CMPE 256 - Project Report

Classifying fit of product

Name of team: The Standard Deviants

Members: 012584819 - Aakash Manishbhai Thakar

013530530 - Devanshi Trivedi

013748917 - Reeya Vani

Link to source:https://github.com/trivedidevanshi/CMPE256LargeScaleAnalytics

Link to dataset: http://cseweb.ucsd.edu/~jmcauley/datasets.html

INDEX

Chapter 1. Introduction

1.1 Motivation and Objective

Different products come in different sizes. All these sizes, however, have different fit. The sizes of different brands have a different fit. Even the items in the same brand have a different fit depending upon the material, type, and category of the item. For example, the size of Reebok shoes 7 is 15cm and that of Nike is 16cm. Hence to a particular customer, Reebok shoes might be fit(or small) and Nike shoes will be large(or fit). Similarly, customers may have different fit preferences. Some customers might find an item's fit small which others find it proper. Because of these discrepancies, customers prefer not to buy items from online websites without trying out the item physically.

So we propose to classify a fit of an item based on user preference and item similarity. For this, we considered two datasets ModCloth and Renttherunway. In our project, we have applied three classification algorithms to classify the fit of an item in {small, fit, large}. For understanding the classification algorithms, their behavior, and the impact of the dataset on these algorithms, we conducted a series of experiments, whose results are discussed in a later section. For experiment purposes, we first applied the algorithms individually on the two datasets and noted the results. Later we merged the datasets and noted the results. While preprocessing, we realized that datasets are biased towards size_review as fit. So we preprocessed the data and conducted the experiment on both biased and unbiased data (after preprocessing) and noted the result.

Chapter 2. System Design & Implementation

2.1 Algorithms Considered

1) **Tensorflow Keras**

Keras is TensorFlow's high-level neural network API for building and training deep learning models to train our model. We chose this algorithm because the network trains by adjusting the weights to predict the correct class label of input samples which leads to high tolerance to noisy data, as well as their ability to classify patterns on which they have not been trained.

This algorithm, assembles neural layers to build a model. We have implemented a model which is a stack of layers: the tf.keras.Sequential model. The features column is given as an input to the model. After constructing the model, we started the learning process by calling the compile method. The model is trained using RMSprop optimizer to minimize Loss function. The model is "fit" to the training data using the fit method. The data is run for multiple epoch. An epoch is one iteration over the entire input data. The result is then evaluated to calculate the accuracy achieved.

The dataset has numerical and categorical data. The categorical data is converted into numerical using feature_column library of tensorflow which uses one hot encoding. Now that we have defined feature columns, we use a DenseFeatures layer to input them to our Keras model. Using DenseFeatures makes our layers fully connected. Each neuron in a layer receives an input from all the neurons present in the previous layer.

2) **Logistic Regression Model using the tf.estimator API**

It is a statistical model that uses a logistic function to model a binary dependent variable. The data is fit into TenserFlow’s linear regression model, which is acted upon by a logistic function to predict the target categorical dependent variable.

Since it is easy to implement and efficient to train, we started off with this model and then implemented other models to improve the accuracy achieved. The dataset and label are given as an input to a function that converts the data into TensorFlow dataset which is added to an input pipeline. It is run for multiple iterations. On each iteration, the model is trained on shuffled data in order to achieve data variance.

A Linear classifier is created using the feature columns array. This model is then used to train the function created for test dataset. Finally, evaluation is done on validation and test data to obtain the accuracies.

3) **Deep Neural Network Classifier**

A deep neural network is an artificial neural network with multiple layers between the input and output layers. It finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. It is different from the neural network in terms of the number of hidden layers. Deep neural networks have multiple hidden layers and not one.

The selected categorical features are converted to numerical and an array of the features is fed as an input to the training function. We then build a DNN with 2 hidden layers of 30 and 10 nodes respectively. The model then chooses between 3 ‘fit’ classes. The classifier is then trained and evaluated to obtain the accuracy.

2.2 - Technologies and Tools Used

- Tensorflow
    - Keras
    - Linear Classifier
    - DNN Classifier

  To create a model, train it using the training dataset and evaluate it to find the accuracy on test dataset.
- Sklearn ‘s model selection library to split the data into train, test, and validation
- Pandas - to load the json data into a dataframe and preprocess it.
- Google Colab - to share files with other developers remotely and it provides free GPU service which is useful for large data evaluations

2.3 - System Workflow

- Data observation and preprocessing.
- For each of the datasets- ModCloth and RentTheRunway we performed the following evaluation:
    - Implemented the three algorithms to compare accuracy achieved for each dataset.
    - Observed that the data is biased towards the ‘fit’ of the cloth
    - To check our model for unbiased data, we considered limited number of data which is ‘fit’ for the user.
    - Performed the three algorithms again to compare the accuracy for biased and unbiased data.
- The same processing and evaluation are done for the combined dataset of ModCloth and RentTheRunway.

Chapter 3 - Experiments

3.1 DataSets Used

The datasets used contains cloth fitting feedback from customers along with details like the reviews, ratings, product categories, sizes and customer measurements.
We have used these features to determine if an item will be a small, large or fit.
The two datasets and its sizes are:

Table 1: Datasets used

	ModCloth	RentTheRunway	Total
Number of users	47,958	105,508	153,466
Number of items	1,378	5,850	7,228
Number of transactions	82,790	192,544	275,334

3.2 Data Preprocessing

By looking at the head of the data, we made a few observations and preprocessed it accordingly:

- There are multiple null values for bra_size, cup_size, and hips which are important features in training our model. Therefore features were filled with negative values.
- The height is in feet and inches. It should be in one size unit. So, we defined a function to convert the height into cm.
- There are features with multiple missing values which affects model training. These columns were dropped.

	total_missing	perc_missing
item_id	0	0.000000
waist	78831	95.218022
size	0	0.000000
quality	64	0.077304
cup_size	0	0.000000
hips	0	0.000000
bra_size	0	0.000000
category	0	0.000000
bust	69872	84.396666
height	0	0.000000
user_name	0	0.000000
length	34	0.041068
fit	0	0.000000
user_id	0	0.000000
shoe_size	54133	65.385916
shoe_width	63278	76.431936
review_summary	6621	7.997343
review_text	6621	7.997343

	total_missing	perc_missing
fit	0	0.000000
user_id	0	0.000000
bust size	18411	22.238193
item_id	0	0.000000
weight	29982	36.214519
rating	82	0.099046
rented for	10	0.012079
review_text	0	0.000000
body type	14637	17.679671
review_summary	0	0.000000
category	0	0.000000
height	677	0.817732
size	0	0.000000
age	960	1.159560
review_date	0	0.000000

ModCloth data for missing values RentTheRunway data for missing values

3.3 - Methodology

1. The data is preprocessed to remove the null values and convert the data into numerical form from categorical so that it can be used as feature columns to train the models.
2. Since the data is biased, we manipulated it to create unbiased data. All the algorithms were used to test this data and observe its effect on accuracy.
3. We changed following parameters in DNN and Keras Sequential to increase the accuracy:
    a. Learning rate: Decides how much weight are we giving on parameter. If we set it too high then cannot converge and will overfit.
    b. Regularizer: Controls how much importance we give to training set and validation set
    c. Epoch: The number decides how many time we shuffle our data and train on neurals
    d. Layers: The layer helps in deciding the weights predicted by the model.

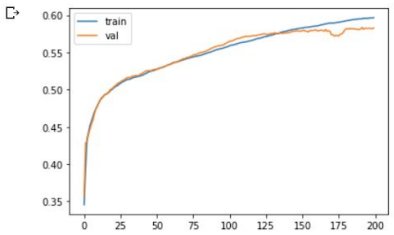
4. We implemented one hot encoding on the label data i.e the fit data to reduce the processing time.

3.4 - Graphs for accuracy comparison

We just produced graphs for just one algorithm that is keras Sequential.
Accuracy graph for unbiased dataset of renttherunway

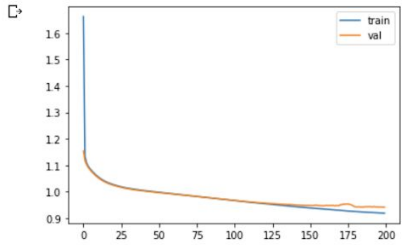
```
[47] from matplotlib import pyplot


    pyplot.plot(history.history['accuracy'],label = 'train')
    pyplot.plot(history.history['val_accuracy'],label = 'val')
    pyplot.legend()
    pyplot.show()
```



Loss graph for unbiased dataset of renttherunway

```
[48] pyplot.plot(history.history['loss'],label = 'train')
     pyplot.plot(history.history['val_loss'],label = 'val')
     pyplot.legend()
     pyplot.show()
```



3.5 - Observation of experiments and reasons

The table below describes the accuracies achieved. The columns represent the datasets on which training models were run. The datasets are ModCloth data, RunTheRunway data and a combined dataset of both these data.

Both the datasets are biased. To evaluate the model, we have manipulated the data to make it unbiased and have calculated the accuracy for each.

On each dataset, we have implemented three algorithms, Keras Sequential, DNN, and Logistic Regression, and calculated the accuracy achieved for each.

Table 2: Accuracies achieved for different algorithms

Algorithms	ModCloth		RentTheRunway		Total (ModCloth +RentTheRunway)
	Biased Data	UnBiased Data	Biased Data	UnBiased Data	Data
Keras Sequential	0.683	0.7088	0.7487	0.575	0.720
DNN	0.687	0.680	0.742	0.615	0.729
Logistic Regression	0.664	0.678	0.761	0.619	0.734

From the analysis above, the observations are:

1. The accuracy achieved for RentTheRunway is more as compared to that of ModCloth because it has a bigger dataset size. Thus, it is trained on a larger dataset.
2. Since we manipulated to dataset to create an unbiased data, the data size reduced. Due to this, the accuracy achieved for unbiased data is less as it is trained on reduced data size.
3. The accuracy achieved for RentTheRunway is more as compared to that of ModCloth because it has more number of features which are used for training the model.
4. We observe that the accuracy for DNN and Keras Sequential algorithms is almost the same. This depends on the parameters set for these models like epoch, layers, learning rate and regularization. The accuracy can be improved by testing the model for different sets of parameters.
5. The accuracy can also be improved by using more features to train the model. One of the features that can be used is text mining of the user reviews.

Chapter 4 - Discussion & Conclusions

4.1 Decisions Made and Difficulties Faced

1. We implemented the three algorithms and determined which one is best suited for our dataset.

2. After deciding on the best algorithm, we implemented one hot encoding to improve the accuracy.

3. We tried to implement PCA on the dataset to determine which features are best suited to determine the accuracy of the dataset. However, we faced difficulties while integrating it with tensorflow libraries.

4.2 - Approached that did and did not work

1. Our approaches for Three different algorithms worked well.

2. One Hot Encoding on 'fit' on the data reduced the processing speed.

3. We also tried to implement Boosted Trees algorithm for the dataset but it did not work as pruning is not supported with multi-class. It is a technique which reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances

4.3 - Conclusion

1. We implemented three different algorithms on the dataset and measured the accuracy achieved for each. The model build using tensorflow Keras which sequentially ran the data on multiple neural layers, gave the maximum accuracy of around 75% for biased dataset.
   The algorithm with second-best accuracy achieved, 74% was the DNN Classifier.
   The Linear Classifier achieved the least accuracy.
   So, we concluded that Tensorflow's Keras algorithm for neural network is best suited for our dataset.

2. The data size of RentTheRunway is more than that of ModCloth. So, training the model with a bigger dataset, gave better accuracy. This leads to the conclusion that more the data available to test the model, better is the model trained.

3. "More the number of features we feed to the training model, better is the accuracy achieved". We concluded this by using different feature columns of the two different datasets for model training. RentTheRunway, having more number of features, achieved a higher accuracy.

4. The given dataset is biased. Training a model on a biased dataset leads to overfitting which negatively impacts the performance of the model on new data. To check our model, we manipulated the dataset to make it unbiased. The accuracy dropped by 9%. Even though the accuracy dropped, it was 61% which is a really good measure considering the reduced data size.

Chapter 5 - Project Plan and Task Distribution

1. We started off by discussing various problems faced in our day to day life and also at the same time, searching for datasets available which can be used to train a model to solve those problems.
2. We came across a common problem in online shopping where the cloth fitting suggested is not what actually fits the user. So, we planned to create a model that will be trained to check if the fit suggested is correct for a particular user size and height.
3. One of the datasets was ModCloth data which had a set of user characteristics and if the item fits the customer.Then we observed and determined the different processing which needs to be done on the raw data.
4. We implemented different algorithms to train the data. To implement neural networks, we used TensorFlow libraries.
5. To reduce the processing time, we included one hot encoding and regularizer while optimizing the data

Work done by Team members

Aakash:
For data processing, I evaluated the missing values in the dataset and cleaned the dataset for those feature columns. I worked on implementing Neural network algorithm using TensorFlow's Keras library. To decrease the runtime, I implemented one hot encoding on the 'fit' of the data. Also, I created the accuracy graphs for the algorithm. To improve accuracy, I implemented regularizer on the Tensorflow Keras Algorithm.

Devanshi:
I cleaned the dataset of null values and change the categorical height column which had data in feet and inches to numerical data(cm). So that it can be used as a feature column in my training model. I implemented Logistic regression model on the dataset and determined the accuracy. Displayed the accuracy using graphs. Also, I tried to implement PCA and one hot encoding on each column using sklearn library.

Reeya:
I converted the categorical columns into numeric columns so that it can be used as a feature to train the model. Also, I observed that the dataset is biased and so manipulated it to form an unbiased dataset. Used these datasets to prevent overfitting of data and determined accuracy. I implemented DNN Classifier algorithm to determine the accuracy. Also, I tried to implement the Boosted tree algorithm for classification and I vectorize review_text by calculating TF*IDF score of the word to use it as a feature.

APPENDIX

References:

1. https://www.tensorflow.org/guide/keras/overview
2. https://www.tensorflow.org/tutorials/estimator/premade
3. https://www.tensorflow.org/tutorials/estimator/linear
4. https://www.tensorflow.org/tutorials/structured_data/feature_columns
5. https://www.kaggle.com/agrawaladitya/step-by-step-data-preprocessing-eda/code
6. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
7. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html