Master in Artificial Intelligence
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Money Tracker
Final Report

Gilles Charlier
Jason Klimack

2021-01-13

Contents

1 Executive Summary

1.1 Problem Statement

Throughout history, money has always been an important feature in society. It is the currency that allows a person to own and possess goods. It allows a person to purchase both necessary items such as food, water, shelter, and clothing, as well as luxurious items such as movies, books, beauty items, fancier foods, etc. However, quite often we find ourselves over-spending the money that we earn, and left wondering what the money was spent on in order to reduce unnecessary expenditures.

To solve this problem, we had the idea to create a money tracker application that would allow a person to know exactly how much money they are spending, and where they are spending it. The core idea to our solution is to allow a user to take a picture of their receipt after every purchase, and later be able to view the different types of items that they bought in a series of graphs. This will allow the user to know which categories of items they are spending an unnecessary amount of money on.

1.2 Task Analysis

There are two major tasks to undertake to apply our solution: 1) extracting text information from an image of a receipt and storing the information in a database, and 2) automatically classifying the different products that were bought on the receipt to a predefined set of categories. Each receipt can contain several products purchased at a time, and not all of the products necessarily belong to the same category (eg. a loaf of bread belongs to the food category, while a bus ticket belongs to the transportation category). Hence, during the text-extraction step each product needs to be extracted individually from the receipt. During the product classification step, a machine learning algorithm will be used to predict a category label for each of the products on the receipt. The ML model will be trained using a dataset consisting of product name-category pairs from Amazon.

The different categories that we will train our model on are:

- Appliances
- Electronics
- Grocery Gourmet Food
- Home Kitchen
- Musical Instruments
- Office Products

- Patio Lawn Garden
- Pet Supplies
- Sports Outdoors
- Tools Home Improvement
- Other

The user will be allowed to manually add new categories, but they will have to add manually the information from the receipt for those categories.

Finally, after creating the text extraction and product classification steps, a user interface will be built that allows a user to select an image and add the contents of the image to a database. The information from the receipt that needs to be stored in the database is the type of products bought, and how much money was spent on each type of product. This information can then be used to create a set of graphs/diagrams that can be displayed to the user to visually show where the user spends all of their money.

1.3 Requirement Analysis

The main goal of the system is to classify the product names into different categories, such that the user can visually see the amount of money they spend on each category. Therefore, the system must be able to correctly classify a product name into its associated category (ie. bread is a food), as well as extract the proper information from the receipt image, and display the information to the user. There must also exist a user-interface that allows the user to interact with the system. The user interface must provide the user with capabilities to select an image, extract the information from the image, and store the image in a database. Furthermore, the user interface must also be able to provide feedback to the user in the form of graphs, representing the amount of money the user has spent on each product category.

2 Economic analysis

First we need to make a list of human and material resources that we will use. We will make the assumption that we are working in a small enterprise that owns two normal computers but no powerful computer to train the model. With this in mind, the economic requirements of the final system are the following:

- Two programmers in artificial intelligence, working in total for 180 hours (3 ECTS credits each corresponds to 90 hours of work each).

- Powerful Computer
- Receipts printed at shop's checkout.
- Two computers to make the project.
- Internet connection.
- Commercial use of python external libraries.
- Support and bug fixing after delivery.

Let's analyse the cost of each element step-by-step.

2.1 Working hours

If we try to evaluate the work that will be done during these four months, we can estimate that both programmers will work each 90 hours on the project. This estimation includes mid term deliverable and the development of the final product. So if we make a small calculation, we will have :

$$26000/(254 - 20) = 111.11$$

where 26000 euros is the average raw salary according to Glassdoor[1], 254 is the number of working days in 2020 [2], and 20 is the average number of vacation days for a programmer in a random enterprise, which gives us 111.11 euros per working day for a programmer.

$$111.11/8 = 13.889$$

To compute the number of euros per hour of work a programmer is earning, we need to divide this amount by 8 (the average number of hours per working day). This gives us 13.889 raw salary per hour. Then we need to compute how much it will cost in total :

$$13.889 * 90 * 2 = 2499.975$$

Where 90 is the number of hours per programmer, and 2 the number of programmer. This gives us an average of 2500 euros for the human resources, support excluded.

2.2 Material resources

Powerful computer

Text recognition on receipts and expenses classification are not the hardest deep learning task. We shouldn't expect to use GPU or server power to train our model. Already existing libraries python will extract the text from the ticket, and our job will mainly consist of classifying these extracted expenses, which can be done on a good desktop PC. This kind of PC costs around 900 euros nowadays. The more we invest on it, the faster will be the training of the model, and the more variety of tickets we will be able to classify. That is also why we do not include electricity costs into account in this report.

Two computers to make the project

These computers are already owned by the enterprise, so we will not take this expense into account.

Internet connection

We will not take into account this expense.

Commercial use of external python libraries

Some external python libraries cost a fee in order to use in a system. However, we only used free libraries, so we will not take this value into account.

Receipts printed at shop's checkout

We should not expect any cost from this resource.

Support

If we decide to deploy the system as soon as it is ready, we need to provide support to the customers. Fortunately we can't evaluate right now how much people we will need in the support team, but it needs to be taken into account later.

2.3 Conclusion

For now we have only taken into account human resources, and the price of a powerful computer. This gives us a sum of $900 + 2500 = 3400$ euros.

3 Preliminary environmental and sustainability analysis

When talking about environment's footprint of the project, we need to think about devices and environment costs that we will use, and how any positive or negative impacts that these devices may have on the environment. Here is a list of what we will use :

- A powerful computer to train the model
- Receipts printed at shop's checkout
- Two computers to make the project
- a team of two programmers that will program the system

The first major impact on environment is the use of electronic devices such as computers. Materials of electronic devices that are reusable and not specifically bought for this project are not taken into account, since it's hard to evaluate how much we will use these devices. But we can take into account the power consumption of them.

Firstly, two simple computers will be used to make the system. These two computers are specifically made to be a good trade-off between power consumption and computing power, since they are designed to work on battery. 100W each.

Secondly, one powerful computer that will train the model. Nowadays powerful computers that are able to train models are desktop or high-end laptops PC. The power consumption of these models will be around 5 times higher than a standard laptop PC since it is made to perform heavy tasks . (500 W)

We also have to take into account the fact that the user will have to ask for receipts each time he is buying something, which is not really eco-friendly. An alternative would be to train the model on bank transfer details directly, but our model does not take this into account.

Finally, users try to manage and reduce their money consumption, which means that overall they will realise where they are spending too much money. We can therefore expect that users will reduce their expenses. This effect will have a positive impact on environment due to the reduces amount of expenditures.

To summarize, we have both positive and negative impacts on the environment. On one hand, we need devices that we use everyday (except for the powerful computer) and paper receipts that are slightly bad for the environment. However, the overall goal of the system is to get users to reduce the amount of items they purchase, which provides a positive impact on the environment.

4 Scheduling

In this work, we have successfully completed a money tracker application. This section describes the gantt diagram, showing the time taken to complete each section of the work. The work consistsed of three main deliveries (milestones): D1, D2, and D3. D1 was the project description, consisting of theoretical background information, as well as the planning of the project. D2 was the midterm delivery. The midterm delivery contained the initial code for the project, as well as a presentation describing the current progress, and a report. D3, the final delivery, contains the final functional code for the application, as well as a presentation of the final product. Furthermore, the delivery also contains three documents: Final Report, Technical Manual, and User Manual.

Figures 1 and 2 show the final Gantt diagrams. The milestones are separated based on the color of the diagrams. D1 and D3 are represented as green plots, while D2 is represented as a yellow plot.

References

[1] *Sueldo: Programmer en Barcelona*. es. URL: https://www.glassdoor.es/Sueldos/barcelona-programmer-sueldo-SRCH_IL.0,9_IM1015_KO10,20.htm (visited on 10/14/2020).

[2] *Business days calculator in Spain — Cataluña*. URL: https://www.dias-laborables.es/EN/Catalu%C3%B1a.htm#a20 (visited on 10/14/2020).

[3] Google. *Getting started with Natural Language Processing: Bag of words*. 2019. URL: https://www.youtube.com/watch?v=UFtXy0KRxVI&t=206s&ab_channel=GoogleCloudPlatform (visited on 01/13/2021).

[4] *Stemming and Lemmatization in Python*. Oct. 2018. URL: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python (visited on 01/13/2021).

[5] Sara Robinson. *Interpreting bag of words models with SHAP*. URL: https://sararobinson.dev/2019/04/23/interpret-bag-of-words-models-shap.html (visited on 01/13/2021).

[6] *Amazon review data*. URL: https://nijianmo.github.io/amazon/index.html (visited on 01/13/2021).

[7] *Fraud Detection Contest : Find it !* fr-FR. URL: http://findit.univ-lr.fr/ (visited on 01/13/2021).

[8] Thiago Carvalho. *Basics of Kernels and Convolutions with OpenCV*. en. July 2020. URL: https://towardsdatascience.com/basics-of-kernels-and-convolutions-with-opencv-c15311ab8f55 (visited on 01/13/2021).

[9] Chloe Artaud et al. "Find it! Fraud Detection Contest Report". en. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. Beijing: IEEE, Aug. 2018, pp. 13–18. ISBN: 978-1-5386-3788-3. DOI: 10.1109/ICPR.2018.8545428. URL: https://ieeexplore.ieee.org/document/8545428/ (visited on 11/18/2020).

[10] Michael McTear, Zoraida Callejas, and David Griol. *The Conversational Interface*. en. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-32965-9 978-3-319-32967-3. DOI: 10.1007/978-3-319-32967-3. URL: http://link.springer.com/10.1007/978-3-319-32967-3 (visited on 10/15/2020).

[11] Edda Leopold and Jörg Kindermann. "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?" In: *Machine Learning* 46.1/3 (2002), pp. 423–444. ISSN: 08856125. DOI: 10.1023/A:1012491419635. URL: http://link.springer.com/10.1023/A:1012491419635 (visited on 10/15/2020).

[12] M Ikonomakis, S Kotsiantis, and V Tampakas. "Text Classification Using Machine Learning Techniques". en. In: (), p. 10.

[13] Bharath Sriram et al. "Short text classification in twitter to improve information filtering". en. In: *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*. Geneva, Switzerland: ACM Press, 2010, p. 841. ISBN: 978-1-4503-0153-4. DOI: 10.1145/1835449.1835643. URL: http://portal.acm.org/citation.cfm?doid=1835449.1835643 (visited on 10/15/2020).

[14] Kowsari et al. "Text Classification Algorithms: A Survey". en. In: *Information* 10.4 (Apr. 2019), p. 150. ISSN: 2078-2489. DOI: 10.3390/info10040150. URL: https://www.mdpi.com/2078-2489/10/4/150 (visited on 10/15/2020).

[15] Ritika Mishra and Navjot Kaur. "A Survey of Spelling Error Detection and Correction Techniques". en. In: *International Journal of Computer Trends and Technology* (2013), p. 3.

[16] Hitoshi Iida, Eiichiro Sumita, and Osamu Furuse. "Spoken-language translation method using examples". en. In: *Proceedings of the 16th conference on Computational linguistics -*. Vol. 2. Copenhagen, Denmark: Association for Computational Linguistics, 1996, p. 1074. DOI: 10.3115/993268.993369. URL: http://portal.acm.org/citation.cfm?doid=993268.993369 (visited on 10/15/2020).

[17] Elizabeth D Liddy. "Natural Language Processing". en. In: (), p. 15.

[18] Anh Duc Le, Dung Van Pham, and Tuan Anh Nguyen. "Deep Learning Approach for Receipt Recognition". In: *Future Data and Security Engineering*. Ed. by Tran Khanh Dang et al. Cham: Springer International Publishing, 2019, pp. 705–712. ISBN: 978-3-030-35653-8.

[19] Tao Wang et al. "End-to-End Text Recognition with Convolutional Neural Networks". en. In: (), p. 5.

[20] J. M. White and G. D. Rohrer. "Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction". In: *IBM Journal of Research and Development* 27.4 (July 1983), pp. 400–411. ISSN: 0018-8646. DOI: 10.1147/rd.274.0400.

[21] H. K. Anasuya Devi. "Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script". In: *Ancient Asia* 1 (Dec. 2006), p. 161. ISSN: 2042-5937. DOI: 10.5334/aa.06113. URL: http://www.ancient-asia-journal.com/articles/10.5334/aa.06113/ (visited on 10/15/2020).

[22] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. *Optical Character Recognition*. 1st. USA: John Wiley & Sons, Inc., 1999. ISBN: 0-471-30819-6.

[23] Rafi Ullah et al. *OCR Engine to Extract Food-Items, Prices, Quantity, Units from Receipt Images, Heuristics Rules Based Approach*. Feb. 2018. DOI: 10.13140/RG.2.2.16640.74242.

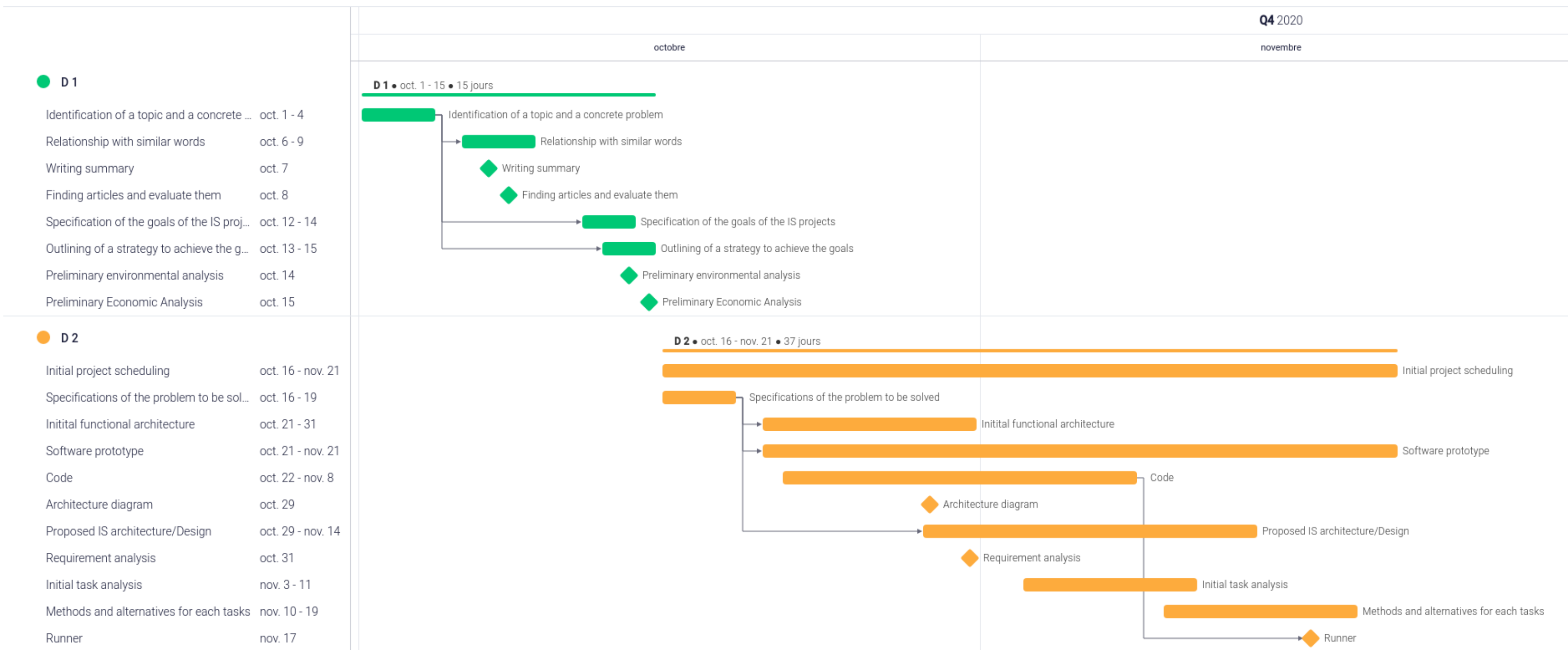

Figure 1: D1 and D2 Gantt diagram

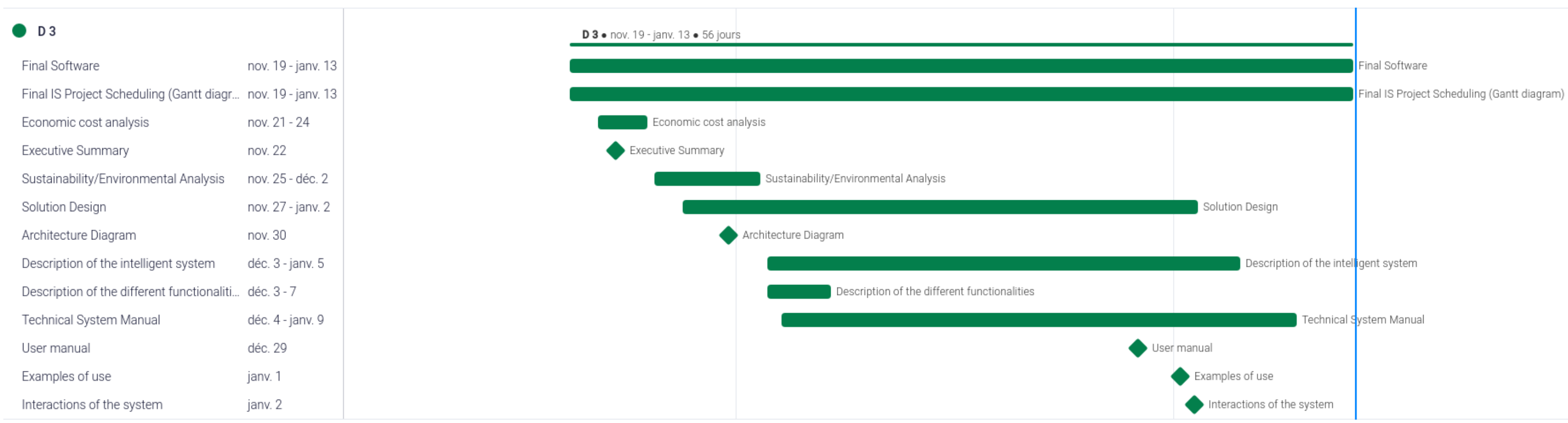

Figure 2: D3 Gantt diagram