CQF Exam Three (ML Assignment)

January 2020 Cohort

A. Predicting Price Direction [48 marks]

Unpredictability of short-term asset returns is a subject of asset pricing research: efficient markets produce near-Normal daily returns with low correlation to past values. Popular time series models – autoregression on lagged returns – are of little use. However the progress is possible, in this assignment you will make DIRECTION PREDICTION only using Classifiers **listed in A.1 - A.3**.

Predict the sign of next daily move. It is welcome to modify the task to predict direction for the longer periods, a 5-day move. Certain classifiers are more suitable to that but beware of pronounced positive autocorrelation in 5D/10D/21D returns.

The assignment limits the task to binomial prediction in asset price movement: positive or negative move $-1, 1$. For some classifiers, particularly if using bagging/boosting, re-label as $0, 1$.

Start with lagged log-returns $r_{t-1}, r_{t-2}, \ldots$ as your features. Use ADDITIONAL simple variations around price P_t from Table 1. More complex indicators (eg, RSI, Stochastic K, MACD, CCI, Acc/Distrib) are beyond the scope of the assignment.

Feature	Formula	Description
OHLC	-	Open, High, Low, Close price
Return Sign	$\text{sign}[\ln \frac{P_t}{P_{t-1}}]$	predict from sign only
Momentum	$P_t - P_{t-k}$	daily or longer period $k > 1$
Moving Average	$\text{MA}_i = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i}$	helps to identify the trend
Exponential MA	$\text{EMA}_t = \text{EMA}_{t-1} + \alpha\,[P_t - \text{EMA}_{t-1}]$	recursive scheme
Sample Std Dev	$\sigma_{t-1} = \sqrt{\frac{1}{n-1}\sum(r_i - \mu)^2}$	to predict negative moves

Table 1: Features to use in prediction in addition/in place of past returns

Study Design:

- Choose 2 equities with a base of comparison (eg, same industry), or 2 market indexes, or 2 Fama-French factors. FF factors are good candidate series to try prediction of monthly returns.
- Avoid using GOOG/AAPL/MSFT and alike popular large cap stocks. If using a market index be prepared to reduced accuracy.
- Use 7 or fewer lagged values of return. Sample but not limiting set of features *wrt* t: Ret_{t-1}, Ret_{t-2}, $\text{Ret}_{5D,t-1}$, Mom 1D $P_{t-1} - P_{t-2}$, Mom 5D $P_{t-1} - P_{t-6}$, SMA 5D, EMA 7D, Std Dev 21D.

Classifier A.1 Logistic Classifier and Bayesian Classifier

a) Make sure to implement penalised versions of logistic regression and discuss impact on coefficients. Apply and discuss the difference between L1 and L2 cost functions, the impact made on regression coefficients (comparison table recommended). b) Demonstrate the use of *sklearn.model_selection* for *reshuffled samples* and *k-fold crossvalidation*.

Classifier A.2 Support Vector Machines

a) Consider soft vs. hard margin, present in mathematical notation and consider impact on your 2D relationships. b) Specifically consider Momentum Feature vs Return $t-1$ and provide 2D visualisation (up/down points in different colour). While support vectors are difficult to present, use *SVM_SVC.supportvectors* and prepare interpretable visualisations. c) No need to vary type of kernel.

Classifier A.3 K-Nearest Neighbours

a) Given KNN's dependence on distance computation scale the features, eg, *StandardScaler* from *sklearn.preprocessing*. b) Report on sensible values for *n_neighbors* hyperparameter and provide comparison of *metric*, particularly Manhattan vs Euclidean vs Mahalanobis – see *DistanceMetric* class. c) Plot decision boundaries, if sensible and describe in brief what is 'lazy' about classification with KNN.

B. Prediction Quality and Bias (each chosen classifier) [52 marks]

Work on these tasks can be appended to each classifier use case but it is necessary to make comparisons across all Classifers as well.

Task B.1 Investigate the prediction quality using **confusion matrix** (precision/recall statistics) and **area under ROC curve** – these are possible for all classifiers if prediction is binomial. Particularly check the quality of predicting the down movements (negative sign of return).

Task B.2 Attempt feature scoring or selection, for instance with *sklearn.model-selection.GridSearchCV*. Briefly compare results between all-feature/selected features models. Alternatively, introduce bagging and discuss prediction implications and accuracy. Make recommendation on how to reduce **misclassified negative returns**.

Task B.3 Utilise transition probabilities *predict_proba()* from two most sensible classifications (specific set of Classifier+features) in P&L Backtesting. First, provide **scatter plots for transition probabilities** of up and down moves *separately*, color-code for correct/incorrect predicted values.

Second. Assume daily betting on price direction. Use the realised return wth the PREDICTED sign to compute profit (loss) for end of the day. However, your allocation to asset is not 100% but Kelly optimal fraction to bet $p-(1-p)=2p-1$. Therefore, you remain in cash for $1-(2p-1)=2(1-p)$ percentage which reduces your gain (and loss). *Example:* with allocation 75%, asset went up 5%, position gain is $0.75\times 0.05+0.25\times 0=3.75\%$ of the total 100%.

You can vary the scheme to bet only when p is above a threshold 52-55%, and for other days to remain in cash. For information only your series of bets has a steady-state distribution for $y=\{1,0\}$ as:

$$pdf(y;p)=p^y(1-p)^{1-y}=\exp\left[y\log\left(\frac{p}{1-p}\right)+\log(1-p)\right].$$

Task B.4 mathematical Find the way to present the *pdf* of Normal distribution in the form below, state explicitly what are $a(x), b(\mu), c(\mu), d(x)$

$$f(x;\mu) \quad = \quad e^{a(x)b(\mu)+c(\mu)+d(x)}.$$

pdf expression not provided, you have to identify places of x, μ in it.
END OF EXAM

Instructions

Work on ALL tasks in the format required. Recite maths for each chosen Classifier. Code must be submitted and be producing the computational output.

Submit ONE .pdf report file and ONE .zip file with data and code. Please name files to start with your *LASTNAME*.

- Implementation is best done in Python using *sklearn*. *price_direction_prediction.ipynb* provided as a template to start the work.
- It is acceptable to implement classification in R/Matlab, but tutor's support might be limited. Matlab use should not devolve to exploration with Classification Learner App only.
- It is possible to have a limited implementation in Excel (eg, logistic regression), however that risks to be below passing mark (60%) because other Classifiers/prediction quality analysis missing.

Report Content and Analytical Quality:

- If printing out Python Notebook as your report – please ensure it comes across **as an analytical report** with a) headers to separate sections, b) clarity which sections address Questions A.1 - A.3 and B.1 - B.3, and c) avoid large output (eg, show the head/tail).
- It is not expected that you will have particularly high accuracy and recall in all classes in predicting daily direction from past returns.

Data Access: refer to Webex on Equities Data and *FP_EquitiesData_Pandas.ipynb*.

Portal and upload questions to Orinta.Juknaite@fitchlearning.com. Clarifying only questions to Richard.Diamond@fitchlearning.com.