Ahmedabad
University

Title:- Cloth Size Prediction Using Machine Learning

CSE523 Machine Learning
Prof. Mehul Raval

Authors

[1]Rishabh Misra Somani Shrey
Mengting Wan Aditi Vasa
Julian McAuley Vandan Shah
Ronit Shah

***Abstract*— The study presents a cloth size prediction using machine learning algorithms such as K-Nearest Neighbours(KNN), Logistic Regression and Random Forest Classifier. The proposed model aims to predict the appropriate size of clothes for an individual based on a set of input features such as body weight, height, burst size. Data set used for this study is collected from kaggle and it contains measurements of 150000 values. The data was then cleaned and processed to remove null and missing values and outliers. Exploratory Data Analysis was performed to identify outliers in the dataset and to find patterns within data. Logistic regression, Random Forest Classifier and KNN algorithm were trained on the dataset. Hyper parameter tuning was performed on KNN and Random Forest Classifier to optimize their performance.**

***Keywords*— Logistic Regression, Size Prediction Model, Correlation, Labels, Classification., Exploratory Data Analysis, KNN, Random Forest Classifier, Hyperparameter Modulation.**

I. Introduction

Cloth size prediction is an interesting and useful machine learning project that involves building a model and developing an algorithm that can predict the size of clothing that would fit a person based on their body measurements. With an increase in online shopping, such a type of recommendation system has become a necessity and it can also be useful for in-store shopping experiences by reducing the number of returns and exchanges. Variation in body shape and size can be a major factor causing the size of cloths to vary because people have different body types and proportions. Body shape and proportions can also affect how a garment fits in different areas of the body, such as the bust, waist and hips. Many times, some companies have variations in the same size of a product made by them as compared to the other company. This happens because they try to make a size available for as many people as possible. The goal of this project is to help people find clothes that fit them better, and reduce the frustration of buying clothes that don't fit. The project involves collecting data on body measurements and clothing sizes, and using this data to train a machine learning model which can identify patterns between different features and the appropriate clothing size. Overall, this project aims to improve the customer experience while shopping by providing accurate sizes and help retailers by reducing the need for returns and exchanges.

II. LITERATURE SURVEY

The problem of recommending a product's size is relatively new; only a few research have been put forth so far. Uncovering the true sizes of products and consumers and using them as features in a conventional classifier for fit prediction is one method. In contrast to this research, our approach focuses on capturing fit semantics and addresses label imbalance problems using metric learning techniques with various algorithms. To discover the hidden properties of customers and products, another recent method uses skip-gram models. This method, however, relies on platform-specific features being available, whereas our model operates on more constrained (and thus more readily accessible) transaction data. Numerous recommendation issues have previously been addressed using metric learning. In contrast, we acquire transaction representations that accurately reflect the ordinal nature of fit. Recently, metric learning and collaborative filtering were merged, and several recommendation tasks demonstrated the usefulness of this combination. The method (which is based on implicit feedback data that is binary) does not however, immediately apply to the ordinal character of the product size suggestion problem. Numerous research have combined prototyping methodologies with classification algorithms based on nearest neighbours. When compared to k-Nearest Neighbour (k-NN) classification, Köstinger et al.'s proposal to concurrently identify acceptable prototypes and train a distance metric results in improved generalisation. For the 1-NN situation, a different work offers a unique approach for generating optimal prototypes.

III. IMPLEMENTATION

The implementation has been divided into the following stages:

1. **Finalizing the data set:** We examined the fields and null values of two datasets. One of the datasets had a significantly higher percentage of null values than the other one. As a result, we chose to keep working with the second dataset because it helped us to comprehend the nature of our challenge in a better way.

2. **Removing null values:** The next step was to look through null values. We chose to discard them since a dataset with 1,90,000 items had a maximum of 10,000 null values. We attempted data imputation using the standard deviation to scale the target feature, but this had an impact on the data and prevented us from getting predictions for all three categorical variables. As a result, we used SMOTENC oversampling for the data, which proved to be very successful and significantly improved accuracy.

3. **Data cleaning:** The dataset's fields were initially all objects. For exploratory data analysis, we discovered that some of them need to be transformed to either floats or ints. The next step was to give each column a suitable field name. Additionally, we developed methods that translate a height object measured in feet and inches into a height measured in inches. Another aspect was the conversion of weight from pounds to kilogrammes for objects that were present. Features like user_id, review_rating, and product_id were eliminated because they had no utility.

4. **EDA:** Next stage was to find patterns and make a raw hypothesis on the dataset. We performed exploratory data analysis using a heat map, correlation matrices, histograms, and box and whiskers plots to do the same. It helped us study our data from multiple dimensions. Box and whiskers plot helps identify outliers and median. Using this, many outliers in features like height, weight, product size were removed. These outliers hindered in increasing the accuracy of the model and gave measurement errors. Using the correlation matrix and heatmaps, we found the most significant parameters.

5. **Encoding**: The following stage was to apply nominal and ordinal encoding on categorical data. We concluded that five columns of prime importance needed to be encoded- fit, bust size, body type, rented for, and product category. We used ordinal encoding on fit and bust size. OneHotEncoding was used on body type, rented for, and product category. When it was found that user_weight and user_height were of great importance after performing EDA and were used to create a feature called BMI. This was further used for training the data.

6. **Choice of classifier:** Given that the data was categorical, we selected logistic regression as the most fundamental approach to justify preliminary EDA results..

7. **Applying regression model:** Finally, after understanding the data and learning new patterns, we applied a regression model to the data. Through

Logistic Regression, we tried to predict the accuracy of the model.

8. **Implementing other models:** Logistic regression though gave a low accuracy so, we chose to perform k-NN and Random Forest Classifier as they are the best algorithms for classification problems since they are optimised when certain data is missing or non linear relationships are present between features and target and also as our dataset is categorical. While k-NN gave an accuracy of 72%, Random Forest gave an accuracy of 77%.

9. **Hyperparameter Tuning:** In an effort to further increase the accuracy, we tried using hyperparameter tuning. Hyperparameter tuning helps to optimize the performance by selecting the best hyperparameter. Hyperparameters are the parameters which cannot be trained but need to be set before training the data and thus help in increasing the accuracy. The algorithm we used for hyperparameter tuning is GridSearchCV. GridSearchCV is an algorithm which searches for best combinations of hyperparameters all over the grid while also performing cross validation to evaluate the performance of that combination.

IV. Results

The dataset contained many null values, so we first eliminated them all. Height has been transformed from feet to inches, and the data types for the variables height and weight have been changed to integer. After that we encoded necessary variables and changed their data types to integer. Then using EDA we determined the correlation between variables. We implemented logistic regression in our model and we got accuracy of 68%, further we tried to improve the accuracy, so we implemented Random Forest classifier as it suits our model and is more of accuracy focused algorithm. We also implemented KNN algorithm to check which algorithm gives better accuracy to our model. It was found out that random forest classifier give the best accuracy score. Still the accuracy was 77% so we did hyper parameter tuning in all the algorithms. In hyper parameter we have used the GridSearchCV approach to predict the accuracy. This further improved the accuracy of the model and we observed 94% accuracy for Random Forest and 96.5% for KNN where 10th neighbour was considered the optimised one.

V. CONCLUSION

1. Our goal was to predict whether a product of a given size will fit the user or it will be too small or large.

2. We applied logistic regression, but the accuracy was poor. To improve our results, we tried several other algorithms but still did not get satisfactory results. After trying oversampling, we achieved better accuracy for KNN and random forest algorithms, but there was still room for improvement.

3. The necessity of proper data preparation and exploratory data analysis in machine learning initiatives is generally highlighted by our study. To increase the model's accuracy, additional research is conducted by experimenting with other methods, engineering features, or optimizing hyperparameters.

4. After that we have implemented random forest classifier and KNN algorithm to improve the accuracy of our model.

5. The success of our machine learning project can be attributed to the following steps:
    a. Trying multiple algorithms to find the best fit for our data.
    b. Applying oversampling to balance the class distribution in our dataset.
    c. Using GridSearch to optimize hyperparameters for our models.

6. The results of our project demonstrate the potential for using machine learning techniques to solve practical problems in various fields, such as e-commerce, fashion, and retail.

VI. REFERENCE

[1] Clothing Fit Dataset for Size Recommendation. (2018, August 21). Kaggle. https://www.kaggle.com/datasets/rmisra/clothing-fit-dataset-for-size-recommendation.

[2] Misra, R., Wan, M., & McAuley, J. (2018). Decomposing fit semantics for product size recommendation in metric spaces. Conference on Recommender Systems. https://doi.org/10.1145/3240323.3240398

[3] Vandanshah. (n.d.). CSE523-Machine-Learning-2022-Data-Dynamos-/Codes at main · vandanshah17/CSE523-Machine-Learning-2022-Data-Dynamos-. GitHub. https://github.com/vandanshah17/CSE523-Machine-Learning-2022-Data-Dynamos-/tree/main/Codes

[4] Flowchart. (n.d.). miro.com. https://miro.com/app/board/uXjVMSsaeBA=/?share_link_id=269082615364