UDACITY

MARCEL JACQUES MACHADO

MACHINE LEARNING: CAPSTONE PROPOSAL

CURITIBA

2017

CONTENTS

1 DOMAIN BACKGROUND

"If your business is not on the Internet, then your business will be out of business". This Bill Gates quote said many years ago reflects our current reality. Indeed he was right and nowadays the Internet allows us to buy and sell all kind of goods from almost all business categories without leaving home. No one wants to be out of business, right?

The most popular online stores negotiate hundreds of thousands of different products. And there are hundreds of stores and millions of customers interested in buying online. They just need to find what they want for the price they can afford.

But with so many different online stores, products and prices how could a customer to be sure that he is making a good deal? Could he do a relevant search to select just a complete group of products with the exactly features he wants and then compare these features and prices?

2 PROBLEM STATEMENT

Let’s say we want to help the customers to find the best offers on the Internet building some kind of product comparator to them. Maybe there are already some sites like that. But maybe their systems are too slow. Or maybe they show a lot of irrelevant products in their searches. Or maybe there are some good stores missing in their datasets. Whatever. We have some reasons to do that.

And to do that we basically need to group similar products from different stores together and provide this organized data to the customers. Moreover, it should be possible to the customers to sort the products of a selected group by price.

In this way the customers that want to find the products with the best cost-benefit wouldn’t need to visit each online store individually. So they wouldn’t be just saving money, but time as well. And time is money…

But unfortunately it wouldn’t be so easy to create the product comparator. I have at least one difficulty on my mind. And this major problem is the classification of the similar products from different stores into a same category. And this task is completely necessary for our system.

Note that different online stores can have very different categories for a same product. For example, a TV could be classified in a store as Smart TV, LED TV, Monitor TV, 4K TV, QLED TV, OLED TV or LCD TV. And in another store, the same TV could be classified as Portable TV, Curve TV, Android TV, Nano Crystal TV etc.

So with so many possibilities, which category labels we should use? And a more important question: would there be a way to group similar products together if they don’t even belong to the same category label? Let’s take a look at the dataset first.

3 DATASETS AND INPUTS

Thanks to affiliate programs we can easily get parts of the datasets of the online stores. These datasets are available for anyone that has a website and would like to place the store’s advertisements/products there in exchange for making some money. The webmasters receive a commission for every purchase made by their users that clicked on the ad and bought in the advertiser store.

To be more specific we will use datasets from Awin, a global affiliate network with 6,000 advertisers and 100,000 contributing publishers. Among the advertisers, we have the most important e-commerce sites from Brazil. We will be limited to my country for now due to restrictions of access to foreign datasets. So the datasets of Brazilian stores will be the input of our product comparator.

About the datasets, each store provides a CSV file with some common columns such as name, price, brand, category, URL etc. Some fields can be null but at least name and price are mandatory. The largest dataset I could get belongs to Walmart and it has 2.6 millions of products.

So now we have all those datasets, how to use their information to solve our classification problem? Is the available data enough for our product comparator to come true?

4 SOLUTION STATEMENT

The creation of our product comparator will go through six steps:

a) get the datasets;

b) solve the classification problem;

c) reorganize and merge the datasets into a new one;

d) code the web application;

e) put that on the Internet;

f) automate the three first steps to update the datasets once a day.

Even thought all steps seem to be interesting, only one is unfeasible to be achieved both for traditional programming and for human effort: the step b.

I'm saying that because we have dozens of datasets with millions of products and every day these datasets change. New products are added and new categories are created. There is no way to predict these changes and consequently there is no way to classify the products. At least not without the use of Artificial Intelligence.

For this reason the classification of products will be the scope of this Machine Learning project. And even though I'm determined to complete all the six steps, the step b will be the only one to be covered here.

So we are going to use Machine Learning to solve our problem. Or in other words, a computer is going to learn how to classify those millions of products without being explicitly programmed for that.

5 BENCHMARK MODEL

To help us to know if we are on the right track, we will use as a benchmark model the academic paper “Applying Machine Learning to Product Categorization” by Sushant Shankar and Irving Lin from the Department of Computer Science of the Stanford University. As the title says they tried to solve a very similar problem using a very similar solution.

In this short but valuable paper we also can find an awesome information about the usage of this type of Machine Learning model in industry. We figure out that the product classifier would need an accuracy range of 95% to be accept by small-to-medium sized companies. Maybe this can be our optimistic goal.

Moreover the paper provide details about the datasets, data preprocessing, algorithms, impediments, perceptions and results, including accuracy and training/prediction time. There are also a list of suggestions that weren’t implemented but promise to improve the model, such as the usage of the product images as an input of the classifier. Python was the programming language and Orange the Machine Learning library.

About the results, to make a long story short, their model was able to classify thousands of products in 3 seconds with an accuracy of 79.6% using Naive Bayes algorithm; using K-Nearest Neighbors, 4 minutes and 69.4% of accuracy; using a Tree classifier, 8 hours and 86% of accuracy.

Could our Machine Learning model overcome these results? We really hope so. But only the evaluation metrics can answer that.

6 EVALUATION METRICS

During the development of our product classifier, we will make a lot of changes in the dataset structure and try different algorithms with different parameters until we get the best result. But what would be the best result?

To compare different versions of our model we need some evaluation metric to say if we are improving our classifier or if it’s getting worse. And once we are trying to solve a typical classification problem (and not a regression problem because our expected output isn’t continuous) we can use F1 Score.

F1 Score is an evaluation metric which takes into consideration both precision (number of correct positive results divided by the number of all positives results) and recall (number of correct positive results divided by the number of all true positives) of the test to compute the score (the harmonic mean of them). The best score value is 1 and the worst is 0. As you can see, our F1 Score metric is equivalent to the accuracy of our benchmark model.

So that will be our main evaluation metric even though the usage of other ones during the development process is not completely ruled out.

7 PROJECT DESIGN

The development of our product classifier probably will go through three main steps that will be repeated in this sequence again and again until we get a very good model:

a) improve the dataset;
b) train and test the model with different algorithms and parameters;
c) evaluate the model.

In the first step we have insertion, removal and update of data. The dataset for sure will need to be normalized or standardized and maybe we will have to remove some outliers or add new features. Some irrelevant words could be removed and plural ones changed to singular. It’s possible that the dataset of a store will be much better for training/testing than others. And it’s possible as well that some datasets need to be ruled out. Some additional research can be required in this first step once mistakes in the data transformation may interfere with the operation of the algorithms.

In the second step we will try different supervised learning algorithms to fit the data and make predictions, such as Linear Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbors, Decision Tree and Neural Network. About the parameters issue, we can use Grid Search, an approach to parameter tuning, to improve our model. And maybe K-fold cross-validation as well, depending on the training time, that could help us to avoid overfitting.

Finally, in the third step, our evaluation metric will measure our progress. It’s certain that we will have to go back to the previous steps multiple times until we have a good F1 Score here.

So that’s our workflow, that will be executed with the help of Python (programming language), Scikit-Learn (Machine Learning library) and also Pandas, NumPy and maybe Matplotlib (additional libraries). A Linux terminal with all those commands (grep, sed, cut, awk, wc, head etc) for sure will be very useful to handle the datasets. And Jupyter Notebook will be considered for presentation purposes.

To finish, and I don’t know if it will be possible, due the short deadlines, I would like to have in the end of the project not only the product classifier, but a link for our complete product comparator available on the Internet. So let’s start?

REFERENCES

ERICSON, Gary; ROHM, William A. **How to evaluate model performance in Azure Machine Learning**. 2017.

LIN, Irving; SHANKAR, Sushant. **Applying Machine Learning to Product Categorization**. Department of Computer Science, Stanford University, Stanford, 2011.

MACHADO, Marcel J. **Model Evaluation & Validation: Predicting Boston Housing Prices**. Project – Machine Learning Engineer Nanodegree, Udacity, 2017.

MACHADO, Marcel J. **Supervised Learning: Building a Student Intervention System**. Project – Machine Learning Engineer Nanodegree, Udacity, 2017.

WIKIPEDIA. **Awin**. 2017. Accessed on December 16, 2017.

WIKIPEDIA. **F1 Score**. 2017. Accessed on December 16, 2017.