EXAMEN: Trade a single stock with Machine Learning

Objective of the project: Develop a daily trading strategy using a machine learning classifier.
Pick one of the underlying assets of your choice in the list I will send. You will develop a strategy to **either go long on the stock**, **or to go long/short.** (We already covered the case of being short in class).
You want to build a strategy that has a good timing for buying a stock, (signal will be flat or long) or a strategy that has good timing for buying AND short selling a stock (signal will be short or long).
Build a trading strategy with a machine learning classifier of your choice, make sure to follow the steps we have seen in class to build a robust machine learning framework.
 You will not be judged on how "profitable" the strategy is, but rather on how you conducted the study.

- **Create features for your model (input)**

Use closing price from the stock and transformation to create meaningful features for your model.
Use TA-LIB library to build new features (ex: RSI, MACD,Z-score, etc…)

- **Create labeling for your model (output)**

Hint: Calculate forward returns of your stock on a horizon of X days (5days is a good start). Draw a distribution of the forward returns, split the distribution into bins, and affect each weekly returns to a class. One example: if you have weekly returns, split the distribution with the median, if median is 0.5%, any weekly return above will be assigned 1, 0 for the rest. You can also take 0 as arbitrary threshold.

- **Show me how you cleaned data.**
- **What metrics should you use as objective function?**

Hint: Careful about the ML logic, predicting a market crash is different than stock picking. When you try to predict a market crash, you want to make sure to identify as many "market crash" as possible (=recall), when you do stock-picking, you want to make sure that the stock you pick performs well. (Different logic). Which metrics you should optimize for?

- **Reduce the number of features. Apply some feature selection method**. (This step can also be conducted during the hyperparameter tuning optimization process)

Bonus point : if you use pipeline method from sklearn during the optimization process to reduce the number of features (see: SelectKBest for example)
This step is not necessary if your work with few features.

- **Select the best model using a cross-validation schema and tune your hyperparameters with the solution of your choice. (GridSearchCV, Optuna, etc….)**
- **Return metrics of the final model on training, cross-validation, and test sets and compare the result to a benchmark model.**
- **Back-testing of your trading strategy out-of-sample using daily signal.**

Bonus point : Integrate slippage and trading fees (example: 0.5% per trade)

- **Plot results (strategy + corresponding stocks), explain your results, be critical: what is good and what is bad. Which area of your model could be improved? What could you work on to make it better?**

What you need to deliver:
- Create a zip file with the whole project.
- *Bonus point: Create a clean poetry set-up, or pip set-up.*
- **Minimum requirement**: a jupyter-notebook with all the code (if you need some dependencies, have at least a requirements.txt with libraries in it).
- **Mandatory**: A PDF file that explains every steps above but **NO CODE** (only screenshot of results, visualization) I want to understand why you picked a particular metrics, why you decide to use this algorithm, why you got rid of X features, why your results are or not satisfying, etc….
  Remember : it is not about the results, but how you got there in the first place.
- **PDF: approx. 5 pages, concise, and clear**.
- **Group of 3-4people.**

Adress : lamblindustin@gmail.com, subject: UNICE – Exam – <family_name_participants>

Final Comment:
I am aware that the deadline is relatively tight, do not stress about having good results, or making a fancy set-up of the project, focus on answering questions with explanations and why you solve this question in this particular manner. Remember that you already have most of the code available, it is more about adapting it to your business problem.
Focus on what we have seen in class, and what are the most relevant points when building a machine learning trading system.
If you have issue with some library (SHAP, optuna), you can use conventional tool (GridSearch and other methods already implemented through sklearn).