LELEC2870 Machine Learning Project: Predicting shares of articles

Friday 20th November, 2020

Introduction

Machine learning methods can be used to solve many practical problems in a wide range of applications such as weather forecast, customer clustering, medical diagnostics, spam blocking, financial time series prediction or signal de-noising, ... In this project, you will apply machine learning to predict the number of shares of an online article based on its components.

Data

The data you'll be working with was garnered from Mashable articles [1] and represents different features extracted from the article's text such as adherence to keywords, sentiment analysis etc. You'll find the data on the course website in three csv files called X1.csv, Y1.csv and X2.csv. The first one is the labeled dataset and the second will be used to make your prediction. Paste these files in your working directory. The data can be loaded in your workspace by running the following commands:

```
import pandas as pd

# Use pandas to load into a DataFrame
#     Y1.csv doesn't have a header so
#     add one when loading the file
X1 = pd.read_csv("X1.csv")
Y1 = pd.read_csv("Y1.csv", header=None, names=['shares'])

# If you prefer to work with numpy arrays:
X1 = X1.values
```

Each row in the data corresponds to an observation (a measurement). Each column represents one of the features listed here :

Feature	**Type (#)**
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	**Type (#)**
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

Target	**Type (#)**
Number of article Mashable shares	number (1)

Figure 1: Features split up by category [1]

Instructions

The project is realized by groups of two or alone. It is composed of different aspects as specified below.

Model

You will build regression models that predict the number of shares of an online article. You can use any of the methods seen during the lectures. **We expect you to implement linear or lasso regression *as a baseline*, KNN[1], an MLP and at least one other non-linear method** (you can chose one outside those seen during the lectures).
Feature selection and model selection shall also be part of your work. Once again you're allowed to use any tools available (e.g. statistical tests seen during other courses). Pay attention to the fact that the model selection can require a lot of computation time. You are advised to explore the metaparameters (e.g. number of neighbors for KNN, learning rate for MLP's etc.) space according to the time available.

Prediction

Once your model is properly selected and validated, **you are asked to produce predictions Y2 on the data X2 for which we have kept secret the corresponding targets**. This prediction vector should be uploaded on Moodle in a csv file named "Y2.csv" that contains **one**

[1] K-Nearest Neighbour : Regression model with metaparameter K that predicts the output of a sample as the mean of output of the K nearest neighbours in the features space.

line per prediction and no header for a total of 19822 numeric values one under the other. Your prediction quality criterion should be the average binary f1-score at different thresholds:

```
import sklearn
def score_f1(y_true, y_pred, th):
    return sklearn.metrics.f1_score(y_true>th, y_pred>th)

def score_regression(y_true, y_pred):
    scores = [
        score_f1(y_true, y_pred, th=th) for th in [500, 1400, 5000, 10000]
    ]
    return np.mean(scores)
```

Although most algorithms will optimize the RMSE we found this subdivision of thresholds to be more representative of a real world use case. With these thresholds, we can divide the articles in 5 categories: flop (th < 500), mild success (th $\in [500, 1400[$), success (th $\in [1400, 5000[$), great success (th $\in [5000, 10000[$), viral hit (th > 10000). In addition, **you should provide in your report, an estimate of the score** that you expect on your predictions. Don't forget that this is a score you should try to **maximize**.

Report

You should produce a report documenting your technical choices and experimental results. Note that:

- We do not need a course on the methods you use. We are more interested in what you did and why.
- Be straight and to the point!
- Try to illustrate your results with *clear* graphics (with *legends* and *labeled axes*) and comment them.
- Be critical about what you observe and try to give a possible justification of the obtained results
- Summarize your *main* results and observations as well as your predicted score in a conclusion.
- **A strict maximum of 7 pages** (font of size 11 or larger) will be observed. Annexes might be included in the digital version that you'll submit on Moodle, but shouldn't be resorted to unless something really interesting was found.
- All your figures and computation need to be reproducible by us running your implementation code on the provided data.

Programming languages

The programming language you need to use is **Python**. You can use any toolbox/library available on-line. In particular, we strongly recommend using the `scikit-learn` library as it provides many useful implementations of standard machine learning approaches. For the MLP we recommend you to use the one integrated in sklearn or `pyTorch`. `skorch` is a python package that links `pyTorch` with the `sklearn` syntax, some of you might find it handy, but don't hesitate to come to us with questions if you encounter bugs with these packages.

Agenda (Important)

- As soon as possible: Register your group (maximum two people) on the course website (starting Friday 20/11).
- Thursday, December 3 at 16:15: We will hold an online Q/A session
- Sunday, December 20, 23h55 (to avoid any confusion on Moodle)
  - submit your work as 3 separate files:
    - Your report (.pdf)
    - A csv file called "Y2.csv" **no header line: one line per prediction values**.
    - A compressed folder (.zip) containing all scripts you wrote for the project. The code should be commented well enough and installation instructions for non-standard packages you used should be provided.

Evaluation Criteria

- Respecting the instructions and deadlines
- Quality of the report
- Your machine learning approach (model choices and validation)
- Reproducibility of your results
- Consistency between the report, your implementation and your predictions

Please note that the performance of your models is not critical in the evaluation. New this year, the oral part of the project will be replaced by a question on the exam. Your report + your points on this question will encompass 10 of the 20 points of your final mark.

Tips

Here is a list of advice for the project.

Before any analysis:

- Visualizing the data is always useful.
- Evaluate your model correctly
- Normalize or Standardize your data if necessary.
- For some algorithms redundancy between features may negatively impact their performance. Do you know which ones? How can you avoid this?
- Some outliers might be excluded from the learning set (if you decide to remove some observations, explain why you removed them).
- Categorical data like weekdays has already been split in a one-hot encoding form. What does that mean for your algorithms? Is there perhaps a better way of encoding this (think cyclically)?

You can discuss the project with other students, in fact, it is a great idea! You could compare your results to those obtained by other groups, but remember that it is not allowed to copy what others did... Every piece of code taken from the internet or other ideas you found while researching *have to be cited*!

We will be happy to answer your questions during the Q/A sessions or on appointment. Good luck !

References

[1] K. Fernandes, P. Vinagre and P. Cortez. (2015) A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.