Question: You are given the following training examples. Each example has only one attribute, and the classification into positive / negative

Index	*X*	*Label*
1	*1.0*	*Positive*
2	*2.0*	*Negative*
3	*4.0*	*Positive*
4	*5.0*	*Positive*
5	*6.0*	*Negative*
6	*7.0*	*Negative*

Your main task is to evaluate the following algorithm that use a set *S* of training examples to classify the example with attribute value of *x.*

Algorithm:

Let S_p, S_n be the sets of positive and negative examples in S.

If S_p is empty classify x as negative. If S_n is empty classify x as positive.

Otherwise, compute u_p, the mean of the x values in S_p, and u_n, the mean of the x values in S_n.

If x value is closer to u_p than it is to u_n then classify x as positive. Otherwise classify x as negative.

Example: Using all the training examples above we have: $u_p = 3.33$, $u_n = 5$. Therefore, an example with $x = 2.5$ is classified as positive.

(a) Use leave-one-out cross validation to estimate the errors of Algorithm above [3 Points]
(b) Use 3 Fold CV to estimate the errors of Algorithm above. [3 Points]

Solution (a) Error = 0.5;
Example a: test, 1.0 +ve; Sp={4,5}, Sn={2,6,7}, thus mean(p) = 4.5; mean(n) = 5, Thus 1 is near to mean(p). Correct
Example b: test, 2.0 -ve; Sp={1,4,5}, Sn={6,7}, thus mean(p) = 3.33; mean(n) = 6.5, Thus 2 is near to mean(p). Incorrect
Example c: test, 4.0 +ve; Sp={1, 5}, Sn={2, 6,7}, thus mean(p) = 3; mean(n) = 5, Thus 4 is near to both. Tie. Incorrect
Example d: test, 5.0 +ve; Sp={1, 4}, Sn={2, 6,7}, thus mean(p) = 2.5; mean(n) = 5, Thus 5 is near to mean(n). Incorrect
Example e: test, 6.0 -ve; Sp={1, 4, 5}, Sn={2, 7}, thus mean(p) = 3.33; mean(n) = 4.5, Thus 6 is near to mean(n). Correct

Example f: test, 7.0 -ve; Sp={1, 4, 5}, Sn={2, 6}, thus mean(p) = 3.33; mean(n) = 4, Thus 7 is near to mean(n). Correct

Index	*X*	*Label*	*Predicted*	*Mean*
A	*1.0*	*Positive*	*Positive*	*4.5,5*
B	*2.0*	*Negative*	*Positive*	*3.33, 6.5*
C	*4.0*	*Positive*	*Negative / Positive*	*3,5*
D	*5.0*	*Positive*	*Negative*	*2.5,5*
E	*6.0*	*Negative*	*Negative*	*3.33,4.5*
F	*7.0*	*Negative*	*Negative*	*3.33,4*

Solution (b)

Index	*X*	*Label*
A	*1.0*	*Positive*
B	*2.0*	*Negative*
C	*4.0*	*Positive*
D	*5.0*	*Positive*
E	*6.0*	*Negative*
F	*7.0*	*Negative*

b. Use 3 Fold CV to estimate the errors of Algorithm above.) [3 Points]

Ist Fold; train.index = {A,B,C,D} ; test.index = {E,F} => train.sample ={1,2,4,5}; test.sample = {6,7}
SP = {1,4,5}, Sn = {2}, mean(p) = 3.33; mean(n) = 2. Both {6,7} i.e. E, F near to positives thus both examples incorrect

2st Fold; train.index = {A, B, E, F} ; test.index = {C,D} => train.sample ={1,2,6,7}; test.sample = {4,5}
SP = {1}, Sn = {2,6,7}, mean(p) = 1; mean(n) = 3. 4 and 5 near to 3 thus both incorrect

3rd Fold; train.index = {C, D, E, F} ; test.index = {A,B} => train.sample ={4, 5, 6, 7}; test.sample = {1,2}
SP = {4,5}, Sn = {6,7} = mean(p) = 4.5, mean(n) = 6.5. One correct and one incorrect

Total Error = 5/6

Index	*X*	*Label*	*Predicted*	*Mean*
A	*1.0*	*Positive*	*Positive*	*4.5, 6.5*
B	*2.0*	*Negative*	*Positive*	*4.5, 6.5*
C	*4.0*	*Positive*	*Negative*	*1.0,3.0*
D	*5.0*	*Positive*	*Negative*	*1.0, 3.0*
E	*6.0*	*Negative*	*Positive*	*3.33, 2*
F	*7.0*	*Negative*	*Positive*	*3.33,2*