Name: Qixiao Zhu

Part 1
Exercise 1

a)
1-itemset:
{M}: 0.6
{O}: 0.6
{N}: does not meet min-sup
{K}: 1
{E}: 0.8
{Y}: 0.6
{D}: does not meet min-sup
{A}: does not meet min-sup
{U}: does not meet min-sup
{C}: does not meet min-sup
{I}: does not meet min-sup

2-itemset:
{M, O}: does not meet min-sup
{M, N}: apriori
{M, K}: 0.6
{M, E}: does not meet min-sup
{M, Y}: does not meet min-sup
{M, D}: apriori
{M, A}: apriori
{M, U}: apriori
{M, C}: apriori
{M, I}: apriori
{O, N}: apriori
{O, K}: 0.6
{O, E}: 0.6
{O, Y}: does not meet min-sup
{O, D}: apriori
{O, A}: apriori
{O, U}: apriori
{O, C}: apriori
{O, I}: apriori
{K, E}: 0.8
{K, Y}: 0.6
{K, D}: apriori
{K, A}: apriori

{K, U}: apriori
{K, C}: apriori
{K, I}: apriori
{E, Y}: does not meet min-sup
{E, D}: apriori
{E, A}: apriori
{E, U}: apriori
{E, C}: apriori
{E, I}: apriori
{Y, D}: apriori
{Y, A}: apriori
{Y, U}: apriori
{Y, C}: apriori
{Y, I}: apriori

3-itemset:
After careful check to make sure eliminated subsets are not included, I found one. I think it is a little bit tedious to list all the combinations and mark the reasons that they are eliminated.
{O, K, E}: 0.6

All frequent itesmets:
{M}: 0.6, {O}: 0.6, {K}: 1, {E}: 0.8, {Y}: 0.6, {M, K}: 0.6, {O, K}: 0.6, {O, E}: 0.6, {K, E}: 0.8, {K, Y}: 0.6, {O, K, E}: 0.6

b)
An itemset X is a closed itemset if there exists no proper superset of X with the same support as X
{K}, {M, K}, {K, E}, {K, Y}, {O, K, E}

c)
An itemset X is a maximal frequent itemset (or max-itemset) if X is frequent and there exists no proper superset of X that is also frequent
{M, K}, {K, Y}, {O, K, E}

d)
{M} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.6) = 1.67
{K} -> {M} confidence = 0.6 / 1 = 0.6; lift = 1.67; eliminated
{O} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 1) = 1
{K} -> {O} confidence = 0.6 / 1 = 0.6; lift = 1; eliminated
{O} -> {E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{E} -> {O} confidence = 0.6 / 0.8 = 0.75; lift = 1.25; eliminated
{K} -> {E} confidence = 0.8 / 1 = 0.8; lift = 0.8 / (1 * 0.8) = 1
{E} -> {K} confidence = 0.8 / 0.8 = 1; lift = 1
{K} -> {Y} confidence = 0.6 / 1 = 0.6; lift = 0.6 / (1 * 0.6) = 1; eliminated

{Y} -> {K} confidence = 0.6 / 0.6 = 1; lift = 1
{O} -> {K, E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{O, K} -> {E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{O, E} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 1) = 1
{K} -> {O, E} confidence = 0.6 / 1 = 0.6; lift = 1; eliminate
{E} -> {O, K} confidence = 0.6 / 0.8 = 0.75; lift = 1.25; eliminate
{K, E} -> {O} confidence = 0.6 / 0.8 = 0.75 ; lift = 1.25; eliminate

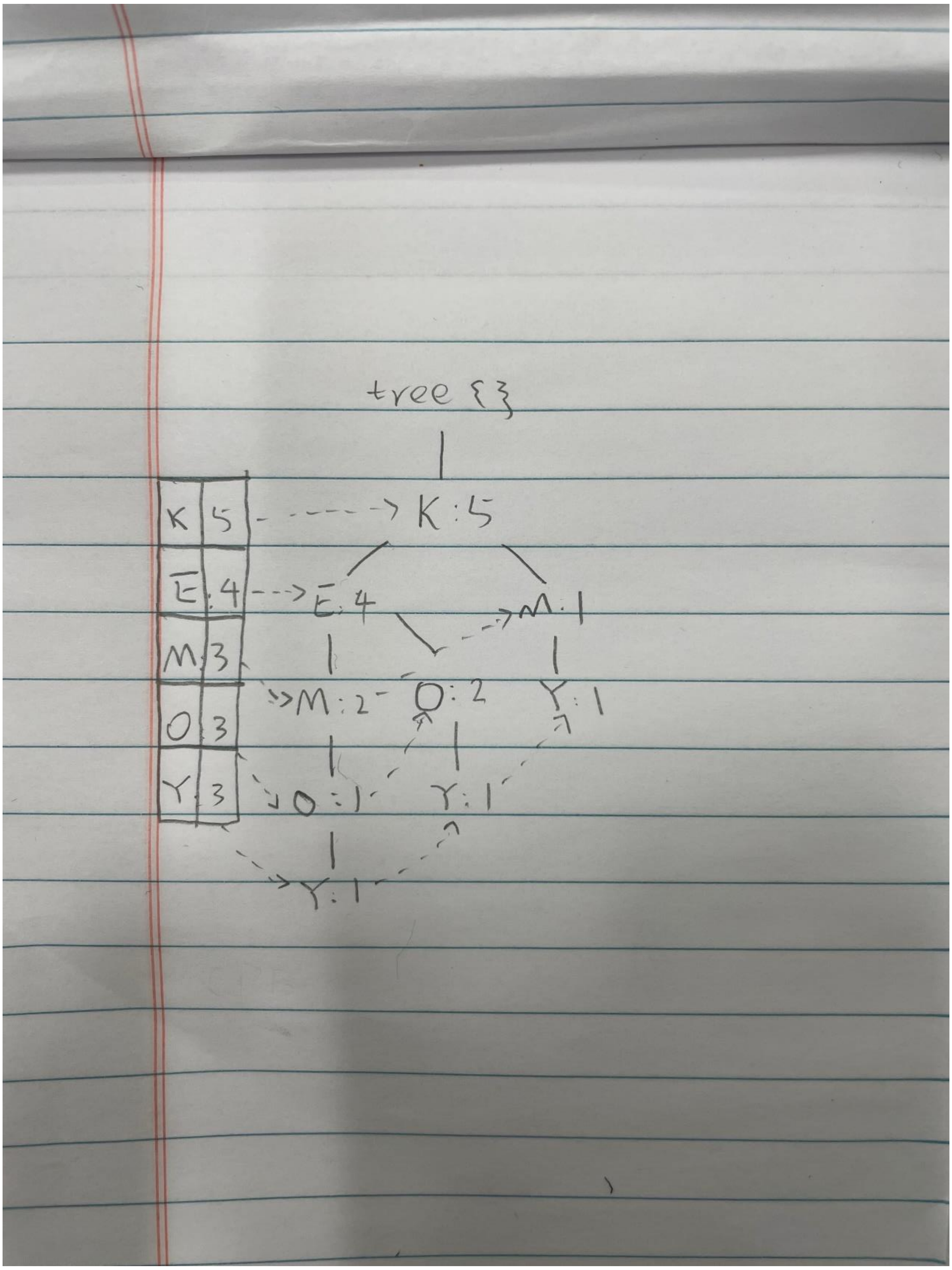
Without all the eliminated rules, here are the strong association rules:
{M} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.6) = 1.67
{O} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 1) = 1
{O} -> {E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{K} -> {E} confidence = 0.8 / 1 = 0.8; lift = 0.8 / (1 * 0.8) = 1
{E} -> {K} confidence = 0.8 / 0.8 = 1; lift = 1
{Y} -> {K} confidence = 0.6 / 0.6 = 1; lift = 1
{O} -> {K, E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{O, K} -> {E} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 0.8) = 1.25
{O, E} -> {K} confidence = 0.6 / 0.6 = 1; lift = 0.6 / (0.6 * 1) = 1

d)
{M} -> {K} since this rule has a confidence of 1, which means in this itemset every time M occurs K occurs, and has a lift of 1.67, which is the highest positive correlation in all the rules.

Exercise 2

a)
{K}: 1, {E} 0.8, {M}: 0.6, {O}: 0.6, {Y}: 0.6

b)



K	5
E	4
M	3
O	3
Y	3

FP-Growth(tree{}, null)
Generate freq. pattern {Y: 3}
CPB for Y: {K, E, M, O: 1}, {K, E, O: 1}, {K, M: 1}

Forms a database of patterns conditioned on Y
Find frequent items, L = {K: 3}
Construct tree with new header
FP-Growth(tree{Y}, {Y})
Recursively mine new FP-tree
Result: single path, which is a base condition
Result:
{K, Y: 3}

Generate freq. pattern {O: 3}
CPB for O: {K, E, M: 1}, {K, E: 2}
Forms a database of patterns conditioned on O
Find frequent items, L = {K, E:3}
Construct tree with new header
FP-Growth(tree{O}, {O})

```
                {O}
                 |
                K: 3
                 |
                E: 3
```

Recursively mine new FP-tree
Result: single path, which is a base condition
Result:
{K, O: 3}, {E, O: 3}, {K, E, O: 3}

Generate freq. pattern {M: 3}
CPB for M: {K, E: 2}, {K: 1}
Forms a database of patterns conditioned on M
Find frequent items, L = {K: 3}
Construct tree with new header
FP-Growth(tree{M}, {M})

```
                {M}
                 |
                K: 3
```

Recursively mine new FP-Tree
Result: single path, which is a base condition
Result:
{K, M: 3}

Generate freq. pattern {E: 4}
CPB for E: {K: 4}
Forms a database of patterns conditioned on E
Find frequent items, L = {K: 4}
Construct tree with new header

FP-Growth(tree{E}, {E})

```
{E}
 |
K: 4
```

Recursively mine new FP-Tree
Result: single path, which is a base condition
Result:
{K, E: 4}

Generate freq. pattern {K:5}
CPB for K: {}
Forms a database of patterns conditioned on K
Find frequent items, L = {}
Construct tree with new header
FP-Growth(tree{K}, {K})
Result: base condition
Result:
{}

All frequent patterns: {K, Y: 3}, {K, O: 3}, {E, O: 3}, {K, M: 3}, {K, E: 4}, {K, E, O: 3}

d)
Apriori generates all possible pairs in order to find the frequent datasets. It eliminates some sets each time it bumps the number of elements up by one, but if the confidence is 0, it will need to go through every possible combinations. FP-Growth only finds the frequent one-element datasets. It then generates the tree using these datasets and find frequent patterns recursively in the tree. This makes the FP-Growth generally more time efficient and more space efficient than the Apriori algorithm.

Apriori Complexity
    d = # of possible items
    n = # of transactions
    O(d^2 n)

FP-Growth
    O(n * log(n) * d^2)

Exercise 3

a)

M:	{T100, T300, T400}
O:	{T100, T200, T500}
N:	{T100, T200}
K:	{T100, T200, T300, T400, T500}
E:	{T100, T200, T300, T500}
Y:	{T100, T200, T400}
D:	{T200}
A:	{T300}
U:	{T400}
C:	{T400, T500}
I:	{T500}

Eliminate items with transactions < 3 (< 0.6 of percentage)

M:	{T100, T300, T400}
O:	{T100, T200, T500}
K:	{T100, T200, T300, T400, T500}
E:	{T100, T200, T300, T500}
Y:	{T100, T200, T400}

b)

Go through all possible pairs by intersection

{M, O}: {T100}
{M, K}: {T100, T300, T400}
{M, E}: {T100, T300}
{M, Y}: {T100, T400}
{O, K}: {T100, T200, T500}
{O, E}: {T100, T200, T500}
{O, Y}: {T100, T200}
{K, E}: {T100, T200, T300 T500}
{K, Y}: {T100, T200, T400}
{E, Y}: {T100, T200}

Eliminate item sets with transactions < 3

{M, K}: {T100, T300, T400}
{O, K}: {T100, T200, T500}

{O, E}: {T100, T200, T500}
{K, E}: {T100, T200, T300 T500}
{K, Y}: {T100, T200, T400}

Go through all pairs by intersection,
Only three-item set: {O, K, E}: {T100, T200, T500}

Exercise 4

a)
Support = 65 / (65 + 40 + 35 + 10) = 65 / 150 = 0.43 > 0.4
Confidence = 65 / (65 + 35) = 0.65 > 0.6
Yes, this is a strong rule.

b)
lift(A, B) = (65 / 150) / (((65 + 35) / 150) * ((65 + 40) / 150))) = 0.8839 < 1
When lift is less than 1, this means A and B have a negative correlation. This means that the rule is still relevant, but as one appears, the other is more likely to not appear.

c)
A and B: ((65 + 40) * (65 + 35)) / 150 = (105 * 100) / 150 = 70
B not A: ((65 + 40) * (40 + 10)) / 150 = (105 * 50) / 150 = 35
A not B: (35 + 10) * (65 + 35) / 150 = 45 * 100 / 150 = 30
Not A not B: (35 + 10) * (40 + 10) / 150 = 45 * 50 / 150 = 15

	A	Not A
B	70	35
Not B	30	15

d)
(65 – 70)^2 / 70 + (40 - 35)^2 / 35 + (35 - 30)^2 / 30 + (10 - 15)^2 / 15 = 1.88
df = (2 – 1)(2 – 1) = 1, significance = 0.05
from Chi-Square table, critical value = 3.84
1.88 < 3.84, we fail to reject the null hypothesis. There is not enough evidence to suggest that the variables are dependent.

e)
support = 35 / 150 = 0.23
confidence = 35 / (65 + 35) = 35 / 105 = 0.33
lift = (35 / 150) / ((100 / 150)(45 / 150)) = 0.78

f)
confidence = 35 / (35 + 10) = 35 / 45 = 0.78

lift = 0.78
This rule is stronger because while having the same lift, this rule has a higher confidence.

g)
(35 / (65 + 35) + 35 / (35 + 10)) / 2 = 0.56

h)
IR(A, -B) = |sup(A)-sup(-B)| / (sup(A) + sup(-B) – sup(A U -B))
= |100 / 150 – 45 / 150| / (100 / 150 + 45 / 150 – 35 / 150) = 0.5

This suggests that the two categories are not equally present in the data. Because the rules have different confidence, this result confirms my observations.

Exercise 5

We first use FP-Growth on each site to find all the frequent patterns on each site. We then find all the possible association rules. We then combine the frequent itemsets to make a global frequent itemset. We then generate global association rules.