Zijing Wang
Yayi Guo
Jiawei Zhou

Predicting “fit or not” for customers
Using data from “RentTheRunway: clothing fit feedback”
CSE 158 Assignment 2

Zijing Wang
ziw253@ucsd.edu
A13808607

Jiawei Zhou
jiz108@ucsd.edu
A92119932

Yayi Guo
yag010@ucsd.edu
A13566976

Dataset

Description:

The dataset we use includes 190k data about the measurements of clothing fit and reviews from *RentTheRunway*. There are 15 data fields in total in our dataset, which are:
['age', 'body type', 'bust size', 'category', 'fit', 'height' ', 'item_id', 'rating', 'rented for', 'review_date', 'review_summary', 'review_text', 'size', 'user_id', 'weight'].

We choose ‘fit’ to be our major field. This field shows whether the items of clothing customers rented can fit them or not. In total, we have three values in this field: ‘fit’, ‘large’ and ‘small’. Our goal is to predict if a rented clothing will fit the customer given the customer’s physical features including age, body type, bust size, etc. and other fields like reviews.

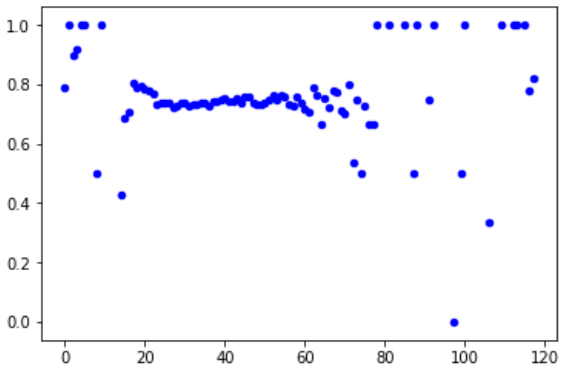
Before analyzing the data, we take the look at the overall data and notice that there are some extreme values. For example, in the ‘age’ field we find someone is 1 year old and someone is over 100 years old. These situations are obviously impossible to happen. Hence, when we take ‘age’ as a part of the feature, we will we limit the number range in the age field. to eliminate irregular data. Not only ‘age’ but ‘height’, ‘weight’, ‘body type’, ‘bust size’, ‘size’ fields have irregular values. We will deal with them before we build the feature vector and models.

Exploratory Analysis:

Since we have 3 entries in ‘fit’ field, we explore the whole data set and we get:
{'fit': 141995, 'large': 24691, 'small': 25776}.

Firstly, obviously ‘height’, ‘weight’, ‘body type’, ‘bust size’ and ‘size’ fields are related to ‘fit’ field. We will take them all as the feature. From exploring the training data, we found that items have a higher rating (rating score of ‘8’ and ‘10’) tend to have more user rated as ‘fit’ and for those that have a lower rating (scores of ‘2’, ‘4’, and ‘6’) have more ‘unfit’ labels than ‘fit’ based on the following data:
**{'10': Counter({False: 23784, True: 100753}),
'2': Counter({False: 755, True: 291}),
'4': Counter({False: 1793, True: 998}),
'6': Counter({False: 5711, True: 4986}),
'8': Counter({False: 18424, True: 34967})}).**
Hence, we will take rating as one part of our feature.

Also, we explore the relation between the probability of ‘fit’ with respect to age in the following figure. As we can see, this scatter plot is random, so we think age may not be the determinant for the predictions of ‘fit’.



We may reduce the weight of the ‘age’ field in modeling. However, when we see data in the range [20,60], the relation tends to be a constant. Hence, we still take ‘age’ as one part of our feature.

Predictive Task:

Pre-Process Data:

After exploring the data, we found that we could not directly use the raw data in our prediction models because there are some data that have fields containing unreasonable values, as well as some data absent some fields that we might potentially use as features in the prediction, like 'age' Therefore, we have to correct and filter the data to make it suitable for training.

Firstly, we compute a default value for each field including 'age', 'height', 'weight', 'body type', 'bust size', 'size' by taking the most common value of the respective field of the dataset. We choose the most common values because the most common values are the most representative ones. The user is more likely to have a physical feature of the most representative value.

For example, we want to limit the range of user age in the 'age' field. We set the range to be from 14 to 80. The data with 'age' field value outside our range will be given the most common age of the dataset. By doing so, we can potentially minimize the side effects of these data to our model.

After dealing with the default value and strange age problem, we decide that our task is to predict if an item of a certain size will fit the user, given the user's physical features, the size, and the item. By tuning different models, we will consider more in our feature vector. The reason we choose this task is: first, the 'fit' or not is clearly in the dataset, so we can build up a test set to evaluate our model easily. Second, we did some similar tasks before in the assignment and homework, so we have more experience and confidence to build the predictors.

The features in raw data are not enough, so we want to generate some new object field which is hidden in the raw data. The Body mass index (BMI) is a reasonable value to show the person's body situation(BMI =(weight in pounds * 703)/height in inches^2). Generally, this body situation(fat or skinny) will directly influence the fitting of a cloth. Hence we think this value is worth to be considered compared with 'weight'. Although we calculate the BMI, we still have to keep 'height' because height determines the length of a cloth.

Further, to improve the accuracy, we add 'category' and 'Seasons of Year' because in different seasons people may rent specific types of clothing, and some types of clothing like coats, hoodies, and more may be easier to fit most people.

Baselines:

One of our lowest baselines would be a simple naive model that predicts true (item will fit) if this size is the most common one in the previous deals of the user. We are using the most common size of the user for the predictor because this size might be the standard size for the user and it is likely that an item of this size will be fit. We get an accuracy of around 0.50. This is pretty low so we decide to use simple feature SVM. The feature is ['review_text']. We take the most common words and train the model. In this way, our accuracy is 0.67.

Evaluation:

After reviewing the dataset, the potential features we would use for our model are *['age', 'body type', 'bust size', 'height', 'rating', 'size', 'BMI', 'Seasons of Year', 'category']*. We found those features useful because the physical features are related to the size of item the user fits, and the average rating of the item also describes the global feedback of if the item fits user in general. Before we starting to build up our model, we anticipate that if a person has a larger BMI than the average, he has a lower chance to find a "fit" cloth.

There are many different approaches to solve this "fitting problem". We want to use Support Vector Machines and logistic regression.

Our first approach is Support Vector Machines, which classifies the dataset by determining a decision boundary between two classes that make them as far as possible from

each other. SVMs is an appropriate model and useful because customers who report 'fit' and who report not fit may turn out to be quite similar and hence hard to classify. However, some customer may get an unfit cloth, even the size is suitable, since their body shape is strange. Based on this situation, we cannot care some unimportant aspects all the time; we may not focus on the data points that are way too hard to classify but on those easier classify samples which really help us to better analyze and interpret data.

To evaluate the performance, we would be calculating the prediction accuracy of the model. However, it is not enough, because in the data set the number of 'fit' is much larger than the number of 'not fit'. Hence, if a predictor always outputs 'not fit' then it can get a higher accuracy. Since this, we additionally calculate the BER value which can give us a better measurement of our model because it measures the model depending on the instances of each class. If we get a high BER then we will consider changing our model.

For **validation** of the model, we are going to build up three set: train set, validation set, and test set. The train set is used to train the models and we will adjust the models base on the result of our validation set.

What's more, we try logistic regression, which is applied in our assignment 1. Since Assignment 1 deals with a similar binary problem with our current predictive task, we think this might be an appropriate model. In addition, the probability is useful in real life situation, because 'fit' and not fit are not as simple as a yes or no question. If we know the probability, we can do more practical work in real life. To evaluate its performance, we also calculate the accuracy and BER as we discussed above.

Model:

SVM:

Approach 1 (Simple item feature):

Our first model for the predictor is by using the average ratings of the items the user trying to purchase. In the beginning, we are experimenting with possible features that we want to use and see their performance to evaluate for a better model. The first feature is only the size the user is trying to buy and the average rating of the item. We got an accuracy of around 0.68 but the BER is around 0.50. Even though we have a decent accuracy, the balanced error rate is really high.

Strength: short run time and easy to write

Weaknees: This predictor is really simple and lack of other relevant features that might be useful or correlated to the prediction fitness.

Approach 2 (User feature with one hot):

In our next model, we added in other features including age, weight, size, as well as body type, bust size, and height by one hot encoding. We are using one hot encoding instead of direct values because we found the distribution of the value potentially affects the result more. For example, people that are taller of their weight might found a clothing unfit because the clothing might not be long enough for them. From this predictor, we got an accuracy of 0.71 and a balanced error rate of 0.47. This predictor gives us a pretty decent result by classifying fit with relevant features.

Strength: This model includes relevant user features and improves performance.

Weakness: Features from the item might also be useful that we haven't included.

Approach 3 (Combining one hot user feature with the average rating of the item):

To improve the performance of our SVM model, we consider adding in more features about the item that is potentially useful. Therefore, in this model, we combine the feature of the user with the feature of the item the user is trying to rent. We added the average rating of the item with the one hot encoding of user features. This model does not improve the accuracy or the BER a lot, but it is considered a reasonable modification by adding in a relevant feature.

Strength: includes relevant features from both user and item, got higher accuracy.
Weakness: Some of the features might be redundant and have a negative effect on the model.

Approach 4 (Drop 'weight' and add 'BMI', 'Season of Year', 'Category' to features):

Since we have already used all the features that existed in the data, it is hard to find out what we can add to the predictor to make it perform better. Obviously, we cannot directly convert the data to a simple feature. Instead, we think about how to adjust the data to gain more important information. We get the BMI idea from the gym. In the gym, the personal trainer always wants the student to test the BMI before the train. Using BMI instead makes more sense because a person might be heavy due to his height. The BMI can give us a more direct evaluation of body size than weight. We are not simply adding the value of BMI. Instead, we make the BMI in 0-18 as 0, BMI in 18-25 as 1, etc. Moreover, by 'review_data' we can know the season of the year. People wear more clothes in the winter so when they rent the cloth, it is more possible to report 'small' because they wear some clothes under renting cloth. We decide to use one hot to encode the season of the year, for example, 0 for spring 1 for summer, etc. After we add these features to the vector, we get a more accurate predictor. Now the accuracy is 0.74 and BER is 0.45. The result shows that extract more information from the data is critical.
Strength: this model uses advanced, processed features and dropped redundant features.
Weakness: If we add too many features, the time consuming will increase.

Approach 5 (Tuning the regularization parameter C):

After experimenting the SVMs with different features, we try to further improve the performance of our predictor by optimizing it with different regularization parameter C, which is used to determine how much we want to avoid misclassifying each training sample. We tried C = {0.01, 0.1, 1, 10, 100} and we found our predictor has the best accuracy with C = 0.1. We notice that when C = 0.1, it performs slightly better than C = 0.01 but when we have C higher than 1, the accuracy will be significantly lower on both the validation sets and the test sets. This is maybe when c becoming large, it will overfit the train data. With C = 0.1, we get an accuracy of 0.75 and BER of 0.43.
Strength: When c = 0.1, we get a higher accuracy.
Weakness: When c = 0.1, we have more misclassifications since c is too small.

Logistic Regression:

This method utilizes the sigmoid function and calculates the probability of each class predict whichever has a higher probability. The advantage of logistic regression is obvious: it does not assume the independence among the features. To better use Logistic Regression, we replace 'rating' with '***item_average_rating'***, which is more precise to reflect the fitting situation of items of different ratings.
Strength: The output of a logistic regression is more informative than other classification algorithms. It also reduces the misclassify.

Literature

This is a dataset from the website https://cseweb.ucsd.edu/~jmcauley/datasets.html , which is provided by the professor. It contains measurements of clothing fit from the website *RentTheRunway*. It serves to improve customers' shopping experiences and to reduce products return rates by doing the "fitting" prediction which is the product size recommendation. Since we firstly predict if the clothing fit or not without considering 'large' and 'small', this dataset is basically a binary classification/prediction task, which is essential in the field of machine learning. This kind of dataset contains 1/0 labels and some objects features. We have already studied this kind of

data set before, for example, in hw3 and assignment 1 we get in touch with the Amazon products data which contain users and what item this user buys. We did a similar predictive task, like to predict whether a user will purchase a specific item. In that assignment 1, our group members used Jaccard similarity to improve the accuracy of the predictor. Depending on situations, we choose models to use on our "clothing" dataset. It is worthy to mention that the binary classifiers are not limited to Support Vector Machines and logistic regression, which are used in this dataset. We can build up our own models take advantage of many technics, such as clustering the samples, similarity measures, or even use Neural Networks.

For state-of-the-art methods, since our predictive task is binary, advanced methods are not guaranteed. Actually, we have some complicated models that can be applied like large neural networks and Adaptive Boost, depending on the complexity of the data. Large networks can have numerous layers and weights or parameters so that it can capture complex and latent relationships among features. However, it is not a good choice for our dataset and predictive because our dataset and task are simple and can have other more direct and simple solutions. Adaptive Boost is an advanced and balanced classifier which creates a strong-learner by adding various weak-learners. It adjusts weights based on previous misclassified examples and thus usually gives high accuracy. Further, we can also add kernel methods. There are many state-of-the-art kernel methods such as the RBF kernel. For example, in homework 1, some of us chose to use RBF instead of Linear to shorten the runtime.

Modern methods are countless because similar dataset and binary predictive problem are common in machine learning. Complex and advanced methods are not necessarily suitable for some simple dataset while simple models work well.

Results and Conclusions:

Model	BER	Accuracy
SVM	0.43	0.75
Logistic Regression	0.39	0.78

Feature Representation:

Finally, we get our feature vector after we have tried several approaches in SVM and Logistic Regression. Our final feature vector is ***['age', 'body type', 'bust size', 'height', 'item_average_rating', 'size', 'BMI', 'Seasons of Year', 'category']***. We added some advanced features such as 'BMI', 'Seasons of Year', which definitely improve our predictor to be more accurate. Because directly adding 'review_date' makes no sense in predicting 'fit' while seasons can help us lock on some specific categories of clothing. For example, in winter, we focus on coats which are easy to fit people in different physical shapes. Also, we already have 'height' to decide the length of clothing and we need one measurement to help us decide the width. Compared with 'weight', 'BMI' performs better than 'weight' in predicting the width since some people are weigher because they have more muscle. Hence, our final feature vector works better than others we have used.

Interpretations of Parameters:

To better understand our predictor, here are some interpretations for parameters in SVM and Logistic Regression.

The effectiveness of SVM depends on the selection of the kernel, the kernel's parameters, and soft margin parameter C. We use the linear kernel and take 0.1 as the final value of C. C is a regularization parameter that controls the tradeoff between the achieving a low training error and a low testing error that is the ability to generalize our classifier to unseen data.

In Logistic Regression, we have many predictor variables since we always use one-hot to encode each part of feature ***['age', 'body type', 'bust size', 'height', 'item_average_rating',***

'size', 'BMI', 'Seasons of Year', 'category']. So we will too many variables to run efficiently. Also, we use the sigmoid function as the professor discussed in class.

Reasons for Success:

At first, as we did in hw3, we thought of SVM to train data. Thanks to it, we managed to develop our feature and examine our idea of baselines. When changing feature, we gradually figured out which feature is important and which is not. Then we reorganized the feature vector to get a better one. In our final feature vector, we can see that BMI performs better than weight we used in the beginning. Using SVM is not the best choice since it will focus more on the data points that are hard to predict. And our value of C is so small that it will lead some misclassifications. To improve our model, we began to use Logistic Regression, which improves the accuracy and BER.

Hence, our model performs well because of our advanced features and Logistic Regression. We think our result can be applied in some actual use:

1. Companies can develop an APP to predict the clothing size for customers so that customers can know which size they should choose without trying the clothing or measuring.
2. Some clothing shopping websites can directly recommend a proper size for their customers, offering them a quick and convenient shopping experience.

Reference:

[1]
"Clothing data"
https://cseweb.ucsd.edu/~jmcauley/datasets.html