Capstone Project Mauricio Mani

Machine Learning Engineering July 5, 2020

Nanodegree

Ecce Homo: An Application for Automatic Machine Learning Workflows.

Definition

Project Overview

As we have seen, the machine learning workflow is very clear. Therefore, we can automate the tasks. It is not true, we can accurately automate all the tasks, without more advanced techniques, like reinforcement learning. However, in every task we can think of smart techniques to overcome their problems. In this first version I propose automated solving of binary classification tasks.

Problem Statement

This application expects to help in three ways.

- A benchmark for human beings to achieve better outcomes than an automatic workflow (bot).
- Speed up the process of creating data products, to focus on other issues, like getting better data, analysis and explorations, but also things like deployment of the products.
- As stated by Dan Lubanga, making easier to create machine learning products would attract more people to the field.

The first bullet aims to motivate enthusiasts and professionals to achieve better metrics in their model, having competition, and especially non-human competition. The second bullet, and the one I have found the most annoying for some peers in the industry, is the time they spent doing the same tasks, again and again. And finally, the third one, I expect this product to help more people develop easier machine learning products, hence being captivated and ready to go deeper on their learning.

Inputs and benchmark

Test de application on Kaggle binary classification tasks. The famous titanic dataset (https://www.kaggle.com/c/titanic), churn prediction of customers

(https://www.kaggle.com/c/ic20182) and fraud detection challenge (https://www.kaggle.com/c/competetion1/overview).

As stated on the inputs section, the benchmark models, would be the Kaggle competitor's score. I do not look forward best metric on test, but a fast implementation of machine learning with simple lines of code. Not having to repeat myself every time I do start a project and have control of the process. E.g. I do want to handle missing variables, but that is it.

Solution

Through well-defined pipelines solve the former problems without creating a specific workflow for none of them. Being able to call a library that helps me perform all the machine learning workflow fast and easy but keeping control.

Evaluation

The application would be evaluated using area under the ROC curve on test data set. Because it is the one used in Kaggle for competition. However, a variety of metrics would be printed, and user would be able to specify importance either on recall or precision. For example on the churn dataset, recall would be important (specifically if we can advocate a lot of resources to those found as possible churn), meanwhile, on the fraud you might want high precision, to bring smooth customer experience. I am looking forward to seeing on what quartile of all competitors Ecce Homo application is. And that would be how we measure how efficient is the application.

Project Design

A smooth and replicable machine learning workflow, that allows user interaction to define all the below tasks or none of them.

- Print simple descriptive statistics and plot data. So, the user can understand if she needs more information to solve the problem or specific feature engineering or variables creation.
- Input data on different ways and notify of missing data: mean, median, zero, random, missing indicator, K nearest neighbors[1], clustering.
- Categorical encoding
- Variable transformation: Follow normal distributions.
- Outlier handling: Boxplot outliers[2], isolation forest[3], DBSCAN[4], and discretization.

---

[1] Zhang, "Nearest neighbor selection for iteratively kNN imputation", 2012.
[2] Tukey, J. "Exploratory Data Analysis", 1977. https://archive.org/details/exploratorydataa00tuke_0
[3] Lui, Ting, Zhou, "Isolation Forest", https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf?q=isolation-forest
[4] Easter, Kriegel, Sander, Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", 1996. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

- Feature scaling: Minmax and standardization.
- Balancing datasets: SMOTE[5], SMOTEEN, random over and under sampling.
- Modeling with hyperparameter tuning using Bayesian optimization[6]. On basic machine learning algorithms like logistic regression with and without regularization, random forest and support vector machine.
- Print evaluation metrics on test sets.

Conclusion

I proposed this application, and not tackling down a specific problem of Kaggle or the ones proposed, because I think this could help me improve my OOP, modularity, unit tests, be helpful in my daily basis and really work on a machine learning engineer task, instead of a data science one (I know they are quite similar but not the same).

[5] Chawla, Bowyer, Hall, Kegelmayer, "SMOTE: Synthetic Minority Over-sampling Technique", (2002). https://arxiv.org/pdf/1106.1813.pdf

[6] Wu et al. "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization". 2019.