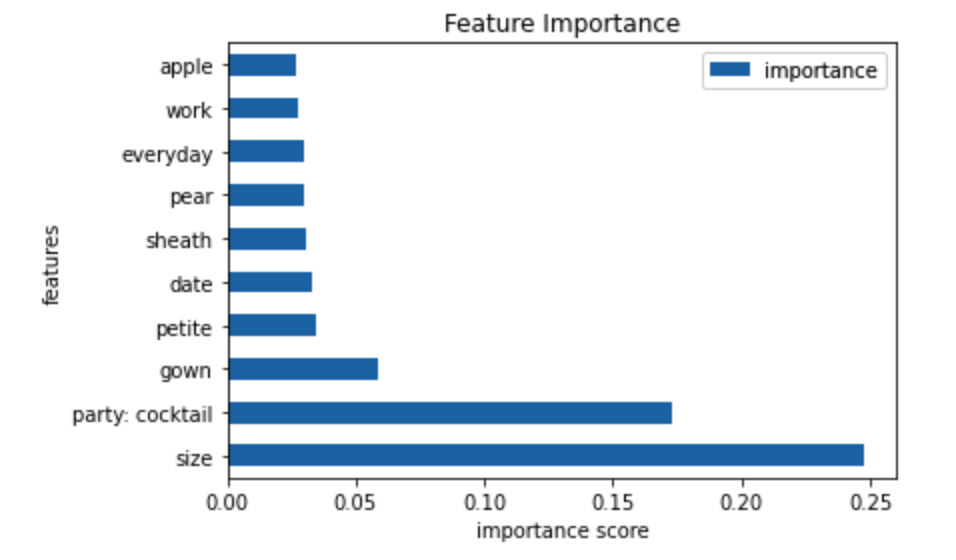


Figure 4.4(3) Ten Most Important Features

We could infer that size was the factor that influenced customer giving high rating. Besides, in the purpose of renting, “cocktail party”, “date”, “everyday” and “work” were most important factors influencing a high rating. For category, we found ‘sheath’ and ‘gown’ would be kinds of products that people preferred giving high ratings. For petite and apple-shaped customers, they would like to give high rating.

# **Conclusion**

*Summary Table for 4 predictions*

|  |  |
| --- | --- |
| Prediction | Result |
| 1. Sentiment Analysis | * Successfully find the 10 most negative factors and 10 most positive factors in the review\_text column. * Get a sentiment analysis model with about 90% accuracy. * Discover that words meanings in review\_text from customers have strong correlation with their attitudes. * Most customers do not use ironies in their comments. |
| 2. Products with High Rating Analysis | “rented\_for”, “fit”, “weight” and “height” are crucial in predicting high rating. |
| 3. Recommendation Analysis | Popular “body\_type” is hourglass and athletic and “bust\_size” is mainly from 34 to 36 or 44. At this time, people match such criteria will be recommended to buy dress or grown. |
| 4. Customer Rating Preference Analysis | * Size was the factor that influenced customer giving high rating. * In the purpose of renting, “cocktail party”, “date”, “everyday” and “work” were most important factors influencing a high rating. * For category, we found ‘sheath’ and ‘gown’ would be kinds of products that people preferred giving high ratings. * For petite and apple-shaped customers, they would like to give high rating. |

*Summary Table for Model Comparison Results (AUC) for 4 Predictions*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GBT | Decision Tree / Random Forest | Logistic Regression | KMeans Clustering |
| Prediction 1 | NaN | NaN | Original: 0.787123 | NaN |
| Best: 0.909823 | NaN |
| Prediction 2 | NaN | Using Rating Number as predictor: 0.65 | NaN | NaN |
| Using rating level as predictor: 0.92 |
| Prediction 3 | NaN | NaN | NaN | 0.748 (highest when k = 3) |
| Prediction 4 | 0.501814 | 0.501355 | 0.501166 | NaN |

*Summary Table for 4 Inferences Results*

|  |  |
| --- | --- |
| Inference Number | Ranked Most Significant Feature(s)(high to low) |
| 1 | Words in review\_text |
| 2 | “fit”, “rent\_for” |
| 3 | “body\_type”, “bust\_size”, “category” |
| 4 | “size”, “rent\_for”, “category”, “body type” |

# **Appendix**

## **References**

1. Misra, R., Wan, M., & McAuley, J. (2018, September). Decomposing fit semantics for product size recommendation in metric spaces. In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 422-426).
2. Brownlee, J. (2020, August 21). How to Use ROC Curves and Precision-Recall Curves for Classification in Python. Retrieved November 28, 2020, from https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.