Instructions for Trees Homework

Please fill in the missing code and if possible run the following 3 ipynb files:

1. Regression Tree\_mod\_Incomplete.ipynb
2. Regarding Classification Tree Model\_mod\_Incomplete.ipynb
3. Class Weights in Decision Trees\_mod\_Incomplete.ipynb

Please answer the following questions in the space provided:

1. Regarding Regression Tree\_mod\_Incomplete.ipynb, state the trading rule in Python for extreme leaf trading:

#data['predicted\_signal'] = np.where(((data.col > A) & (data.col > B)), 1, 0) (A, B should be substituted)

data['predicted\_signal'] = np.where(((data.ret40 > -0.056) & (data.std5 > 0.015)), 1, 0)

2. Regarding Regression Tree\_mod\_Incomplete.ipynb, state the trading rule in Python for full tree trading:

#data['predicted\_signal'] = np.where(CONDITION, 1,-1) (CONDITION should be substituted)

data['predicted\_signal'] = np.where(dtr.predict(X) > 0, 1,-1)

3. Regarding Classification Tree Model\_mod\_Incomplete.ipynb, state the train\_test\_split function call in Python:

#X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, NEEDS PARAMETER) (NEEDS PARAMETER should be substituted)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, stratify=y)

4. Regarding Classification Tree Model\_mod\_Incomplete.ipynb, state trading rule in English (extreme leaf):

Rule: short the stock if both ADX is less or equal to 32.446 and RSI is less than equal to34.542.

5. Regarding Class Weights in Decision Trees\_mod\_Incomplete.ipynb, state the DecisionTreeClassifier function call in Python:

#clf = DecisionTreeClassifier(criterion = 'gini', max\_depth = 3, min\_samples\_leaf = 5, NEEDS PARAMETER) (NEEDS PARAMETER should be substituted)

clf = DecisionTreeClassifier(criterion = 'gini', max\_depth = 3, min\_samples\_leaf = 5,

class\_weight = 'balanced')

6. The way the distribution of target classes is balanced out or corrected in this homework is most compatible with a time series data model or with cross sectional data model (or equally compatible with both)?

Both the time series data model and cross sectional data model are equally compatible. This is because the data is not shuffled therefore the time order of the data is retained, and the added weights just give the less frequent class more weights to balance it out; therefore, it is compatible with both.