PROJECT REPORT

CLOTHES SIZE RECOMMENDATION SYSTEM

01. Introduction

Many years ago, if customers wanted to buy clothes, they must go to the physical stores to try the clothes on to determine if sizes or the are for them or if they like the materials. However, it is not easy for a portion of customers whose houses are not located near the physical stores to shop in-store. Therefore, a lot of stores have developed online stores or websites not only to let customers search for the available products but also to let them shop online. Online stores have many advantages and disadvantages. One of the advantages include flexibility, which means customers can shop at anytime and anywhere and with some simple clicks, the products are delivered to them. Moreover, all available products and prices can be sorted and filtered quickly as they navigate through the online stores. However, we should consider not only the advantages, but also the disadvantages. For example, online stores are not used by those who cannot connect to the Internet or for some elderly who do not know how to use smartphones or the apps. The disadvantages are not only from customers but also from the functions of the websites. For instance, the websites that are not well-designed or well-structured makes it hard for customers to navigate through the sections on it. Many customers shop online and then return the items because of the wrong size selections based on the size charts, and that makes customers have bad experiences with online stores, which decreases the chance that they are going to shop online later in the future. To overcome that, we have designed a new function for the websites, which is basically a system to help customers select the best sizes to buy.

02. Data Preparation

There are a total of about 120,000 records. The columns represent the variables, which are weight, height, age, and size respectively, in which sizes are selected using weights, heights, and ages. There are six available sizes, which are XXS, S, M, L, XL, XXL, XXXL. Because the names and the order of the sizes are not in the standard format, we reformated them to XS, S, M, L, XL, 2XL, 3XL. We then checked for nulls in each column. Column weight and size include no nulls, but column age and height do as shown below.

weight	0
age	257
height	330
size	0

We removed the nulls instead of replacing them with the measures of center because there are only a few of them with respect to the total number of records. We then checked for the count of each size as shown below.

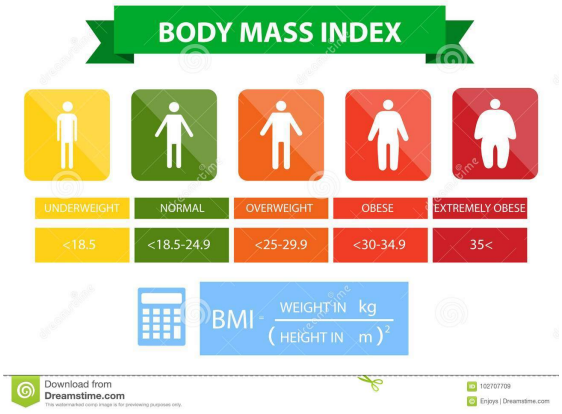
M	29575
S	21829
3XL	21259
XL	19033
L	17481
XS	9907
2XL	69

The number of records are not distributed equally among the sizes, but they are densely populated in size M and sparsely populated in size 2XL. In reality, if those numbers of records are the numbers of sales for each size, we should consider removing the size 2XL. Because of the small number of records for size 2XL, we treated 3XL as 2XL to increase the number of records for 2XL.

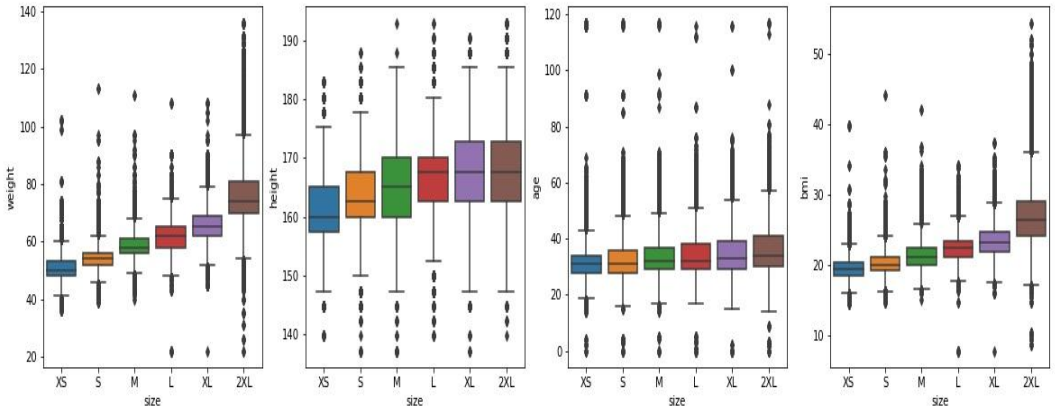
M	29575
S	21829
2XL	21328
XL	19033
L	17481
XS	9907

We then resample to increase the number of records for each size to 50,000. We have tried to (1) upsample size XS and downsample other sizes to 15,000, (2) downsample size M and upsample size S to 20,000, (3) downsample everything to 5,000, and (4) upsample everything to 50,000. We selected the last one because it yielded higher performance than the other.

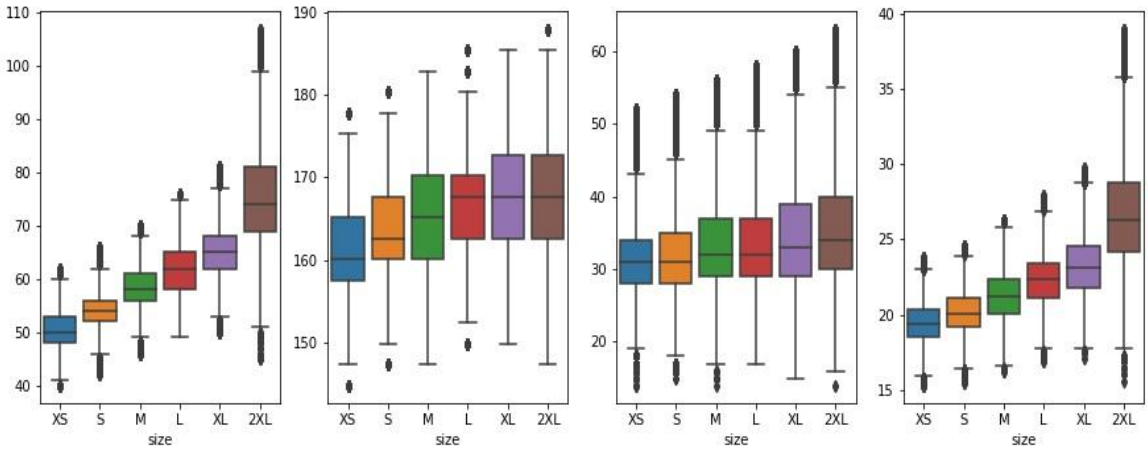
Because we have a few variables, we developed a new feature called BMI based on the weight feature and the height feature because it helps increase the performance of the machine learning models.



We checked for outliers regarding weight, height, and age for each size using boxplots. They must be removed because they did not make any sense, but created noises that make machine learning models perform badly. For example, a person whose weight is 110kg but wears size S; or a person whose height is 185cm but wears size XS. We removed them by converting everything into z-scores and removed those greater than 3 or -3.



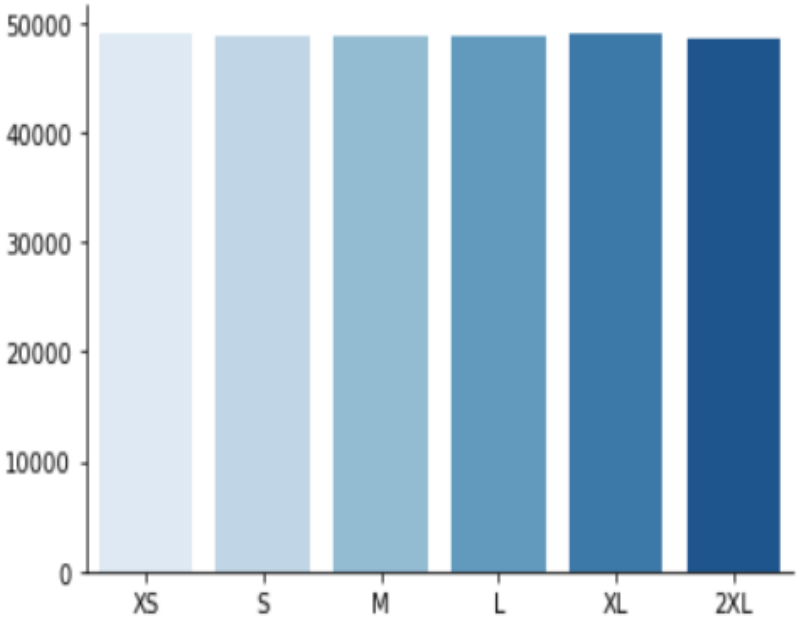
However, because the distribution of each size by each variable is not normal but skewed as mean does not equal median, converting everything into z-scores does not help us remove everything completely. As shown below, we failed to remove those outside the minimum and maximum values. However, we did not continue to clean them because they did not stay extremely far away from, but extremely near the minimum and the maximum values; therefore, they can be treated as small variances. The reason why we did not first do normalization was because we did not know how to convert the results that were in the format of z-scores back to weight in kilograms, height in centimeters and age in years.



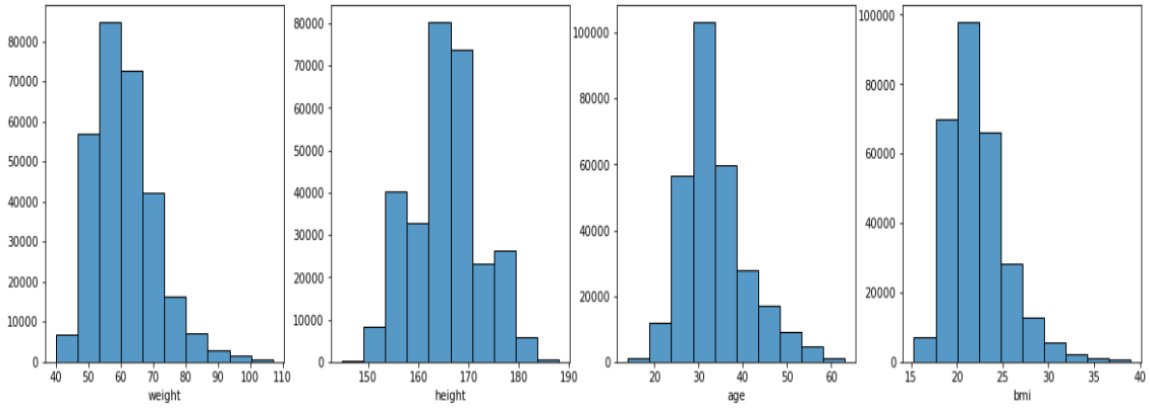
Below are the statistics before and after data preparation. (1) The number of records increased more than two times because of resampling. (2) The minimum increased and the maximum values decreased significantly. (3) However, it was reversed for the means, standard deviations, and percentiles.

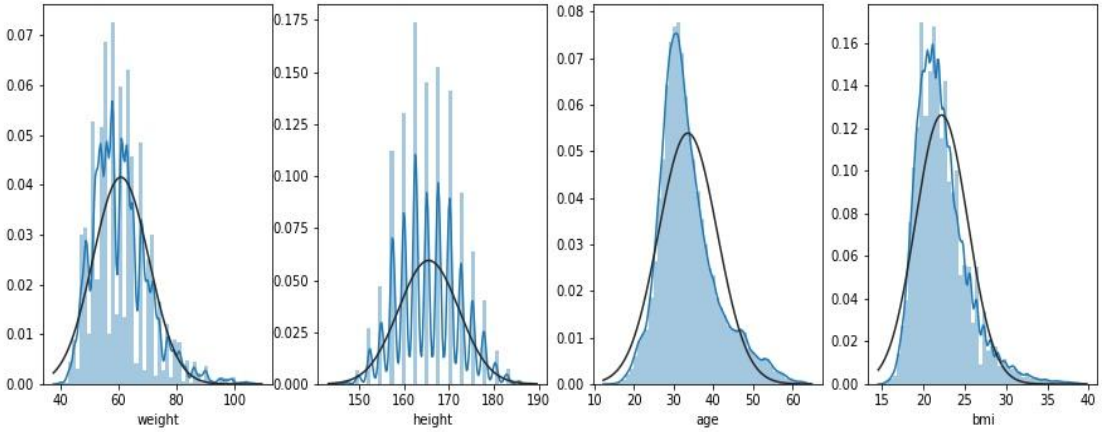
	weight	age	height
count	119153.000000	119153.000000	119153.000000
mean	61.756095	34.032714	165.807068
std	9.942877	8.148302	6.737797
min	22.000000	0.000000	137.160000
25%	55.000000	29.000000	160.020000
50%	61.000000	32.000000	165.100000
75%	67.000000	37.000000	170.180000
max	136.000000	117.000000	193.040000

	weight	age	height	bmi
count	291955.000000	291955.000000	291955.00000	291955.000000
mean	60.881030	33.559994	165.52207	22.204485
std	9.611322	7.404393	6.71491	3.161600
min	40.000000	14.000000	144.78000	15.300765
25%	54.000000	29.000000	160.02000	19.994665
50%	60.000000	32.000000	165.10000	21.705737
75%	65.000000	37.000000	170.18000	23.822169
max	107.000000	63.000000	187.96000	38.977129

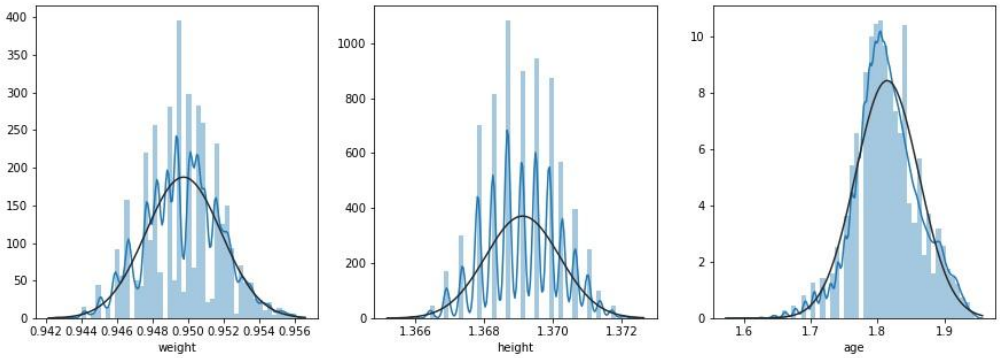


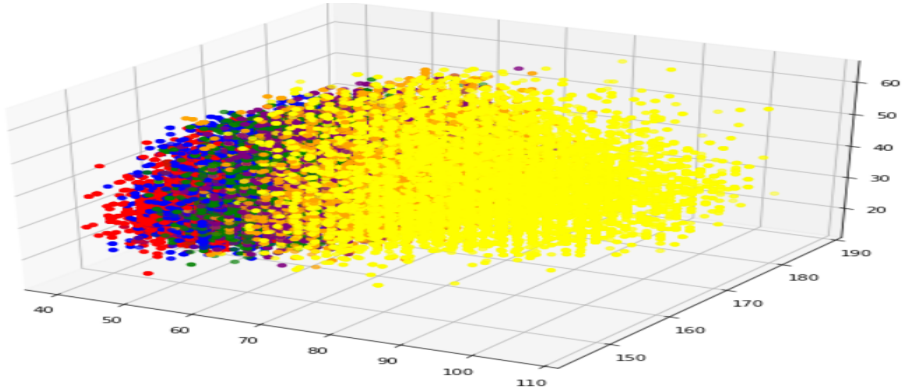
03. Data Visualization



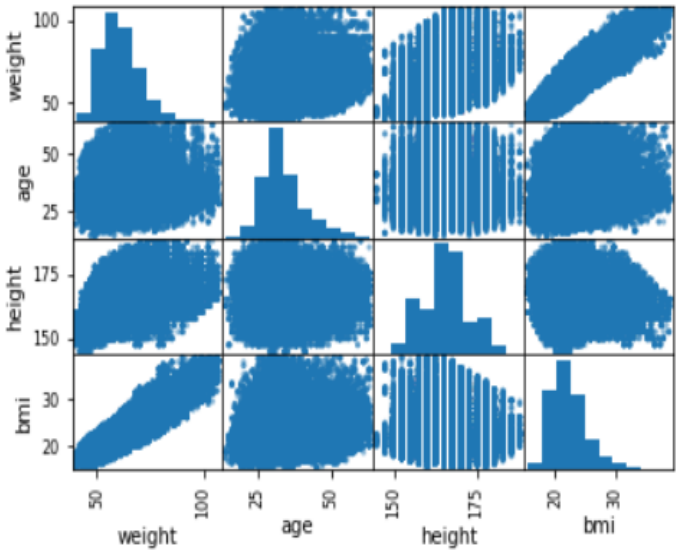


The histograms and the probability curves above are right-skewed. They can be normalized using some transformations such as box cox transformations as shown below; however, we did not use it because once we transformed, we could not get the weight in kilograms, height in centimeters and age in years back.

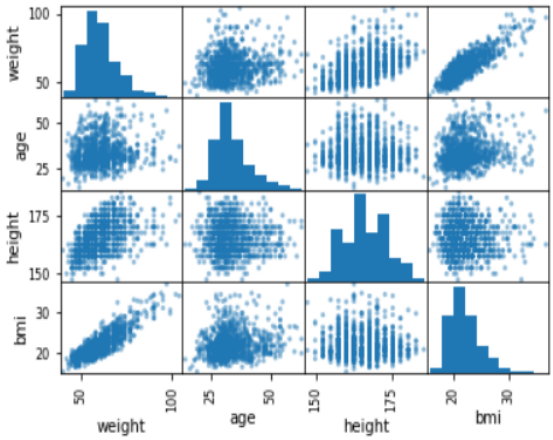




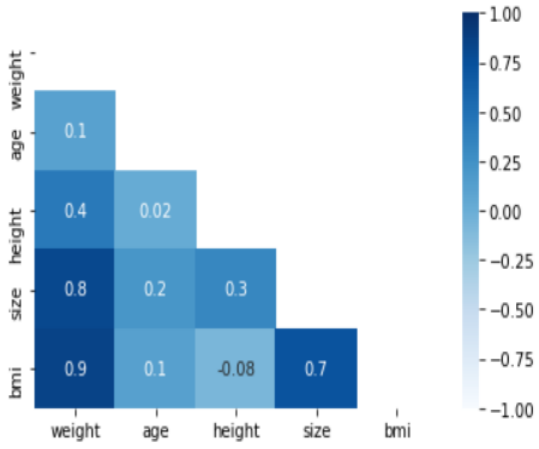
The 3D plot above describes how the sizes are located in the space. The sizes are linearly inseparable in space.



The scatter plots above indicate a strong positive linear relationship for weight and age, and weight and BMI; and no other linear relationships among other variables.



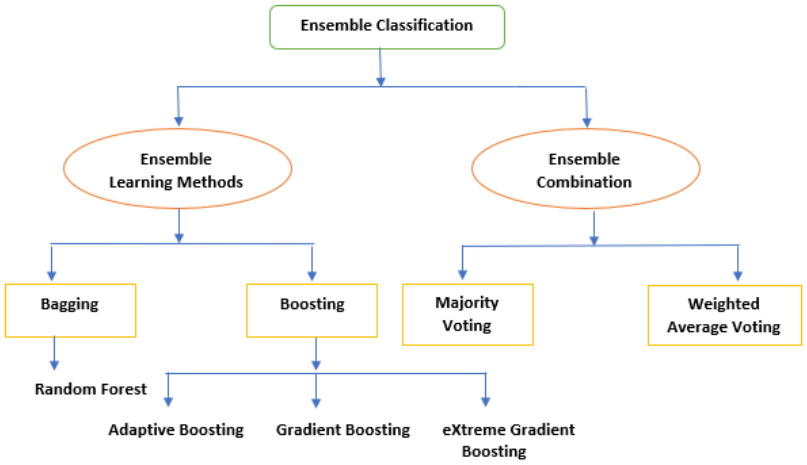
The strong positive linear relationship for weight and height is clearer using a sample of 1000 records from the first scatter plots as shown above.



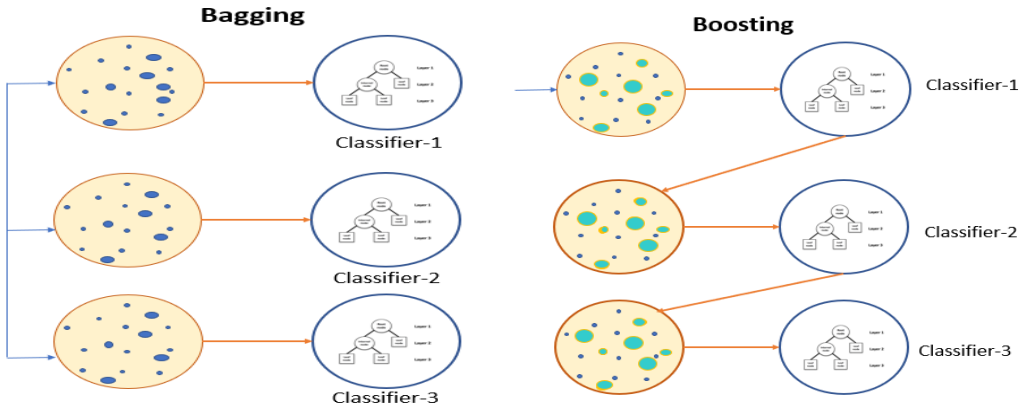
The heatmap above indicates the correlations among the variables. To get this heatmap, we first used label encoding to convert sizes from text to numbers as follows: XS: 1, S: 2, M: 3, L: 4, XL: 5, 2XL: 6. We then converted it back once heatmap was created. Based on the heatmap, we concluded that (1) sizes are strongly correlated with weights and BMIs, (2) sizes are somehow correlated with heights, and (3) sizes are lightly correlated with ages.

04. Machine Learning Algorithms

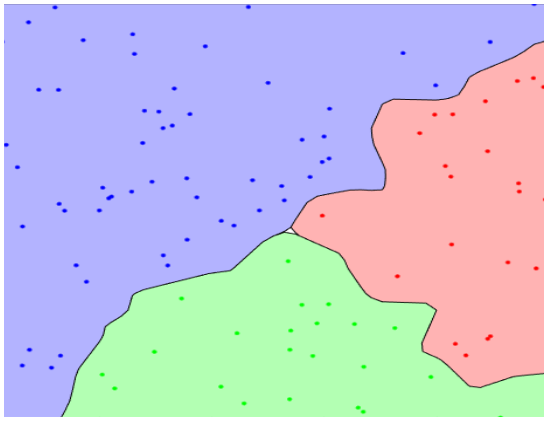
We first split the data into a train set and a test set using train test split. Particularly, we used 75% of the data to train the models and the other 25% to test the models. In order to select the best machine learning algorithms, we identified the type of the problem we are trying to use machine learning algorithms to solve. The problem we solved was multiclass classification. Multiple machine learning algorithms can be used for this such as k-Nearest Neighbors, Decision Trees, Naives Bayes, Random Forest, Gradient Boosting, Extreme Gradient Boosting, Neural Network, etc. Moreover, some algorithms used for Binary Classification such as Logistic Regression or Support Vector Machine can also be used for multiclass classification using One-vs-One or One-vs-Rest methods. We selected k-Nearest Neighbors, Decision Trees, Random Forest, and Gradient Boosting. k-Nearest Neighbors and Decision Trees are Base Learners that are used as bases for more advanced methods such as Ensemble Methods that use multiple base learners to increase the performances by scaling down the variances and biases.



As shown above, Random Forest and Gradient Boosting fall into the two types of ensemble learning methods that are Bagging and Boosting. Bagging returns a prediction from a combination of multiple base learners through a majority voting mechanism while Boosting method includes weighted models that are built sequentially by minimizing the errors from the previous ones while increasing the influence of high-performing ones.

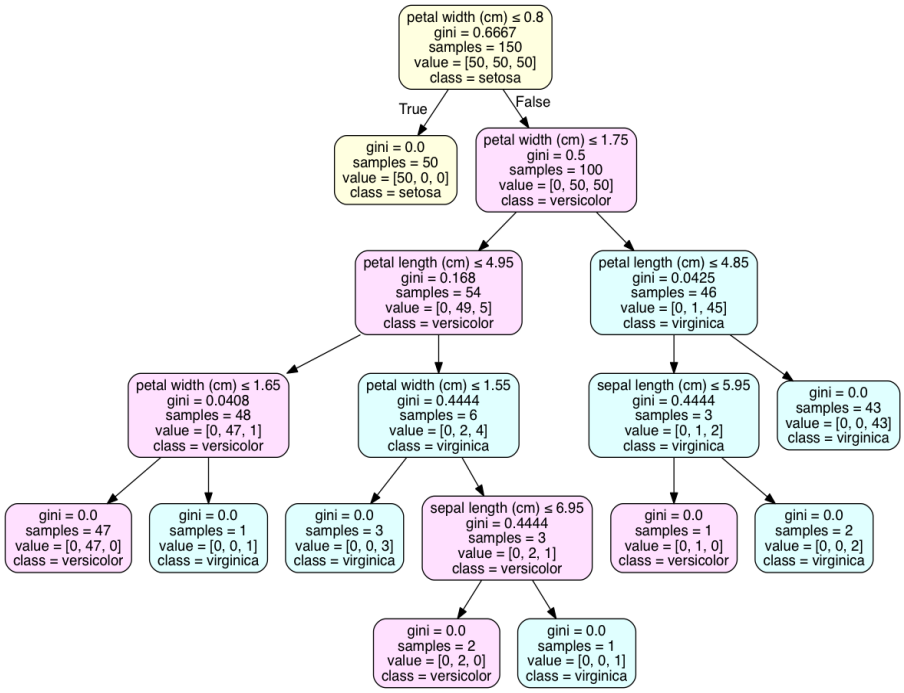


We then defined each algorithm we used together with its advantages and disadvantages.



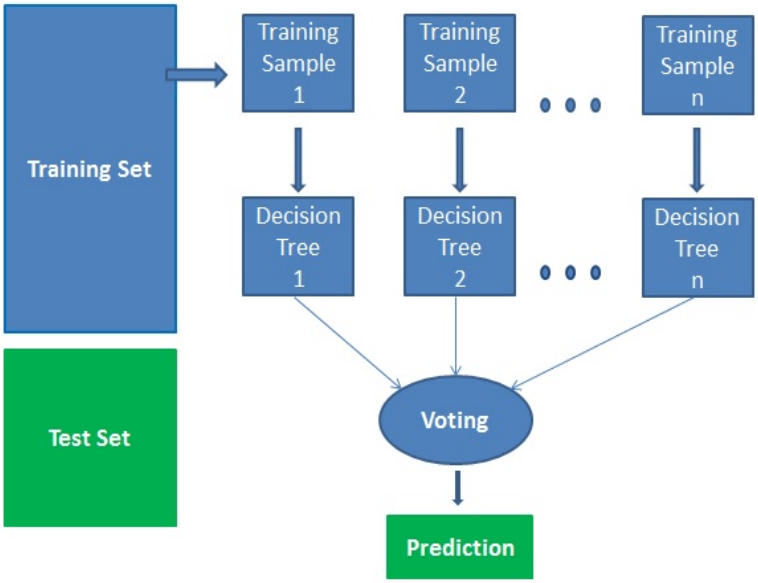
According to lecture 11 on machine learning, “k-Nearest Neighbors is a non-parametric method in which an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k

nearest neighbors (k is a positive integer, typically small).” This algorithm is easy to understand and implement. Its memory based approach allows it to immediately adapt to new training data. There is flexibility from the users side to use a distance metric which is best suited for their application. Besides its advantages, there are some disadvantages we should consider. As the training data increases, the calculation speed decreases. This model performs poorly on imbalanced data, so we need to preprocess the data in advance. This classifier has a hyperparameter called k_neighbors. If that hyperparameter is chosen incorrectly, the model will be underfitting or overfitting.



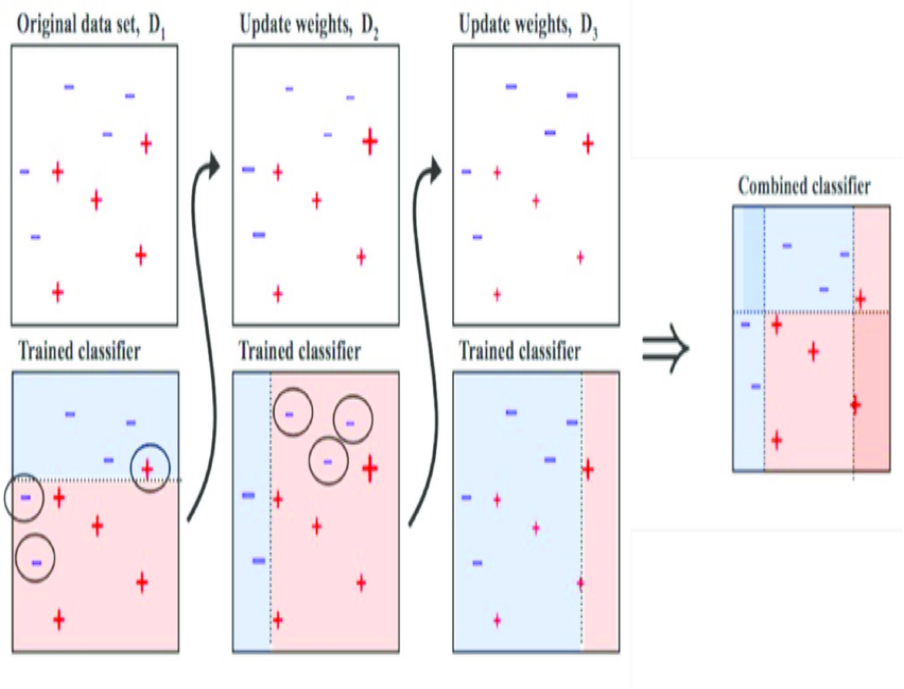
A Decision Tree is a graphical representation of possible solutions to decisions based on certain conditions. As shown above, there is a certain condition at the top of the tree-like graph that leads to other conditions based on the decisions. The process keeps repeating until we get the prediction. This algorithm is easy to implement and to visualize. We can get a decision tree like above by just a few lines of code. When we

utilize this algorithm, we do not necessarily have to preprocess the data because data preprocessing is done automatically in this algorithm. However, the calculations involved in a decision tree generally consumes a lot of memory and time. Moreover, this model is sensitively reproducible. Feature selection or feature engineering as well as changing the number of records can lead to re-generation of the overall tree and all nodes need to be recalculated and recreated. Last but not least, this model is prone to overfitting.



Random Forest is a bagging-based algorithm that sorts out a few features randomly to build a forest including multiple decision trees. The figure above might be a better explanation. In the figure above, the train set is divided into multiple train samples, each of which is used to train a Decision Tree that later returns a prediction. The final prediction is selected using the weighted average of the predictions or prediction with the highest frequency. This algorithm can handle large datasets with high dimensionality, and it has an effective method for estimating missing data and balancing errors in data sets where classes are imbalanced. Therefore, It requires much

computational power and time to make a prediction. Besides that, this model fails to determine the significance of each variable, and it is hard to visualize the this model or understand why it predicts something.



Gradient Boosting is a boosting-based algorithm that minimizes the errors in the sequential models. To explain more clearly, in the figure above, the data is trained into a weighted model in which errors are alleviated and it is then used as a dataset to be trained into another weighted model. The process repeats until all classes are classified. This algorithm performs really well if hyperparameters are tuned correctly. Data does not need to be preprocessed before implementing this algorithm. However, this algorithm is prone to overfitting. Furthermore, it is sensitive to noises.

05. Model Evaluation

We checked if we trained the machine learning models properly by taking a sample of twelve records with 2 records for each size. We then executed the train test split by default train and test size. We then used the train set for training and testing.

```
              precision    recall  f1-score   support

         2XL       1.00      1.00      1.00         6
           L       1.00      1.00      1.00         8
           M       1.00      1.00      1.00         9
           S       1.00      1.00      1.00         8
          XL       1.00      1.00      1.00         6
          XS       1.00      1.00      1.00         8

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45

1.0
```

Above are the performance reports that indicate 100% in all performance metrics. The reports for k-Nearest Neighbors, Decision Tree, Random Forest, and Gradient Boosting are shown respectively below.

```
[[3739  268   50   12  756    8]
 [ 207 2644  890  223  945   67]
 [  26 1033 2136  978  311  335]
 [   1  274  891 2276   38 1362]
 [ 707 1237  420  100 2335   34]
 [   0   19  195 1011    2 3666]]
              precision    recall  f1-score   support

         2XL       0.80      0.77      0.79      4833
           L       0.48      0.53      0.51      4976
           M       0.47      0.44      0.45      4819
           S       0.49      0.47      0.48      4842
          XL       0.53      0.48      0.51      4833
          XS       0.67      0.75      0.71      4893

    accuracy                           0.58     29196
   macro avg       0.57      0.58      0.57     29196
weighted avg       0.57      0.58      0.57     29196
```

```
[[3712  267   56   14  777    7]
 [ 193 2557  932  231  993   70]
 [  21  999 2131 1008  323  337]
 [   1  266  873 2278   42 1382]
 [ 676 1207  419  103 2394   34]
 [   0   16  192  999    2 3684]]
              precision    recall  f1-score   support

         2XL       0.81      0.77      0.79      4833
           L       0.48      0.51      0.50      4976
           M       0.46      0.44      0.45      4819
           S       0.49      0.47      0.48      4842
          XL       0.53      0.50      0.51      4833
          XS       0.67      0.75      0.71      4893

    accuracy                           0.57     29196
   macro avg       0.57      0.57      0.57     29196
weighted avg       0.57      0.57      0.57     29196
```

```
[[3739  267   50   12  757    8]
 [ 208 2641  892  223  945   67]
 [  26 1033 2136  978  313  333]
 [   1  275  889 2278   39 1360]
 [ 707 1237  419  100 2336   34]
 [   0   19  195 1011    2 3666]]
              precision    recall  f1-score   support

         2XL       0.80      0.77      0.79      4833
           L       0.48      0.53      0.51      4976
           M       0.47      0.44      0.45      4819
           S       0.50      0.47      0.48      4842
          XL       0.53      0.48      0.51      4833
          XS       0.67      0.75      0.71      4893

    accuracy                           0.58     29196
   macro avg       0.57      0.58      0.57     29196
weighted avg       0.57      0.58      0.57     29196
```

```
[[3625  215   59   11  916    7]
 [ 167 2178 1135  229 1229   38]
 [  11 1121 1992 1147  260  288]
 [   0  211 1099 2030   11 1491]
 [ 879 1332  445   94 2067   16]
 [   0   13  197 1135    0 3548]]
              precision    recall  f1-score   support

         2XL       0.77      0.75      0.76      4833
           L       0.43      0.44      0.43      4976
           M       0.40      0.41      0.41      4819
           S       0.44      0.42      0.43      4842
          XL       0.46      0.43      0.44      4833
          XS       0.66      0.73      0.69      4893

    accuracy                           0.53     29196
   macro avg       0.53      0.53      0.53     29196
weighted avg       0.53      0.53      0.53     29196
```

In the reports, there are some commonly used metrics to measure performances whose definitions and formulas are described as shown below.

Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+TP)
Precision	= TP/(FP+TP)
Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)

We used accuracy scores to measure the performance of our machine learning models. Below are the accuracy of our machine learning models.

Based on the accuracy scores, we first selected the k-Nearest Neighbors, Decision Tree, and Random Forest. We then weighed the advantages and disadvantages of each and ultimately selected Random Forest.

06. Conclusion

There are some limitations in our work and we are trying our best to make some improvements on it. Firstly, we did not do any cross validation to check for underfitting and overfitting. If any overfitting or underfitting is recognized, we will have to work on more techniques to properly address them. Secondly, though the models were trained properly, the prediction rate is still quite low; therefore we should work on more cleaning, transformation, and scaling techniques to optimize our data. For example, we can do principal components analysis to explore new features for or to select the most representative features for model training. To increase the performance, we should also consider hyperparameter tuning. We trained the models using default hyperparameters and tuned only a few of them. For example, I set the n_neighbors hyperparameter to some thousands maximum when I trained the k-Nearest Neighbors model and set the weights hyperparameter to distance. Some examples of hyperparameter tuning that can be done in the future are tuning the hyperparameter n_estimators of Random Forest Classifier and Gradient Boosting Classifier. Low performances might be caused by the way we resampled our data. The accuracy scores indeed changed each time we resampled in a new way. Therefore, we will try to, maybe, change the number of records for each size or try other resampling methods, or try new resampling methods. We will also train more models so that we can have an objective evaluation on each one and decide which one will be the best for our clothes size recommendation system. Finally, the dataset lacks context. Particularly, we don’t know many features like gender, body measurements to features like the goodness of fit or the locations of customers.

Therefore, the business must try to collect more information from customers and then use feature engineering and feature selection to process those features to make our data clean. Our clothes size recommendation system will be very beneficial to the business because of its learning ability from actual information and its ability to tell how likely a size is going to fit a customer. This is outstanding from the traditional size charts that use body shape simulation to estimate sizes and cannot tell to which extent a size fits the customer.

Data source:

https://www.kaggle.com/tourist55/clothessizeprediction

References:

https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

https://machinelearningmastery.com/types-of-classification-in-machine-learning/

https://prwatech.in/blog/machine-learning/ensemble-methods-tutorial/

https://towardsdatascience.com/types-of-ensemble-methods-in-machine-learning-4ddaf73879db

https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

https://www.educba.com/gradient-boosting-algorithm/

https://www.educba.com/bagging-and-boosting/?source=leftnav

https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/

https://www.kaggle.com/prashant111/bagging-vs-boosting

https://www.educba.com/ensemble-techniques/?source=leftnav

https://www.educba.com/random-forest-algorithm/?source=leftnav

https://www.educba.com/data-science/data-science-tutorials/machine-learning-tutorial/