Data Science: Principles and Practice
Final Report

December 2020
2171 words

Abstract

Here I present a report with four main points on the "Clickstream data for online shopping"[1] from the UCI ML Repository [**1**]. The task was to build a machine learning pipeline to predict whether a customer paid a premium price for the product selected.

1 Data exploration

In this section, I present a general description of the dataset, continue with exposing insights gained from exploring it, and finish off with observations of the features that are most useful for the predictive task, based on the previous findings.

1.1 General Details

The dataset presents clickstream information from April to August 2008 from an online clothing store for pregnant women. It has a number of 165,474 instances, each with 14 features. Each row contains information at a that specific click of a specific session ID such as: the page number within the website, the main clothing category (5), the code for the product (217 products), its colour (14 in total), its location on the page (1 of 6), the type of the photo (en face or profile), the price in US dollars and whether or not that price is higher than the average price for the entire product category. The last one is the one we are interested in.

1.2 Relationship between premium price and other features

Overall, 51% of the entries are annotated with a premium price. Looking at category (trousers, skirts, blouses, sale) in Figure 1, I observe a strong preference for paying more than the average price for skirts. A reason behind this could be the comfort provided by skirts and the femininity associated with them, which makes the spender willing to pay more. Additionally, most of the customers are from European countries where the months of April to August represent spring and summer, a warmer time than the rest of the year, suitable for skirts.

In the opposite direction, I see a strong preference for paying less for trousers. Trousers are also the most popular item here, as they make up almost 50,000 instances of the all the items bought and are at least 28% more popular than any other category. An intuition behind this can be the fact that trousers are loose, free and easy clothes, fitted for almost all functions (e.g. official, casual and evening maternity wear) and weather. Hence, they are an essential element in every pregnant woman's wardrobe. Still, most are only purchased to be worn during pregnancy. It then makes sense for the customers to want to pay far less for each.

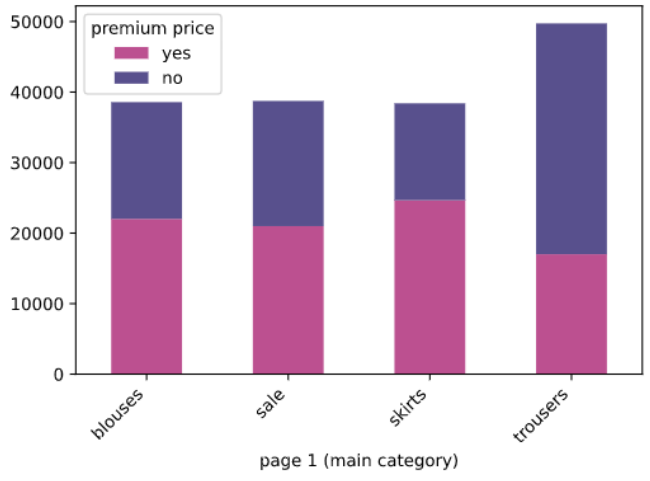

Figure 1: *Main category versus frequency of premium and non-premium prices paid.*

Regarding the remaining attributes, I noticed an exceptional preference (more than 80% of the time) for paying premium prices for clothes of colours olive, pink and red.

[1] `https://archive.ics.uci.edu/ml/datasets/clickstream+data+for+online+shopping`

On the other hand, colours beige, burgundy and green are strongly associated (over 70%) with non-premium prices. From the remaining attributes, I discovered that customers slightly favoured (between 54% and 60% of the time) profile pictures, images located on the top-left and top-right of the page and the 5th website page (in contrast with the 3rd page) when it comes to the premium price ratio.

1.3 Unique observations

Poland, other EU countries and other European (non-EU) all have the same five top colours, in similar rankings as well. Four of those appear in the top five of countries outside of Europe as well. Interestingly enough, beige is the second most popular colour in countries outside of Europe, but it's at least number 8 on any other region. Generally, for the entire dataset, the favourite colour is black. It is a preferable colour for pregnant women, because of its slimming effect. Almost as frequent as black is blue, followed by gray, brown and white, but with a very large gap, as seen in Figure 2.

1.4 Feature correlation and selection

By inspecting the correlation column of the feature keeping record of whether the price is premium or not we get the highest absolute value of 0.74 in relation to price. The rest of the absolute values are below 0.14, significantly smaller.

However, assuming that an e-commerce website owner wants to be able to decide on prices for new products using a classification model on premium price, then the actual *price* column becomes irrelevant, as does the *product code*, since a new one won't be present in the training set. I plan on further investigating the effect of these two features in the next section.

Additionally, *year* is a constant feature and will be discarded. Furthermore, the histograms of *month* and *day* portray a decreasing pattern in the total number of items being bought, as per Figure 2. However, when inspecting percentages for premium items, I observed only very similar values, all around the global 51%, so the premium price might not be affected by the date in this dataset. The range of the months presented here is likely not large enough to capture multiple-season patterns. In the next section, I plan on deciding whether *month* and *day* are worth keeping or not.

2 Machine learning algorithms implementation

Here I embark on a quest: "The search for the holy model". I cover the application of machine learning algorithms learned, and make continuous changes to my approach according to the results.

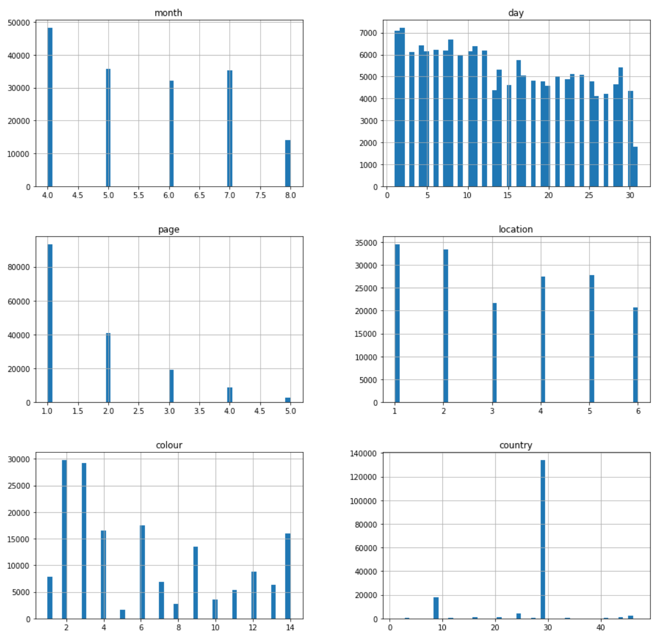

Figure 2: *Observable patterns in histograms*

2.1 Preprocessing

Fortunately, there are no missing values in this dataset.

Order The *order* column is monotonically decreasing in counts. The last 94 clicks (from 101 to 194) have each a maximum of 16 appearance. In total they add up to 486 entries, less than 0.003% of the entire dataset. To avoid overfitting on the clicks appearing in very small numbers, I choose to only keep clicks numbered 1 to 100 and drop the rest.

Countries There country column presents 47 values, but 80% of the dataset is represented by customers from Poland. The problem above arises here as well, since some countries only appear in a handful of examples. As a result, I will group them into four regions: *Poland, other EU countries, other European countries (non-EU), countries outside of Europe and unknown*, as per [**1**]. and turn *countries* into a categorical feature.

Numerical VS Categorical Values The product code is the only categorical feature in the dataset. However, there are some discrete numerical features, such as the *clothing category, colour* and *model photography* that would wrongly make learning models assume that close values encode similar concepts. One-hot encodings are appropriate for these and the categorical *product code*. For the remaining numerical ones, I will apply the *Min-Max Scaler*, as it is most appropriate for the sets of values here.

Sets of attributes I want to find an appropriate subset of features for the task. To address my suspicions about *price* and *product code*, I will shortly explore their influence in order to report on their possible effects on generalisation. If they do produce unrealistic expectations, they will be dropped.

Stratifying data The most disproportionate features, in terms of the frequency counts, are *region, colour*, and *page*, as per Figure 2. Since they are important features, I choose to stratify the dataset with respect to them in a nested fashion, by first adding an extra column concatenating them and then applying classic stratification on that. I will also add the premium choice, despite it being fairly proportionate, to make sure the splits are stratified on the target value as well.

2.2 Classification Algorithms and Ensemble Learning Techniques

I start by challenging the simple classifiers presented in the second practical: *Perceptron (Stochastic Gradient Descent)*, *Logistic Regression* and *Gaussian Naive Bayes* to take on the task at hand. I also examine the effect of the *kernel trick* on the *Peceptron* to check for improvement. A 80/20% training-test split is appropriate.

Training with price and product code

In a drastic overfitting fashion, the *Perceptron* , *Logistic Regression* and *Naive Bayes* all achieved accuracy of 1 or extremely close to 1 both when *price* or *product code* (or both) were present in the training features. This clearly suggests the models wouldn't generalise well, hence these two attributes will be disregarded. There remain 10 features to play around with. I plan on producing results on the full set, but also drop some more features, to further the search for a suitable subset of the entire feature set.

Training with the remaining 10 features

Performance Table 1 shows the fight put up by these classifiers, judged by 5-fold cross-validation. The first row addresses all 10 features, while the second looks at only 7 (*session ID, month*, and *day* are being trialed on their usefulness). As it turns out, they don't make a difference. However, when evaluating the *kernel trick*, the accuracy does rise from 59% to 71% with only 7 features, giving us the best model of this subsection.

Table 1

	SGD	**LogReg**	**GNB**
10 features	0.56	0.67	0.68
7 features	0.57	0.67	0.68

2.3 Ensemble Learning Techniques

With the newly gained insight on the feature set, I continue the search for a knighthood-worthy algorithm with only 7 attributes.

Simple voting classifiers: hard and soft voting strategies I populate the jury with three classifiers: *Logistic Regression, Random Forest Classifier,* and *Support Vector Machines.*

The ensemble learning techniques have risen to the challange, for it seems that the *Random Forest Classifier* on its own had an accuracy of 92.8%, beating both the hard and the soft classifiers themselves. We've come a long way from the straightforward classifiers above, but at the price of an extremely long computation time.

Less computation-hungry ensembles Between a Random Forest with 500 trees, a Bagging Classifier on Decision Trees with out-of-bag evaluation, Adaptive Boosting on Decision Trees and Gradient Boosting, the Bagging Classifier performed the best. The performance of the other choices varied, but still beat the simple classifiers. I officially declare the Bagging Classifier with out-of-bag evaluation (93% accuracy) the winner of our search and choose to further evaluate it in the next section.

2.4 Neural Networks

The popularity of Neural Networks as the ultimate model was almost justified in this case. Here I split the data in three: train, validation, test. I trained several networks with different numbers and shapes of Dense layers with 100 epoch. I got results that were very close to the one obtained by the Bagging Classifier, but not better than it, despite the lengthy training times. Still, it makes sense, given that the reduced feature set simplifies the problem enough for other algorithms, and one should now not to abuse the power of Neural Networks, as they are not necessarily fit or needed for every single problem. Hence, I stand by my choice of crowning the Bagging Classifier as the winner of this round.

3 Evaluation

In this section, I employ several evaluation measurements on the chosen classifier. I hope to further justify the choice made in the last section.

3.1 Accuracy

The accuracy of the Bagging Classifier on the test set it is 92.9%

Table 2

Precision	Recall	F1
0.929	0.928	0.928

3.2 Precision, Recall and *F1* score

As it's visible in Table 2, the values on the weighted test set are almost identical, hence the classifier is harmonically balanced between trustworthiness and coverage. The confusion matrix in Figure 3 complements the results above.

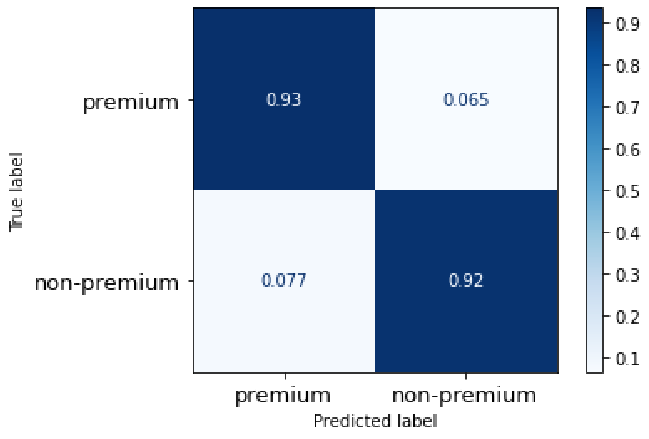

Figure 3: *Confusion matrix*

3.3 Result

The model chosen, a non deep-learning one, trained on only 7 features, each with very small direct correlation to the target value itself, has performed impressively well. In all fairness, the *Bagging Classifier* was also heavy on computation time, but still acceptable compared to other classifiers (such as the *Voting Classifiers*). There isn't much possibility for overtraining since all of the attributes used have a decent number of appearances for every one of their values and most have only a small number of unique labels. That, combined with the very similar performance in the training and test sets, suggests that the algorithm would generalise well. Considering again that each individual feature doesn't hold much influence over whether or not a premium price was made, I would expect the model to be acceptably robust to noisy data.

4 Visualisation and dimensionality reduction

This section is concerned with whether or not I can use dimensionality reduction in my advantage. I explore *PCA: Principal Component Analysis* and *t-SNE: t-distributed stochastic neighbor embedding.* The goal is to find an alternative representation of features (embedded ones) to reveal properties of the dataset. I have only kept the initial numerical values. In thinking the numerical values that truly represent categorical data would do more harm than good, I started by investigating the plots with different subsets of the features. This continuously produced aggressive overlapping. Finally, I gave in and fed the algorithms all the raw numerical values which resulted in more distinguishable clusters.

4.1 Principal Component Analysis

Figure 4 shows how the points are evenly distributed across the First Component axis and how around the 0-value of the Second Component acts as a delimiter for all the *non-premium* labeled data. Hence, the Second Component captures that value well.

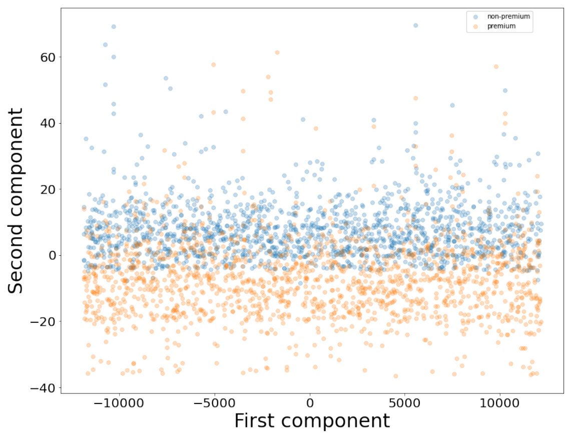

Figure 4: *PCA plot of premium (orange) versus non-premium (blue) prices. prices*

4.2 t-distributed Stochastic Neighbor Embedding

I couldn't find a way to use *t-SNE*'s black magic into producing clusters of premium choices. However, well-defined clusters can be seen at high perplexities (Figure 5). Given how many categorical features were disguised as numerical ones, it is fair for this embedding to have picked up some of those attributes and cluster accordingly.

4.3 Result

Unfortunately, since no helpful interpretation was obtained without the *price* or the *product code*, the embeddings here couldn't be subsequently used as a preprocessing step for a Machine Learning model.

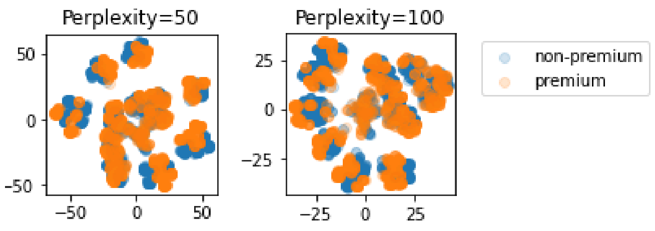

Figure 5: *Clusters of t-SNE embeddings, each with its respective perplexity value*

5 Conclusion

Clickstream datasets are *hot* right now, since every retailer wants to be able to read the minds of their customers in order to continuously increase their profits. For this report, I analysed one such dataset, discovering distinctive patterns in the data, filtered out impractical atttributes in a trial and error methodology, played with a wide variety of classifying mechanisms, and then further evaluated the best one.

References

[1] Mariusz Łapczyński and Sylwester Białowas, 2013 *Discovering Patterns of Users' Behaviour in an E-shop - Comparison of Consumer Buying Behaviours in Poland and Other European Countries p. 144-153.* Cracow University of Economics, Poland