Capstone Project: Proposal for Multi-Class Text Classification

UDACITY MACHINE LEARNING ENGINEER NANODEGREE

Sérgio da Costa
October 30, 2020

Domain Background

With the exponential growth of volume and diversity of data available in organizations, with cheaper, more powerful, faster and more effective processing, with most accessible storage, we have seen a growth in the demand for Machine Learning (ML) solutions as it's possible to produce more quickly and automatically models capable of analyzing complex and huge volumes of data while delivering faster and more accurate results.

When I talk about ML models, and considering the purpose of this particular project, I consider following the classification tasks (supervised learning) for **Text Classification** [1].

When we talk about organizing text, we are basically saying that we intend to organize them into certain subject topics, or specific categories. There are several examples where we can find this organization, the products of a company are organized by category or family of products, customers can be segmented into several categories based on a set of relevant features, or the categorization of tickets that report problems. On the other hand, we have binary classification problems such as sentiment analysis, bank fraud, forum moderation, moderation of classified or e-commerce sites, email spam identification, among many other examples [2].
With the use of ML models, we are able to solve several problems.

In this context, the main objective of this project is to identify the best model to classify complaints about financial products and services [3] [4] from customers into already predefined classes. For this I will evaluate several ML models [1] [4] [5] and identify the best performance so that we can automate a solution that automatically classifies all the complaints that arise on a daily basis.

It's important to mention that my main motivation for this project is to work with several supervised ML models to solve text classification or multi-classification problems. This is because over the last few years I have had several classification problems such as mapping generic products with a specific catalogue, processing data collected in UX research areas through surveys, in several languages, for a possible classification in specific tags predefined in the organization, or even make the sentiment analysis of these responses, or classification in 3 hierarchical levels of categories of the tickets we collect daily in Zendesk, etc.
I would be happy to work on a set of real data from the organization where I work, but for confidentiality reasons this is not possible. So, as an alternative, I chose to select the dataset I refer to below, and the goal is to work on these ML concepts and new algorithms and find solutions that can be applied on a professional level.

Problem Statement

Based on the selected dataset (described below) we are faced with a supervised (multi-class) text classification problem, as there is a classification base already in place. The main goal is to understand within the various ML methods for classification, which are the most appropriate to solve this problem.

And why do I want to solve this problem?

Besides my personal motivation presented above, being in a production environment, every time a new product or financial service complaint comes in, the idea is to automatically classify this complaint to one of the existing products (categories), without the need to do the manual classification, often time-consuming and requiring human resources to do it, a boring task that after a few days leads to demotivation of people.
I'll try to identify the best supervised model of classification (with a high accuracy) to automatically classify only one product and subproduct.

Datasets and Inputs

For this project I will use a dataset called Consumer Complaint Database (Consumer Finance Complaints (Bureau of Consumer Financial Protection)) which can be downloaded from Kaggle [6] or Data.gov [7].

"These are real world complaints received about financial products and services. Each complaint has been labeled with a specific product; therefore, this is a supervised text classification problem. With the aim to classify future complaints based on its content, we used different ML algorithms can make more accurate predictions (i.e., classify the complaint in one of the product categories). The dataset contains different information of complaints that customers have made about a multiple products and services in the financial sector, such us Credit Reports, Student Loans, Money Transfer, etc. The date of each complaint ranges from November 2011 to May 2019" [6].

Dataset structure and resume

Figura 1 - Dataset structure [6]

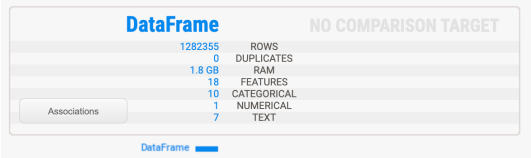

Figura 2 - Resume generated by Sweetviz

Solution Statement

As mentioned in the *Problem Statement* section, the approach proposed here will be to train a supervised model (comparing several models that will be mentioned below), to classify the complaints that enter daily. Once I have trained and identified the best model, I can automatically classify all the complaints in the training set.
Then I compare the characteristics of a certain classification with the training classifications.

In the end I will have a model capable of predicting the best product for the complaints that will enter the system in the future.

Benchmark Model

As I mentioned before, the solution is to identify the best ML model for text classification, for the existing dataset. These types of problems, where there is a knowledge base, are considered supervised problems:

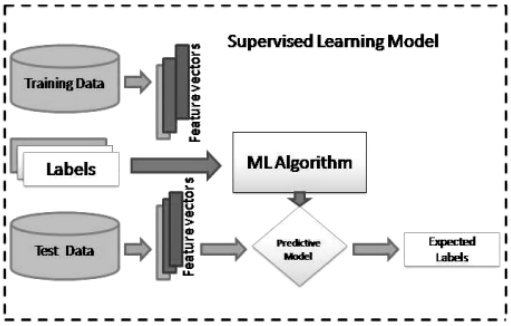

Figura 3 - Supervised ML to a real world problem [1]

The comparison between supervised ML methods can be made by making statistical comparisons of the accuracy of trained classifiers in specific datasets [1].
Considering some of the models that the authors present [1] [8] [9] I will evaluate the following algorithms:

- Support Vector Machines (SVM) (Linear SVM)
- Decision Trees
- Naive Bayes Classifier (GaussianNB / MultinomialNB)
- K-Nearest Neighbors
- Logistic Regression
- Random Forest

I will apply the different models on the same test dataset and the same training dataset.
The performance evaluation of the models will be based on the metrics presented in the evaluation metrics section.

Based on the literature and projects that have been implemented in the ML area, I will seek to use several tools and libraries such as: Python [10], NumPy [11], Pandas [12], Scikit-learn [13], Sweetviz [14], Streamlit [15] [16], for data exploration, pre-processing and

model validation. To present the classified complaints the idea is to present a dashboard solution in Tableau [17] integrated with the model. *If there is still an opportunity, I can create an application in Flask and deployment of the model in Haikoru [16] [18].*

Evaluation Metrics

Being a supervised, multi-class problem, **Accuracy** will be one of the metrics I will use to evaluate the performance of the models [19].

Accuracy: Number of correct classifications / Total number of test cases

** it will make sense to use this metric if there is a uniform distribution of samples across existing classes, in order not to induce and error.*

However, the idea is also to evaluate other metrics that can help us to understand the performance of the models:

Classification report for each label/class to evaluate the metrics (**Precision**, **Recall**, **F1-score**, Support)
Recall: True positives / (True positives + False negatives)
Precision: True positives / (True positives + False positives)
F1-Score: will be high when both metrics are high and similar;

I also consider the use of the **Confusion Matrix** and **Cross-validation** important.

- The matrix will show us the number of cases in which our model was right or wrong in each category.
- Cross-validation is one of the most useful techniques to evaluate different combinations of feature selection, dimensionality reduction, and learning algorithms [20].

In the project I will explore these metrics in order to validate the best model for our problem.

Project Design

In the literature there are several frameworks that support us in the process of understanding, designing and implementing ML models in an organized, cyclical and structured way [21] [22] [23].
Text classification is an example of supervised ML task since a labelled dataset containing text documents and their labels is used for train a classifier. An end-to-end text classification pipeline is composed of three main components [22] [24]:

1. *Dataset Preparation*
2. *Feature Engineering*
3. *Model Training*

Considering the literature, I chose to define the following methodology:

1. **Prepare environment**: the first step is the is to prepare the Python + Jupiter Notebooks environment to run the various models;
2. **Dataset preparation:** the second step includes the process of loading dataset and performing basic data exploration and pre-processing. The dataset is then splitted into train and validation sets;
    a. Data collection;
    b. Data exploration and pre-processing [25];
        i. Exploratory data analysis (EDA);
        ii. The dataset will be cleaned, formatted or added the missing values;
        iii. Class distribution vs imbalanced classes analysis;
        iv. Text representation vs vectorize conversion and analysis;
        v. Bag of words analysis;
        vi. …
    c. Then, I'll also define the datasets for training and testing;
3. **Feature engineering:** it's in this step that the raw dataset is transformed into flat features which can be used in a ML model. Here, I'll include the process of creating new features from the existing data;
4. **Model training:** the final step is the model building step in which a ML model is trained on a labelled dataset;
    a. implement the metrics chosen to evaluate and analyze the performance of the models;
    b. implement the selected supervised algorithms and test the performance of each one on the dataset;
    c. compare the performance of all the applied algorithms;
    d. possibility of creating Streamlit application to evaluate the models;
5. **Improve performance:** based on the results, look at different ways to improve the performance of text classifiers. Finding and applying the best hyperparameters [26].
6. **Choosing the best model** for the text classification problem.
7. **Deployment model**: based on the selected model, ensure an application or tool for viewing and interacting complaints vs. classes.

At any time, I can return to an earlier step to refine.

References

1. Iqbal, Muhammad & Yan, Zhu. (2015). SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. International Journal of Soft Computing. 5. 946-952. 10.21917/ijsc.2015.0133.
2. Monkeylearn.com. Accessed 29 Oct. 2020. Available at: https://monkeylearn.com/blog
3. New Features in the CFPB Consumer Complaint Database. Accessed 29 Oct. 2020. Available at: https://www.jdsupra.com/legalnews/new-features-in-the-cfpb-consumer-40931/
4. cfpb.github.io. Accessed 29 Oct. 2020. Available at: https://cfpb.github.io/api/ccdb/
5. paperswithcode.com/task/text-classification Accessed 29 Oct. 2020. Available at: https://paperswithcode.com/task/text-classification
6. kaggle - Consumer Complaint Database. Accessed 29 Oct. 2020. Available at: https://www.kaggle.com/selener/consumer-complaint-database
7. Catalog.data.gov - Consumer Complaint Database. Accessed 29 Oct. 2020. Available at: https://catalog.data.gov/dataset/consumer-complaint-database
8. 2014 intro supervised learning. Accessed 29 Oct. 2020. Available at: https://sebastianraschka.com/Articles/2014_intro_supervised_learning.html
9. IT support ticket classification using machine learning and ml model deployment. Accessed 29 Oct. 2020. Available at: https://medium.com/@karthikkumar_57917/it-support-ticket-classification-using-machine-learning-and-ml-model-deployment-ba694c01e416
10. Python. Accessed 29 Oct. 2020. Available at: https://www.python.org/
11. Numpy. Accessed 29 Oct. 2020. Available at: https://numpy.org/
12. Pandas.pydata. Accessed 29 Oct. 2020. Available at: https://pandas.pydata.org/
13. Scikit-learn. Accessed 29 Oct. 2020. Available at: https://scikit-learn.org/stable/
14. Fast and insightful eda exploratory data analysis using sweetviz 1. Accessed 29 Oct. 2020. Available at: https://towardsdatascience.com/fast-and-insightful-eda-exploratory-data-analysis-using-sweetviz-1-1-3dd0f2389b6c
15. Streamlit. Accessed 29 Oct. 2020. Available at: https://www.streamlit.io/
16. Build-and-deploy-machine-learning-web-app-using-pycaret-and-streamlit. Accessed 29 Oct. 2020. Available at: https://towardsdatascience.com/build-and-deploy-machine-learning-web-app-using-pycaret-and-streamlit-28883a569104
17. Tableau. Accessed 29 Oct. 2020. Available at: https://www.tableau.com/
18. Build and deploy your first machine learning web app. Accessed 29 Oct. 2020. Available at: https://www.kdnuggets.com/2020/05/build-deploy-machine-learning-web-app.html
19. Sklearn Metrics Accuracy Score. Accessed 29 Oct. 2020. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score
20. Predictive modeling, supervised machine learning, and pattern classification. Accessed 29 Oct. 2020. Available at: https://sebastianraschka.com/Articles/2014_intro_supervised_learning.html#machine-learning-and-pattern-classification
21. CRISP-DM, SEMMA e KDD: conheça as melhores técnicas para exploração de dados. Accessed 29 Oct. 2020. Available at: https://paulovasconcellos.com.br/crisp-dm-semma-e-kdd-conhe%C3%A7a-as-melhores-t%C3%A9cnicas-para-explora%C3%A7%C3%A3o-de-dados-560d294547d2
22. A Comprehensive Guide to Understand and Implement Text Classification in Python. Accessed 29 Oct. 2020. Available at: https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python
23. Organizing machine learning projects: project management guidelines. Accessed 29 Oct. 2020. Available at: https://www.jeremyjordan.me/ml-projects-guide/#data
24. Report on Text Classification using CNN, RNN & HAN. Accessed 29 Oct. 2020. Available at: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f
25. Feature Selection with sklearn and Pandas. Accessed 29 Oct. 2020. Available at: https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b
26. Hyperparameter tuning for machine learning models. Accessed 29 Oct. 2020. Available at: https://www.jeremyjordan.me/hyperparameter-tuning