Run The Fashion - A Fashion Recommender System

1. Introduction

I built a recommender system which recommends the clothes to the female customers for their rental purpose. The renting purposes depend on the activities a user wants to carry out, such as: vacation, meetings, weddings, dates, etc. The recommendation is based on the features like customers' height, weight, bust size, body type, age, and reason to rent. Since a customer might not know their body type, we are predicting their body type for our calculation. 'RandomForest Classifiers' is used to predict customers' body type on the basis of their height, weight, bust size, and age. The objective of this project is to group the users by their predicted body type and renting purpose and apply collaborative filtering on them to recommend the clothes to our customer, based on the ratings of the most similar user. The goal of recommending the clothes is to improve the customers' shopping experience and preserve the ongoing trend in fashion world.

Looking towards our implementation part, we have taken the dataset called 'renttherunway' from University of California San Diego. The raw data provided by the customers comprised of categorical values, which were converted into the numerical values for calculation. In case of numerical data, customer records with the missing values are handled by applying mean-mode imputation. However, categorical data with missing values is handled by binning strategy, which is applied throughout that features of the complete dataset. This is how the data is preserved by handling the missing values and avoiding dimensionality reduction problem.

2. System Design & Implementation Details

- **Algorithm**:

Data Pre-processing:

1. Mean-Mode imputation: Used for handling missing values in features with numerical/categorical data

- E.g. 'Body Type' and 'Bust size' are imputed using mode within the bins of age

2. Binning: Used for handling missing values, data conversion into numerical format, for the features with categorical data

    - E.g. Sequential bins of 'age' are created on which the mode function is applied. 'Bust size' and 'Body Type' are filled with the mode results for the fields with the missing values.

Classification Task:

- 'Random Forest Classifier' algorithm is used to predict customers' body type, where it is trained on the features like age, Bust size, size, height, weight. Parameter tuning is done with the hyperparameter called 'n_estimators' (i.e. tree size) = 100 which yielded the best prediction result.

Collaborative Filtering:

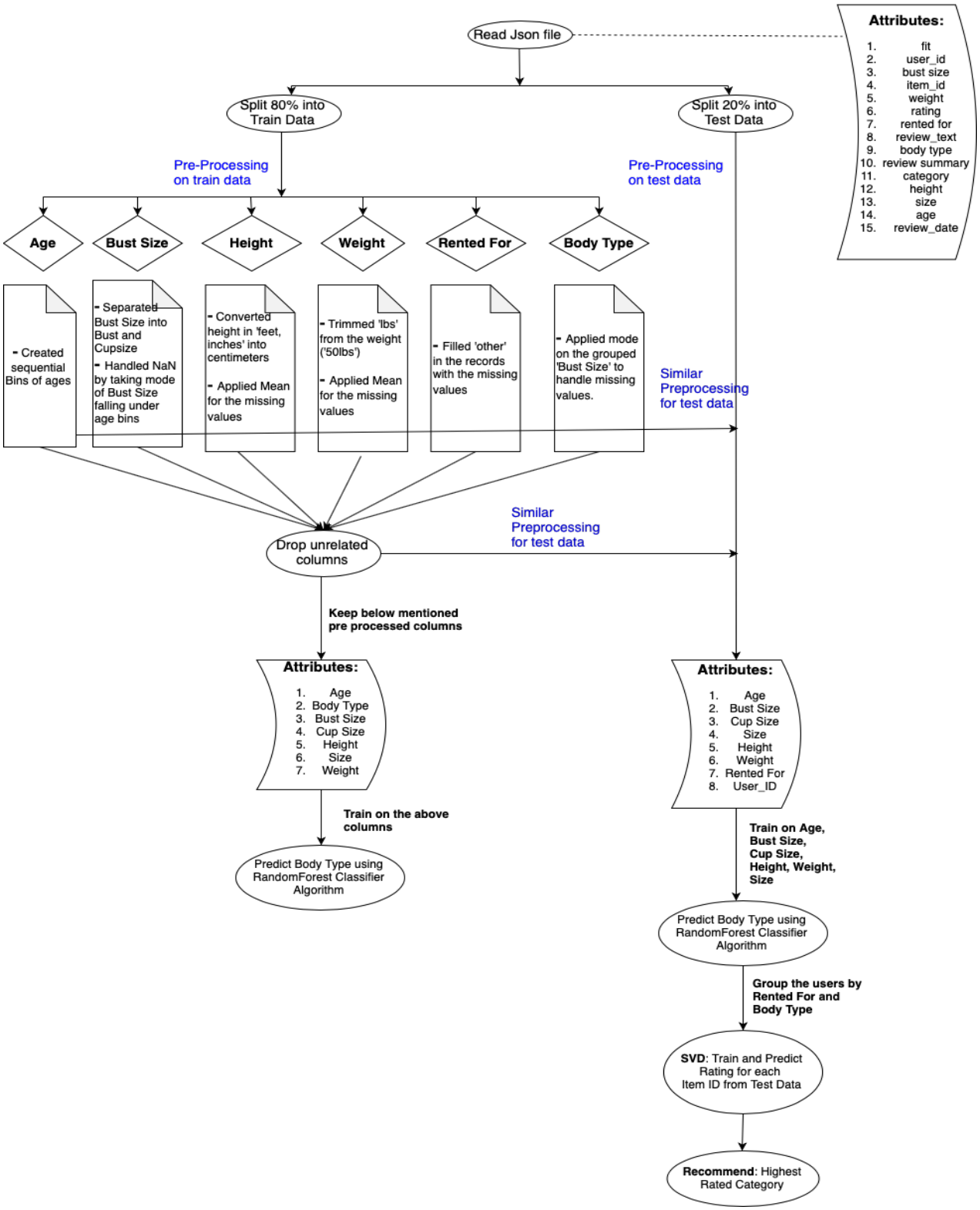
- ‘SVD’- Singular Vector Decomposition algorithm is used to predict customer ratings given the userid and itemid, from which the highly rated cloth type is used for the recommendation. SVD gave the best results with the parameters:

  n_factors=200 and n_epochs=50

  - Technologies & Tools Used:

➢ Google Collab: An online tool used for machine learning research projects, is available free for use with no setup needed except the browser availability: Chrome, Safari, and Firefox. A Jupyter notebook environment supports creating python files and running them on cloud without any intervention of local machine. Furthermore, the processing power needed for the huge computation is carried out on the cloud and not from our local machine, therefore it is a suitable platform for our project in addition of having our 123.3 MB dataset uploaded on the Google Collab cloud.

- **System Workflow :**


Read Json file
Attributes:
1. fit
2. user_id
3. bust size
4. item_id
5. weight
6. rating
7. rented for
8. review_text
9. body type
10. review summary
11. category
12. height
13. size
14. age
15. review_date
Split 80% into Train Data
Split 20% into Test Data
Pre-Processing on train data
Pre-Processing on test data
Age
Bust Size
Height
Weight
Rented For
Body Type
- Created sequential Bins of ages
- Separated Bust Size into Bust and Cupsize
- Handled NaN by taking mode of Bust Size falling under age bins
- Converted height in 'feet, inches' into centimeters
- Applied Mean for the missing values
- Trimmed 'lbs' from the weight ('50lbs')
- Applied Mean for the missing values
- Filled 'other' in the records with the missing values
- Applied mode on the grouped 'Bust Size' to handle missing values.
Similar Preprocessing for test data
Similar Preprocessing for test data
Drop unrelated columns
Keep below mentioned pre processed columns
Attributes:
1. Age
2. Body Type
3. Bust Size
4. Cup Size
5. Height
6. Size
7. Weight
Attributes:
1. Age
2. Bust Size
3. Cup Size
4. Size
5. Height
6. Weight
7. Rented For
8. User_ID
Train on the above columns
Train on Age, Bust Size, Cup Size, Height, Weight, Size
Predict Body Type using RandomForest Classifier Algorithm
Predict Body Type using RandomForest Classifier Algorithm
Group the users by Rented For and Body Type
SVD: Train and Predict Rating for each Item ID from Test Data
Recommend: Highest Rated Category

3. Experiments/ Proof of Concept Evaluation

- **Dataset used:**
  - For this project, we used RentTheRunWay dataset. RentTheRunWay is a special platform designed to allow women to rent clothes for different occasions. This dataset contains rental details for different categories. This dataset contains feedback from customers when they have returned rented clothes. The feedback are in textual and numeric rating format. The other information that this dataset depicts is: product categories, reviews, review dates and personal information including user id, age and customer's measurements. The statistics of dataset is as:

Statistics measure	RentTheRunWay
Number of transactions	192544
Number of features	15
Number of users	105508
Number of items	5850

- The RentTheRunWay dataset has been collected from the UCSD repo : http://deepx.ucsd.edu/public/jmcauley/renttherunway/renttherunway_final_data.json.gz

- **Preprocessing Performed:**

  - Handling missing/NaN values: We created bins based on ‘age’ and then imputed ‘bust size’ with mode value of bust size for the bin to which given data instance belongs. Other features are filled in with mean-mode imputation for each missing value.

  - Handling categorical features: This dataset contained many categorical/nonnumeric features like bust size, weight, height, rented for, category. We converted each of these features into numeric values. We used split and encoding methods to convert ‘bust size’ into numeric values. We used trimming method to convert nonnumeric value of ‘weight’ into numeric value. We used split function and applied formula to convert ‘height’ into numeric value in centimeters. For this, we hard encoded categorical values of ‘category’ into numerical format.

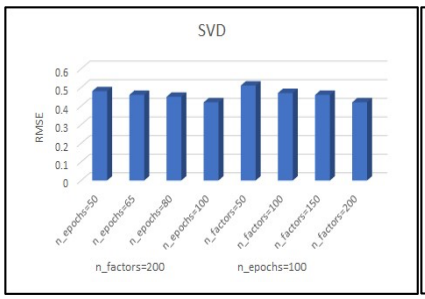
- **Data Preprocessing Decisions**

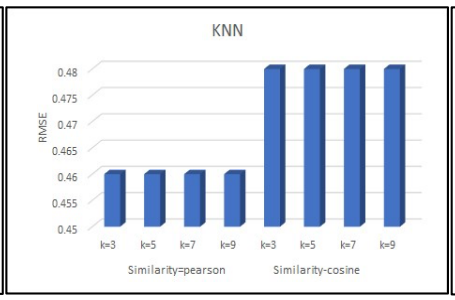
  - Mean-mode imputation: Dropping data instances with missing values would lead to dropping important data instances or data dimensionality reduction on a large scale. Thus, we decided to perform mean-mode imputation of missing values of data instances.
  - Binning: Our dataset contains attribute ‘bust size’ for which many values were missing, and it is illogical to impute them using mean or mode. Thus, we decided to first create bins of age like 5-15, 15-30, 30-50 etc. Then we put our customers into respective bins based on their age. After binning, we applied mean-mode imputation for each bin to impute missing values of bust size.

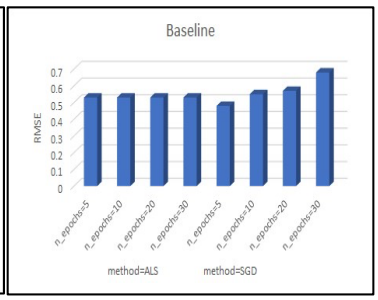
- **Methodology followed**

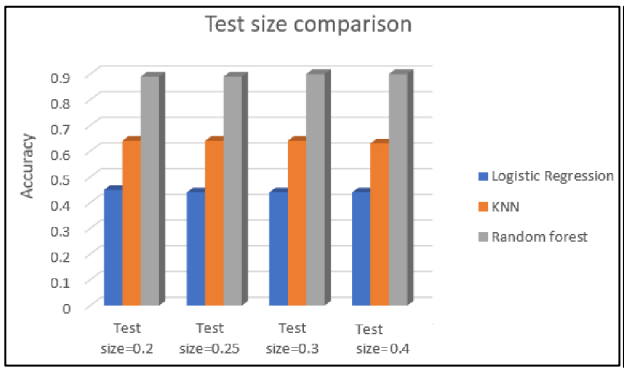
  - Train-test split: We split dataset into training and testing dataset in 80:20 ratio, as it gave the best score as compared to 70:30 and 60:40 ratio.
  - Classification system: We used Random Forest classifier for predicting 'body type' of customer based on body measurement. We also tried with KNN classifier and Logistic regression model. Random forest gave best score among them.
  - Random Forest parameter tuning: We have used 100 trees for random forest. We also experimented with 50,80 trees and 100 turned out to be the best parameter.
  - Grouping: We used groupby function of dataframe to cluster together data instances based on 'body type' and 'rented for' features. This way we achieved dimensionality reduction.
  - Collaborative filtering: We used surprise library and SVD algorithm for predicting rating of customer for each item. We performed experiments with SVD, KNN and Baseline algorithm and SVD outperformed all of them.
  - SVD parameter tuning: We tuned SVD algorithm for different values of n_factors and n_epochs parameters and for final prediction, we kept n_factors=200 and n_epochs=50.

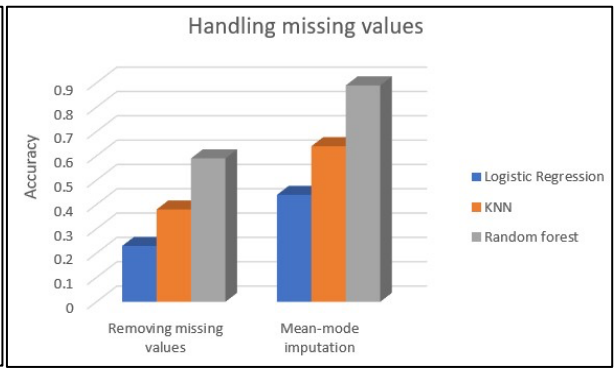
- **Graphical Output Analysis**











- In this project we evaluated different algorithms on different parameters. We experimented with different collaborative filtering techniques for 'rating' prediction like: KNN, SVD and Baseline algorithm. For parameter tuning of SVD, we used values of n_factors like 50,100,150,200 and n_epochs=50,65,80,100 as shown in graph titled 'SVD'. For parameter tuning of KNN, we used values of similarity measure like pearson, cosine and k=3,5,7,9 as shown in graph titled KNN. For parameter tuning of Baseline, we used values of methods like ALS, SGD and n_epochs=5,10,20,30 as shown in graph titled Baseline.

- We also evaluated the performance of classifiers: KNN, logistic regression and random forest for prediction of 'body type'. We first experimented with splitting dataset in train-test dataset using different sizes of test data like 0.2,0.25,0.3,0.4 etc. and train size is complementary to test size as shown in graph titled 'Test size comparison'.

- We evaluated the performance of classifiers: KNN, logistic regression and random forest for different techniques of handling missing values like removing data instances with missing values, mean-mode imputation as shown in graph titled 'handling missing values'.

- **Analysis of Results**

- From graph with title ‘SVD’, we can see RMSE values obtained for different combinations of parameters and hence, we conclude that SVD works best when n_factors=200 and n_epochs=100.
- From graph with title ‘KNN’, we can see RMSE values obtained for different combinations of parameters and hence, we conclude that KNN works best when similarity is pearson coefficient and value of k does not affect performance.
- From graph with title ‘Baseline’, we can see RMSE values obtained for different combinations of parameters and hence, we conclude that Baseline gives constant performance for method ALS and varying performance for SGD.
- When we compare all three collaborative filtering techniques, we found SVD has the best accuracy, hence we I used it.
    - From graph with title ‘test size comparison’, we can conclude that all classifiers maintain constant performance on any way splitting of train-test dataset and hence we used default test size for our project which is test size=0.25.

    - From graph with title ‘handling missing values’, we conclude that all classifiers show improved accuracy if we perform mean-mode imputation for missing values instead of removing missing values. Thus, we have used mean-mode imputation technique for our project.

    - From graphs with titles ‘test size comparison’ and ‘handling missing values’, we conclude that Random Forest classifier outperforms among all classifiers and hence, we used Random Forest classifier technique to predict ‘body type’.

4.Discussion & Conclusions

- **Decisions Made**

    - The given dataset was not clean data. So, we did preprocessing as mentioned above. While doing preprocessing we observed that more than 14000 records were having missing bust size values. This column was one of the main deciding factors of body type. So, we decided to impute those values based on age by binning the data.
    - After preprocessing our main task was to find out the body type of person depending on height, weight and bust size. We could have used the body type of person which was already there in database but we decided to use classifier to get the body type as persons body type may change. So we used machine learning (supervised learning) algorithms. We tried out different algorithms and found out that Random Forest algorithm was working good on our data among KNN, Logistic Regression and Random Forest classifier.
    - After getting the body type of person our next goal was to give recommendations based on the body type, we got from the classifier and the purpose of rent which is provided by the user. So instead of running algorithm on whole data we decided to reduce dimensions in order to get good response time. In order to do that we grouped the data based on the body type and rented for value which reduced the number of rows from dataset drastically and got relevant rows.
    - At last our main aim was to give recommendations to the user. So, we did collaborative filtering. For this, we tried SVD, KNN and Baseline algorithms. Then after looking at the results obtained from these algorithms, we observed that SVD performed well and we finalized the SVD algorithm.

- **Difficulties Faced**

    1) In pre-processing we faced difficulties while pre-processing bust size column.
    2) All values were not numerical in data, it took time to convert those values to numerical format.
    3) While doing classification, we had to try different algorithms as the algorithms which were in our mind at the start did not work well.

- **Things that Worked**

    1) We pre-processed the data successfully with minimal removal of rows.
    2) We were able to apply different data pre-processing techniques.
    3) We successfully did classification based on height weight and bust size using Random Forest algorithm.
    4) We were able to reduce the size of dataframe by grouping the data according to body type and rented for.
    5) At last we successfully recommended clothes to user using collaborative filtering.

- **Things that didn't work well**

    - We were trying to use clustering algorithm for grouping the data. But we ended up grouping data without clustering algorithm.

- **Conclusion**

This project successfully gives recommendations to user depending on the body type and rented for values. User does not need to know body type, they have to just enter the height, weight and bust size and the model automatically find out the body type of user and gives recommendations.