Women Clothing E-Commerce

Data set description:

This data set is about women clothing e-commerce customer reviews. It includes 23486 rows and 10 feature variables, where each feature represents a column. Each row corresponds to a customer review. The following are the columns and its description.

a. Clothing ID: Integer Categorical variable that refers to the specific piece being reviewed. In this data set, there are 1206 unique clothing ID's.

b. Age: Positive Integer variable of the reviewer's age. There are different age groups starting from 17 to 99.

c. Title/Review Title: String variable for the title of the review. It has a total of 13993 unique titles.

d. Review Text: String variable for the review body.

e. Rating: Positive Integer variable for the product score granted by the customer from 1 Worst to 5 Best.

f. Recommended IND: Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.

g. Positive Feedback Count: Positive Integer documenting the number of other customers who found this review positive.

h. Division Name: Categorical name of the product high-level division. There are 3 divisions which refer to clothing or apparel size.

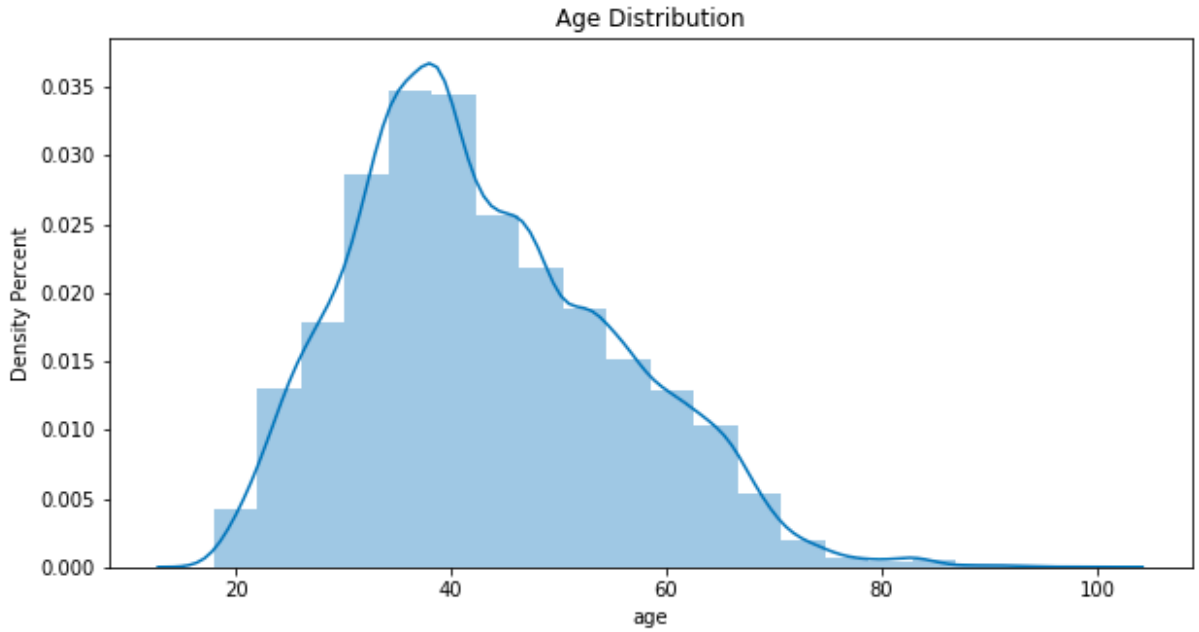
i. Department Name: Categorical name of the product department name. There are 6 departments which refer to apparel or clothing types.

Class Name: Categorical name of the product class name. This column is a subdivision of the department and corresponds to 20 different types of class categories.

Quality of Findings:

S**ummary for Finding 1**: Distribution of Age of people who reviewed the purchased products.

the validity of Finding 1:



Managerial insights: From the results, most reviewers on the products are in the age group from 31 to 45. The age distribution figure shows that this age range has the highest positive reviews. Therefore, the two main insights or points to be considered are:

a. The resulted age group is the most active in purchasing the products and reviewing them. So, the business must concentrate on keeping up with this fragment and

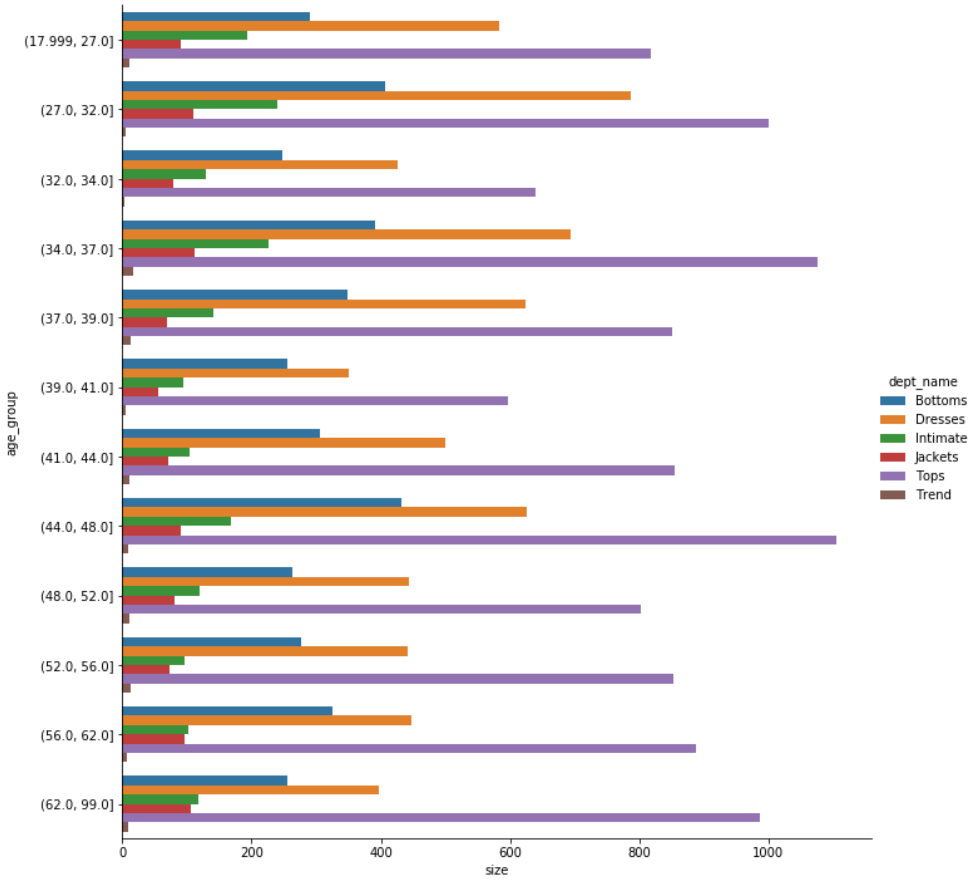
b. The e-commerce business element can investigate why other age groups are relatively less than the age group 31 to 45.

Learnings from this can be that the company is missing out on the very huge potential age group i.e. 20- 30. This group tends to be more vocal about their opinions the lack of reviews from this age group could mean that the company actually failed to make a mark in the category thus the necessary steps should be taken to increase sales to that section and enhance profits. The company can also review why they do not have reviews from the age groups 1-19 and can also target their business towards that customer segment

Summary for Finding 2: Frequency Distribution of Departments According to the Age Groups

Description:

the validity of Finding 2:



Managerial insights:

One of the reasons might be the product is offered in several apparel sizes, which attracts all age group people for a specific department. This finding can measure product reviews and sales based on the category department.

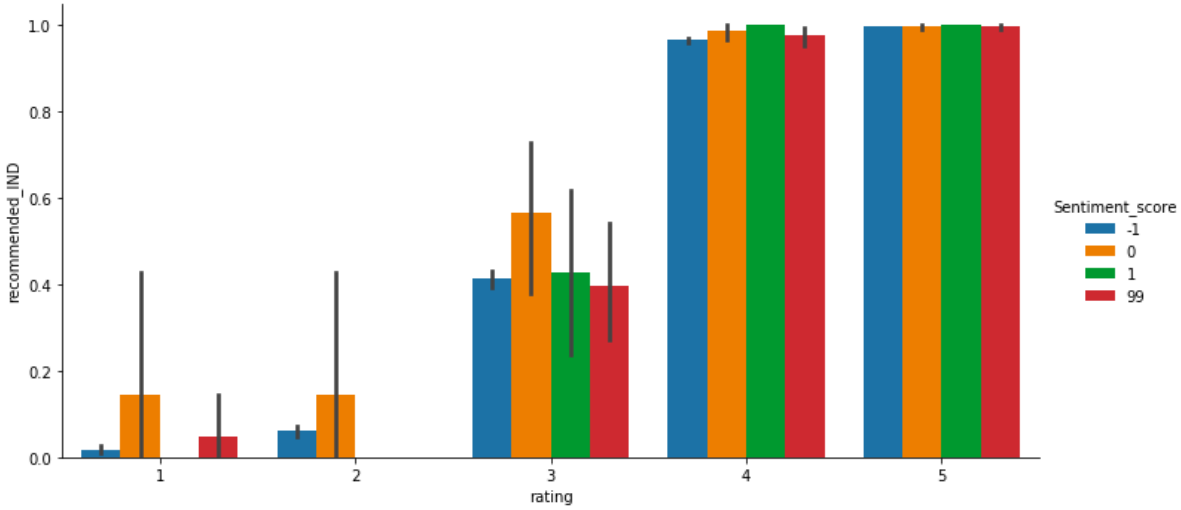
Summary for Finding 3: Distribution of Sentiment Scores Vs Rating Vs Recommended Indicator across Reviews.

Description:

The sentiment score for reviews are 1,0,1,99 represents Negative, Neutral, Positive and a missing review. These are the rows where the review text is not written, or the field is empty. In some scenarios, customers do rate the product and recommend the product but they might not write any review text. For such fields, we have considered the score as 99. From this graph, we can see that the sentiment score is almost similar

for all the ratings. The customers who gave rating 5, wrote a negative text about the product but still gave the recommendation. There might be reasons such as the customer had liked the product, but it might not be the way she expected it. The apparel types and sizes might be one of the reasons for negative review text.

validity of Finding 3:



Managerial insights:

From this e-Commerce we get a sense of what is not working and make necessary changes to improve the shopping experience of the customer. It also gives the company the insight as to how customers perceive their products and/or services and would give them an idea on how to improve their business.

S**ummary for Finding 4:** Distribution of Products Based on Recommendation Factor

Description:

The recommendations are important because it says about the number of times the product is recommended and the total times the clothing products are purchased. Based on this feature, the clothing_id which are more recommended, are the most purchases made. The total size of the recommended IND for each clothing ID is the number of times the product is purchased, and the sum of recommended IND is the Number_of_times the product is recommended.

validity of your finding:

Out[19]:

clothing_ID	times_purchased	Number_of_times_recommended
1078	1024	841
862	806	661
1094	756	621
1081	582	494
872	545	478
829	527	436
1110	480	403
868	430	325
895	404	341
936	358	294

Managerial insights:

The clothing_id which are more recommended, are the ones that are purchased more. This is one of the important findings from which a business can look into, to incur profits and to implement changes in their products.

Machine Learning on the data set:

We tried various machine learning models on the clothing data set including classification (multiple algorithms), clustering and but could not find anything interesting to do with this data set. We tried to predict if any clothing item will be recommended or not but the recommendation is a subjective concept with having any relationship with the demographics of customers available in the dataset. Thus, without sentiment and rating, it was very difficult to predict is any item would be recommended or not. The code that we wrote test different classification algorithms for cross-validate and predicting the recommendations is mentioned below:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier

clfs = [DecisionTreeClassifier(), sk.ensemble.RandomForestClassifier(n_jobs=-1),
sk.naive_bayes.GaussianNB(),
        sk.linear_model.LogisticRegression(n_jobs=-
1),sk.tree.DecisionTreeClassifier(),sk.ensemble.AdaBoostClassifier(),
        QuadraticDiscriminantAnalysis()]


maxAUC = -1
bestCL = ""
for cl in clfs:
auc = sk.model_selection.cross_val_score(cl, X, Y, cv=kf, n_jobs=-1, scoring=
'roc_auc').mean()
    print (str(cl) + ' ' + str(auc))
    if auc > maxAUC:
        bestCL = cl
        maxAUC = auc
print('**********************************************')
print ('Best is... ' + str(bestCL) + ' ' + str(maxAUC))


DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best') 0.5773391692500489
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators='warn', n_jobs=-1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False) 0.6574189625597611
GaussianNB(priors=None, var_smoothing=1e-09) 0.5783492050107766
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn', n_jobs=-1,
```

```
        penalty='l2', random_state=None, solver='warn', tol=0.0001,
        verbose=0, warm_start=False) 0.600083262946455
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
          max_features=None, max_leaf_nodes=None,
          min_impurity_decrease=0.0, min_impurity_split=None,
          min_samples_leaf=1, min_samples_split=2,
          min_weight_fraction_leaf=0.0, presort=False, random_state=None,
          splitter='best') 0.5792383974112773
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
        learning_rate=1.0, n_estimators=50, random_state=None) 0.7092416393396055
QuadraticDiscriminantAnalysis(priors=None, reg_param=0.0,
             store_covariance=False, store_covariances=None, tol=0.0001) 0.5369813459248307
**********************************************
Best is... AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
        learning_rate=1.0, n_estimators=50, random_state=None) 0.7092416393396055
```

Since the prediction model could not be used in the models the decision tree classifier was used to identify the behavior of the classes.