Solving financial and banking problems with machine learning

CISSE Cheikh , CHIBANE Mohamed , FALL Cheikh Ahmed Tidiane , SKIREDJ Selma , YACOUBOU
SOKA Hack
Université de Montréal

INTRODUCTION

Classification algorithms are used in a multitude of ways and simplify a problem into providing us binary or multiclass solutions. We would like to see how classifiers deal with two very different financial problems. The first one applies to trading stocks daily using technical and economic indicators as tools for determining order placement. The second financial problem we seek answers to is, considering the characteristics of a client base and records of previous marketing campaigns, whether a new client be likely to subscribe to a term deposit at a banking institution.

DATASETS

I. Trading Data

- 12 features (2 technical, 10 macroeconomic)
- 4725 days of trading
- 3-class output denoting whether to buy, sell, do nothing when the market opens
- The particularity of this dataset is that it contains time-series data.

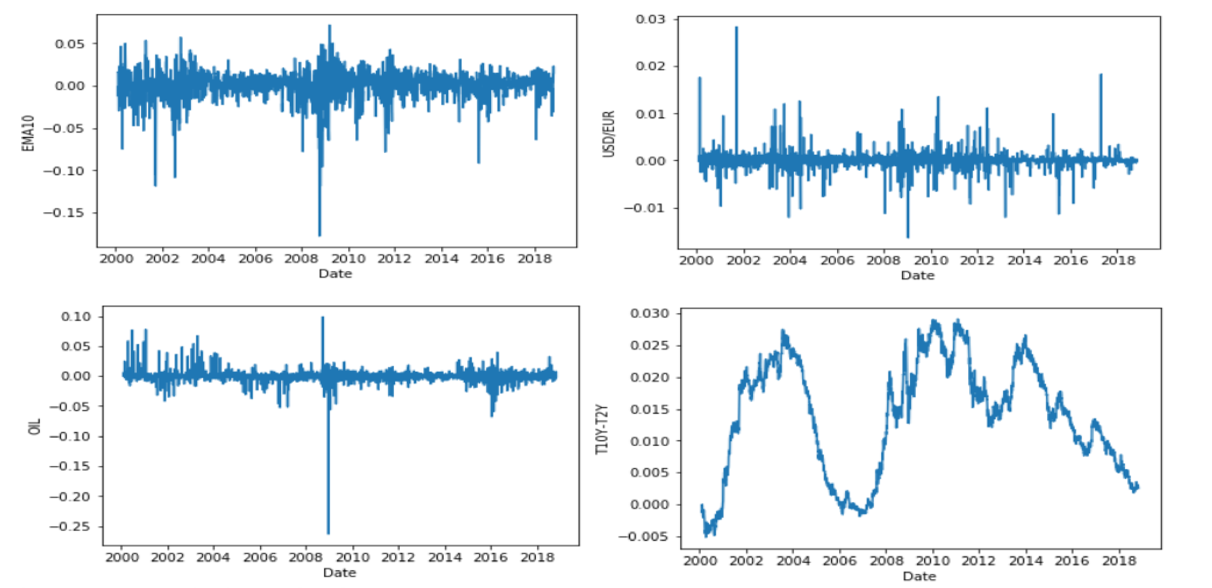

Figure 1: *Evolution of 4 features*

II. Bank telemarketing data

- 20 features (client characteristics)
- 45211 examples ordered by date
- 2-class output labeled “yes” or “no” to refer to the success of failure of the client contact.
- The particularity of this dataset remains in its unbalanced output-classes

Figure 2: *Unbalanced distribution of the output classes*

CLASSIFICATION ALGORITHMS

- Simple algorithms : logistic regression, SVM
- Ensemble method : Random Forest
- Deep learning algorithm : Neural Network

METHODOLOGY

I. Data Preprocessing

I.1) Trading Data

- Daily raw market data (Jan 2000 – Nov 2018) → Returns
- Creation of the output : Introduction of a commission rate of 0.5% of the trade value to determine what constitutes the decision boundaries and to make the exercise more realistic.

I.2) Banking Data

- No missing values
- One-hot encoding for categorical features
- Deleting variables which are very correlated

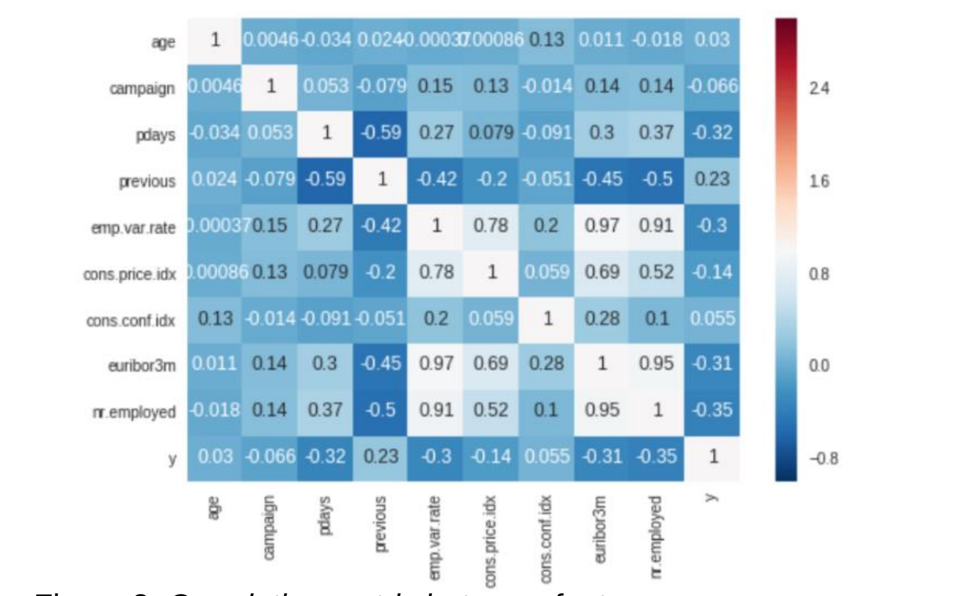

Figure 3: *Correlation matrix between features*

II. Hyper-parameters searching

- Logistic regression: SVM, Random Forest : grid search
- Neural Networks : Manual tuning

III. Learning strategies

III.1) Trading Data

3 splitting strategies

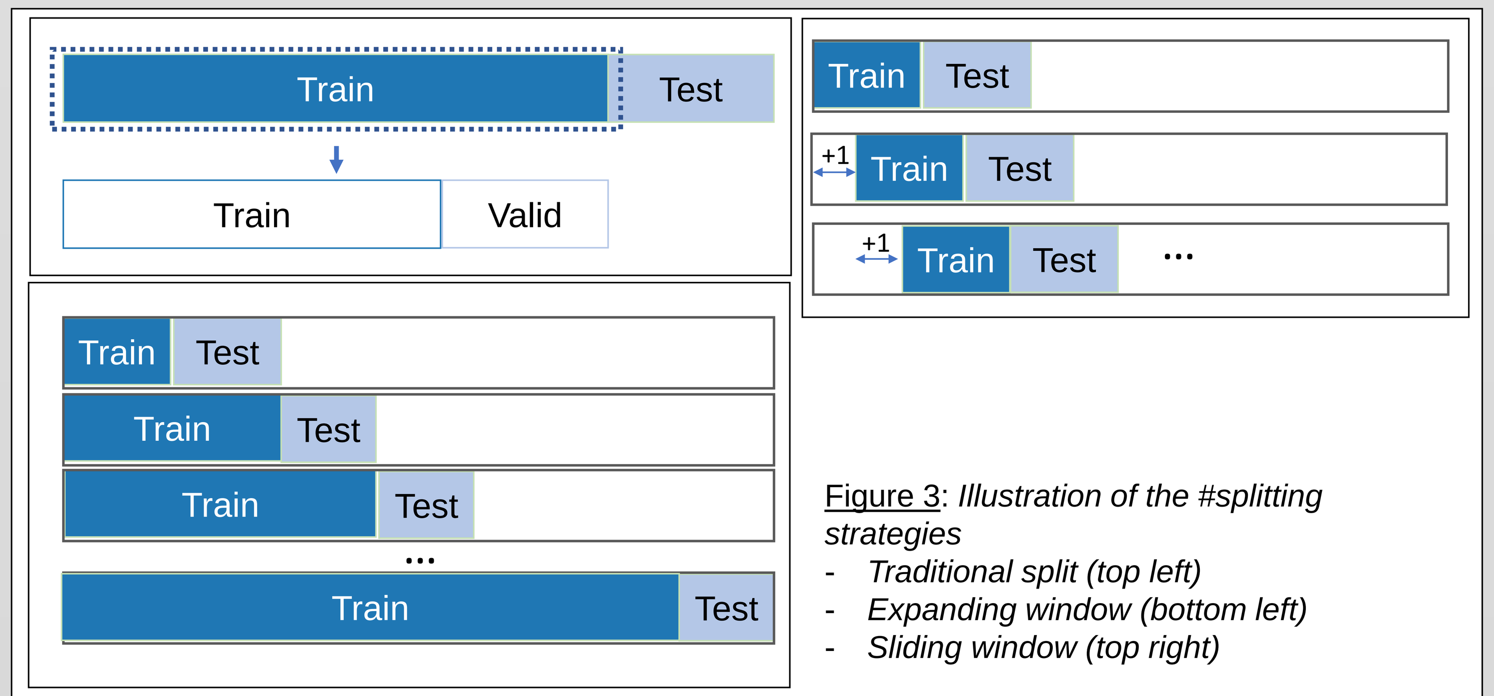

Figure 3: *Illustration of the #splitting strategies*
- *Traditional split (top left)*
- *Expanding window (bottom left)*
- *Sliding window (top right)*

III.2) Banking Data

- The dataset is separated into a training set (70%) and a test set (30%)
- Criteria used to evaluate the model : accuracy and confusion matrix
- 2 strategies comparing the utility of compensating the under-represented class in the dataset by allocating bigger weights

RESULTS

I. Trading Data

Table: Test accuracy score for the trading data

	Strategy 1 (Traditional split)	Strategy 2 (Expanding window)	Strategy 3 (Sliding window)
Logistic Regression	71.6%	44.5%	60.4%
SVM	71.5%	58.7%	60.5%
Random Forest	71.5%	57.4%	60.5%
Neural Network	71.3%	51.2%	x

Table 1: *Comparison of accuracies by algorithm/ splitting strategy*

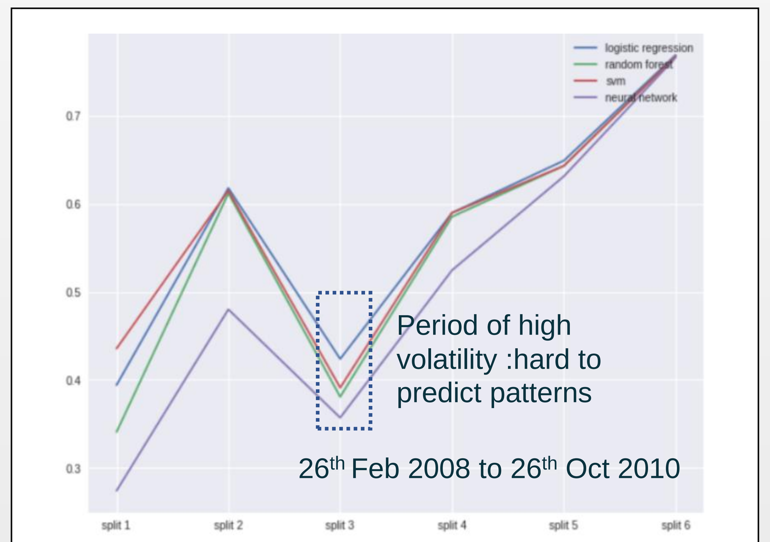

Figure 4: *Evolution of accuracies for strategy 2*

Strategy 1 holds the highest accuracy score and is the fastest to compute.
Strategy 2 gives more robust estimations but requires multiple models to be trained and evaluated.
Strategy 3 is a good compromise between the first two as it yields to good accuracy and is more robust as multiple models are trained. Plus, it is that it is easier to interpret since the model is constantly evolving over time and concentrates the learning phase only on recent data.

II. Banking Data

Very good performance globally.
The random forest is easily interpretable, accurate and detects the presence of the minority class in the test set.
Logistic regression shows outstanding performance compared to other algorithms with higher capacity (neural network).

	Strategy 1: No compensation for the under-represented class in the dataset applied	Strategy 2: Compensate the under-representation of the minority class in the dataset for the algorithms using *class_weight* parameter
SVM	89,30%	88,75%
Neural Network	89,91%	89,78%
Logistic regression	90,06%	89,81%
Random forest	90,12%	89,38%

Table 2: *Comparison of accuracies by algorithm/weighting strategy*

CONCLUSION AND FUTURE IDEAS

Generally speaking, simple algorithms do a very good job on binary and 3-class classification. Ensemble methods and deep learning algorithms are not necessary here. To go further, interesting ideas could be developed; such as using LSTM’s to predict the daily price change by taking into account long term dependencies in the data; and programming a robot to make the calls and learn instantly during the live interaction with the client what formulation or approach could lead with a high probability to a favorable outcome.This type of project is a field of deep learning techniques, specially built upon NLP algorithms and is still developing.