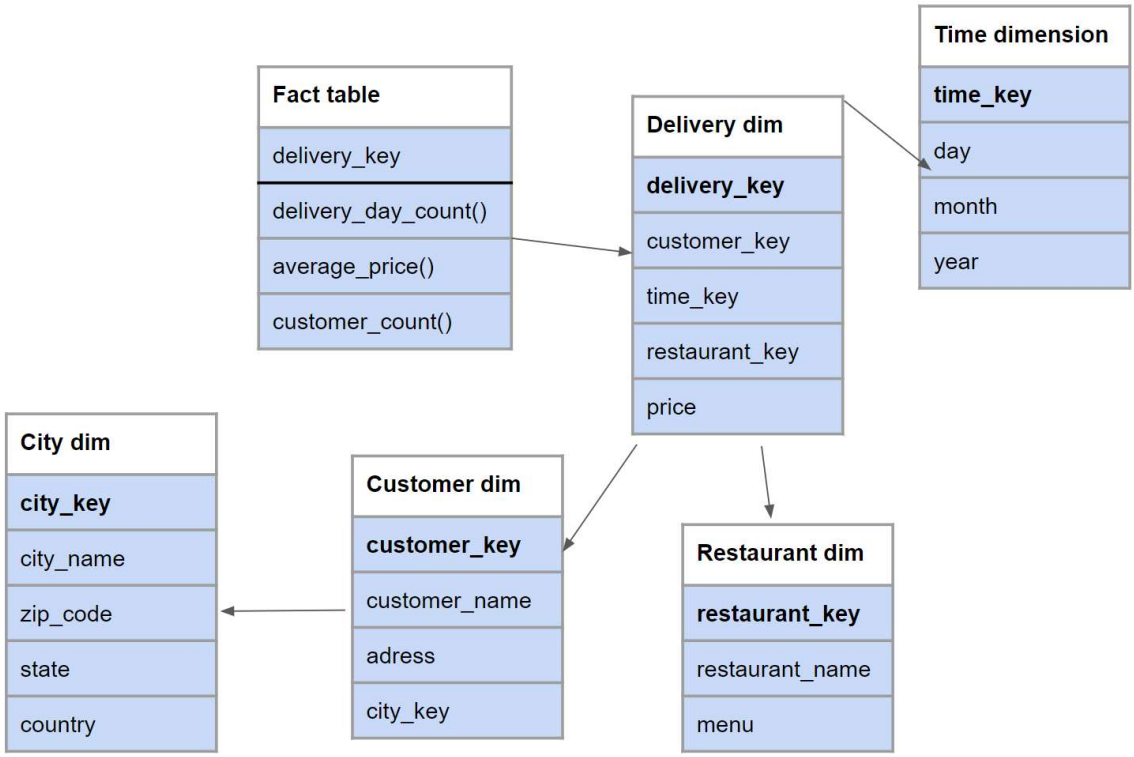
Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

TDT4300 Exercise 5

Task 1: Datawarehousing

a)



b) Only cuboid number 2 can be used to process the query. Since brand is less specific than item_name, the first cuboid cannot be used. Similiarly, the third cuboid has country, which is less specific than city in the location dimension. Therefore you can't “roll up” to item_name and city in these two cuboids. In cuboid 2 you can “drill down” from street to city to get that information.

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

Task 2: Association Rules

Task 2: Apriori algorithm

A)

Minimim Support of 50% (i.e minimum support count is 4).
Using Fk-1 x Fk-1 method for canditate generation:

Initial Itemset

TID	Transaction
T1	BF
T2	ABCDFH
T3	ABF
T4	ABFH
T5	ADEF
T6	ABFH
T7	ABDEFH
T8	AGH

Initial C1 candidate set (K=1)

Itemsets	Sup_count
A	7
B	6
C	1
D	3
E	2
F	7
G	1
H	5

Frequent 1-element sets above min_sup

Itemsets	Sup_count
A	7
B	6
F	7
H	5

Frequent 2-element sets generated from the frequent 1-element set

Itemsets	Sup_count
AB	5
AF	6
AH	5
BF	4
BH	4
FH	4

Frequent 3-element sets generated from the frequent 2-element sets. K = 3, so merging if their 3-2 = 1 first elemts are identical.

		Itemsets	Sup_count
merge(AB,AF)	ABF	ABF	4
merge(AB,AH)	ABH	ABH	4
merge(AF,AH)	AFH	AFH	4
merge(BF, BH)	BFH	BFH	4

Frequent 4-element sets generated from the frequent 3-element sets. K = 4, so merging if their 4-2 = 2 first elemts are identical.

		Itemset	Sup_count
merge(ABF,ABH)	ABFH	ABFH	4

B)

Rule generation for element set {ABH}

Rule	Confidence
{A} -> {BH}	4/7 = 57%
{B} -> {AH}	4/6 = 67%
{H} -> {AB}	4/5 = 80%
{AB} -> {H}	4/5 = 80%
{AH} -> {B}	4/5 = 80%
{BH} -> {A}	4/4 = 100%

Based on the confidence threshold c = 75%, we can see 4 association rules (marked in green)

The numbers from the confidence calculation results from the formula

$c(X \rightarrow Y) = \sigma(X \cup Y) / \sigma(X)$

Task 3: Decision Trees

a) Tree pruning is useful in decision tree induction because it can remove anomalies in the training dataset due to noise. This reduces the complexity of the decision tree and the amount of decisions that has to be made.
One of the drawbacks of using a separate set of tuples to evaluate pruning is that there is a possibility that the tuples are not representative of the training tuples that were used in order to create the decision tree. If so, using

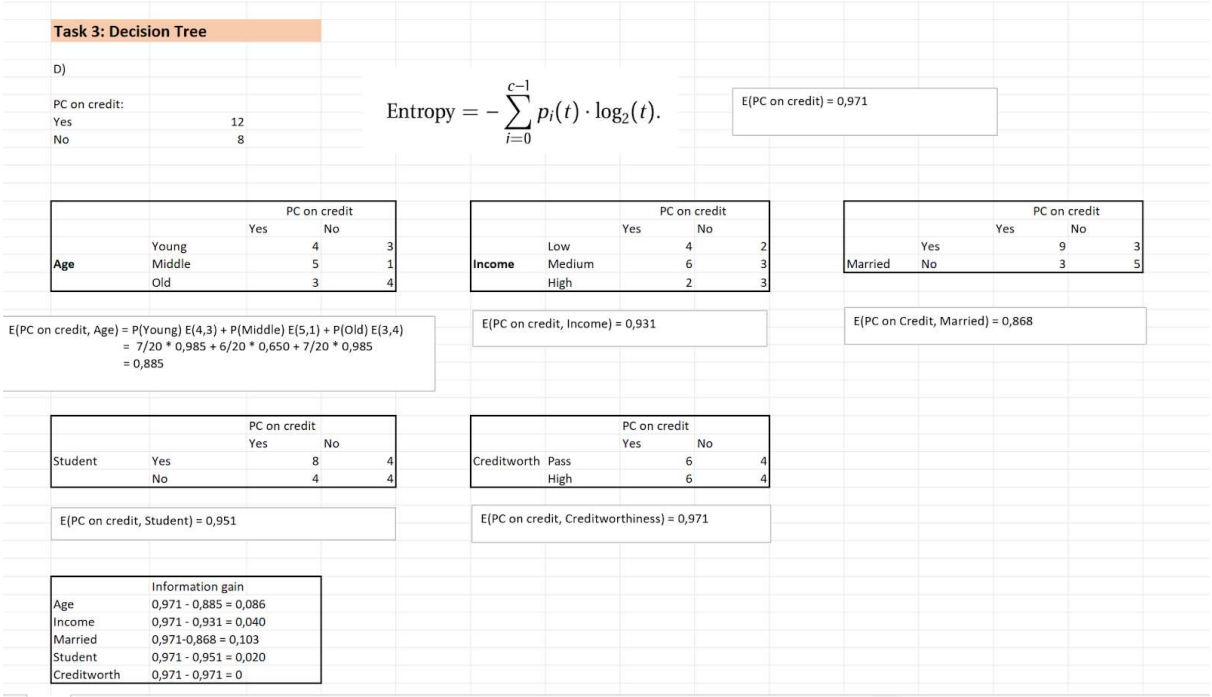
them to evaluate the accuracy of the pruned tree would not be a good indicator.

b) The stopping conditions in decision tree classification is as follows:
   i) Stop expanding a node when all the records belong to the same class.
   ii) Stop expanding a node when all the records have similar attribute values.
   iii) Early termination which is used to avoid too complex trees (i.e avoid overfitting). This can be done by:
      1) Stop if the number of instances is less than some user-specified threshold.
      2) Stop if class distribution of instances are independent of the available features, for example by using a $\chi 2$ test.
      3) Stop if expanding the current node does not improve impurity measures. For example if the Gini-value or information gain is used as an impurity measure and it does not improve.

c)

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

d)



Task 3: Decision Tree

D)

PC on credit:

Yes	12
No	8

$$\text{Entropy} = -\sum_{i=0}^{c-1} p_i(t) \cdot \log_2(t).$$

E(PC on credit) = 0,971

		PC on credit	
		Yes	No
Age	Young	4	3
	Middle	5	1
	Old	3	4

E(PC on credit, Age) = P(Young) E(4,3) + P(Middle) E(5,1) + P(Old) E(3,4)
= 7/20 * 0,985 + 6/20 * 0,650 + 7/20 * 0,985
= 0,885

		PC on credit	
		Yes	No
Income	Low	4	2
	Medium	6	3
	High	2	3

E(PC on credit, Income) = 0,931

		PC on credit	
		Yes	No
Married	Yes	9	3
	No	3	5

E(PC on Credit, Married) = 0,868

		PC on credit	
		Yes	No
Student	Yes	8	4
	No	4	4

E(PC on credit, Student) = 0,951

		PC on credit	
		Yes	No
Creditworth	Pass	6	4
	High	6	4

E(PC on credit, Creditworthiness) = 0,971

	Information gain
Age	0,971 - 0,885 = 0,086
Income	0,971 - 0,931 = 0,040
Married	0,971-0,868 = 0,103
Student	0,971 - 0,951 = 0,020
Creditworth	0,971 - 0,971 = 0

e) Since the attribute Married gives highest information gain, this attribute should be chosen as a splitting attribute.

Task 4: Data Types

(a) Time in terms of AM and PM.

- **Binary** (either AM or PM), **qualitative**, **ordinal** (since AM comes before PM on a specific day)

(b) Brightness as measured by a light meter.

- **Discrete** (not unlimited decimals or values), **quantitative**, **interval** (i.e 3.54 lumen defines interval from 3.535 to 3.5449)

(c) Brightness as measured by people's judgments.

- **Discrete** (not an unlimited amount of descriptions), **qualitative**, **nominal**("very bright")

(d) Angles as measured in degrees between 0 and 360.

- **Discrete** (if you don't count decimals, there are 360 different values), **quantitative, ratio**

(e) Bronze, Silver, and Gold medals as awarded at the Olympics.

- **Discrete**, **qualitative** (not measureable by nominal), **ordinal** (gold is better than silver)

(f) Height above sea level.

- Could argue for continuous and discrete, but in "day to day" speech or in writing it is often referred to in meters, so we'll say **discrete. Quantitative, ratio** (100m is twice the height of 50m etc).

(g) Number of patients in a hospital.

- **Discrete** (finite number). **Quantitative. Ratio.**

(h) ISBN numbers for books. (Look up the format on the Web.)

- **Discrete** (Limited digits), **qualitative**, **nominal** (code, numbers do not have a meaning)

(i) Ability to pass light in terms of the following values: opaque, translucent, transparent.

- **Discrete**, **qualitative**, **ordinal** (can rank by least to most transparent etc)

(j) Military rank.

- **Discrete** (finite number of ranks), **qualitative**, **ordinal** (major > leithenant)

(k) Distance from the center of campus.

- **Discrete** if measured in meters or kilometres. **Quantitative** and **ratio** (1km is twice as far as 500m).

(l) Density of a substance in grams per cubic centimeter.

- **Continuous** (infinite density measures), **quantitative**, **ratio** (water is twice as heavy as …)

(m) Coat check number. (When you attend an event, you can often give your coat to someone who, in turn, gives you a number that you can use to claim your coat when you leave.)

- **Discrete** (numbers 1,2,3 etc.), **qualitative**, **nominal** (represents a code, not a meaningful order).

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

Task 5: Autocorrelation

Daily temperature likely has a higher temportal autocorrelation. It is unlikely that the temperature is doubling overnight, but that is quite common with rainfall on a day-to-day basis.

Task 6: Noise and Outliers

a) Noise is defined as data that should be ignored, and does not contain any valuable information. It is desirable to reduce the noise as much as possible, if the model allows it. Outliers can be valid data points even though they do not fit within the scope of regular data points. Therefore outliers are most times not desirable, but could be interesting for further improvements, analyses etc to the model.
b) It is possible that some instances of noise could appear as outliers, but it is not a common thing.
c) No, sometimes noise objects are in the scope of the model (i.e appear as normal data) and are therefore not always outliers.
d) No, sometimes legitimate data will not fit with the model and the rest of the data, and will therefore appear as an outlier, but it is still not noise.
e) Yes, with enough noise, some typical values could seem unusual, or for instance outliers as typical data.

Task 7: Similarity Measures

Below you can see the code implementation of the different similarity measures and the result for the different tasks can be found in the last picture.

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

```python
import math

# cosine similarity
def cos(x, y):
    dot_product = 0
    lenght_x = 0
    lenght_y = 0
    for val1, val2 in zip(x, y):
        dot_product += val1*val2
        lenght_x += val1**2
        lenght_y += val2**2
    return dot_product / (math.sqrt(lenght_x) * math.sqrt(lenght_y))

# Mean of a vector
def mean(vector):
    n = len(vector)
    summen = sum(vector)
    return summen/n

# Standard deviation
def std_dev(vector):
    s = 0
    m = mean(vector)
    for i in vector:
        s += (i-m)**2
    return math.sqrt(s/(len(vector)-1))


# Correlation
def correlation(x, y):
    s_x = std_dev(x)
    s_y = std_dev(y)
    m_x = mean(x)
    m_y = mean(y)
    n = len(x) - 1
    cov = 0

    for x_i, y_i in zip(x, y):
        c_x = (x_i-m_x)
        c_y = (y_i - m_y)
        cov += c_x * c_y
    return cov/(n*s_x*s_y)

```

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

```python
# Euclidean distance
def euclidean(x, y):
    d = 0
    for i, j in zip(x, y):
        d += (i-j)**2
    return math.sqrt(d)

# Jaccard_coefficent
def jaccard(x, y):
    one_matches = 0
    non_zero_matches = 0
    for i, j in zip(x, y):
        if i and j == 1:
            one_matches += 1
            non_zero_matches += 1
        elif i and j == 0:
            continue
        else:
            non_zero_matches += 1

    return one_matches/non_zero_matches

```

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

```
# Defining vectors:
a_x, a_y = [1, 1, 1, 1], [2, 2, 2, 2]
b_x, b_y = [0, 1, 0, 1], [1, 0, 1, 0]
c_x, c_y = [0, -1, 0, 1], [1, 0, -1, 0]
d_x, d_y = [1, 1, 0, 1, 0, 1], [1, 1, 1, 0, 0, 1]
e_x, e_y = [2, -1, 0, 2, 0, -3], [-1, 1, -1, 0, 0, -1]

# Task a)
print("A) ")
print("Cosine similarity = ", cos(a_x, a_y))
print("Correlation between X and Y is undifined")
print("Euclidean distance = ", euclidean(a_x, a_y))
print("")
print("-------------------------------------------------------------------------")

# Task B)
print("B) ")
print("Cosine similarity = ", cos(b_x, b_y))
print("Correlation = ", correlation(b_x, b_y))
print("Euclidean distance = ", euclidean(b_x, b_y))
print("Jaccard = ", jaccard(b_x, b_y))
print("")
print("-------------------------------------------------------------------------")

# Task C)
print("Task C)")
print("Cosine similarity = ", cos(c_x, c_y))
print("Correlation = ", correlation(c_x, c_y))
print("Euclidean distance = ", euclidean(c_x, c_y))
print("")
print("-------------------------------------------------------------------------")

# Task D)
print("Task D)")
print("Cosine similarity = ", cos(d_x, d_y))
print("Correlation = ", correlation(d_x, d_y))
print("Jaccard = ", jaccard(d_x, d_y))
print("")
print("-------------------------------------------------------------------------")

# Task E
print("Task E")
print("Cosine similarity = ", cos(e_x, e_y))
print("Correlation = ", correlation(e_x, e_y))
```

Arash Aminpour & Sondre Schirmer Mikalsen (group 121)
31.03.22

```
A)
Cosine similarity =  1.0
Correlation between X and Y is undifined
Euclidean distance =  2.0

------------------------------------------------------------------------
B)
Cosine similarity =  0.0
Correlation =  -1.0000000000000002
Euclidean distance =  2.0
Jaccard =  0.0

------------------------------------------------------------------------
Task C)
Cosine similarity =  0.0
Correlation =  0.0
Euclidean distance =  2.0

------------------------------------------------------------------------
Task D)
Cosine similarity =  0.75
Correlation =  0.24999999999999997
Jaccard =  0.6

------------------------------------------------------------------------
Task E
Cosine similarity =  0.0
Correlation =  -5.73316704659901e-17
```