1

2022 年 5 月 23 日

分工：

实习一：王书睿

实习二：虞润飞

实习三：练习一：王书睿，练习二、练习三：罗伟梁，练习四、练习五：王新昊

练习一：各种熵的 SQL 实现

```
[1]: %load_ext sql
```

```
[2]: import pymysql
     pymysql.install_as_MySQLdb()
     %sql mysql://stu1900011117:stu1900011117@162.105.146.37:43306
```

```
[3]: %sql show databases;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
3 rows affected.
```

```
[3]: [('dataset',), ('information_schema',), ('stu1900011117',)]
```

```
[4]: %sql use stu1900011117;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

```
[4]: []
```

```
[154]: %sql create table TheWorldHappinessReport2015 select * from dataset.
         ↪TheWorldHappinessReport2015
       %sql create table TheWorldHappinessReport2016 select * from dataset.
         ↪TheWorldHappinessReport2016
```

```
%sql create table TheWorldHappinessReport2017 select * from dataset.
 ↪TheWorldHappinessReport2017
%sql create table TheWorldHappinessReport2018 select * from dataset.
 ↪TheWorldHappinessReport2018
%sql create table TheWorldHappinessReport2019 select * from dataset.
 ↪TheWorldHappinessReport2019
```

[5]:
```
%sql show tables;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
5 rows affected.
```

[5]:
```
[('TheWorldHappinessReport2015',),
 ('TheWorldHappinessReport2016',),
 ('TheWorldHappinessReport2017',),
 ('TheWorldHappinessReport2018',),
 ('TheWorldHappinessReport2019',)]
```

[6]:
```
%sql select * from TheWorldHappinessReport2015 limit 10;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
10 rows affected.
```

[6]:
```
[(0, 'Switzerland', 1, 7.587, 1.39651, 0.94143, 0.66557, 0.41978, 0.29678,
2015),
 (1, 'Iceland', 2, 7.561, 1.30232, 0.94784, 0.62877, 0.14145, 0.4363, 2015),
 (2, 'Denmark', 3, 7.527, 1.32548, 0.87464, 0.64938, 0.48357, 0.34139, 2015),
 (3, 'Norway', 4, 7.522, 1.459, 0.88521, 0.66973, 0.36503, 0.34699, 2015),
 (4, 'Canada', 5, 7.427, 1.32629, 0.90563, 0.63297, 0.32957, 0.45811, 2015),
 (5, 'Finland', 6, 7.406, 1.29025, 0.88911, 0.64169, 0.41372, 0.23351, 2015),
 (6, 'Netherlands', 7, 7.378, 1.32944, 0.89284, 0.61576, 0.31814, 0.4761, 2015),
 (7, 'Sweden', 8, 7.364, 1.33171, 0.91087, 0.6598, 0.43844, 0.36262, 2015),
 (8, 'New Zealand', 9, 7.286, 1.25018, 0.90837, 0.63938, 0.42922, 0.47501,
2015),
 (9, 'Australia', 10, 7.284, 1.33358, 0.93156, 0.65124, 0.35637, 0.43562, 2015)]
```

信息熵函数

[7]:
```
%sql drop procedure if exists entropy
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

[7]:
```
[]
```

输入 col 为列名，输出 res 为结果

[8]:
```
%%sql

create procedure entropy(in col varchar(30), out res float)
begin
    declare length float;
    # calculate distribution
    set @s = concat('create view e as select count(*) as count from␣
 ↪TheWorldHappinessReport2015 group by ', col);
    prepare stmt from @s;
    execute stmt;
    deallocate prepare stmt;
    set length = (select count(*) from TheWorldHappinessReport2015);
    # calculate entropy
    select sum(-(e.count / length) * LOG(e.count / length)) from e into res;
    drop view e;
end
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

[8]:
```
[]
```

计算 2015 年数据集 happiness 列的熵

[9]:
```
%%sql

call entropy('happiness', @a);
select @a;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
```

```
0 rows affected.
1 rows affected.
```

```
[9]: [(5.053821086883545,)]
```

条件熵函数

```
[10]: %sql drop procedure if exists joined_entropy
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

```
[10]: []
```

```
[11]: %%sql

create procedure joined_entropy(in col1 varchar(30), in col2 varchar(30), out␣
 ↪res float)
begin
    declare length float;
    # calculate distribution
    set @s = concat('create view e as select count(*) as count from␣
 ↪TheWorldHappinessReport2015 group by ', col1, ', ', col2);
    prepare stmt from @s;
    execute stmt;
    deallocate prepare stmt;
    set length = (select count(*) from TheWorldHappinessReport2015);
    # calculate joined_entropy
    select sum(-(e.count / length) * LOG(e.count / length)) from e into res;
    drop view e;
end
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

```
[11]: []
```

```
[12]: %sql drop procedure if exists conditional_entropy
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

```
[12]: []
```

输入 col1 col2 为列名，输出 res 为结果

```
[13]: %%sql

create procedure conditional_entropy(in col1 varchar(30), in col2 varchar(30),␣
 ↪out res float)
begin
    declare res1 float;
    declare res2 float;
    call joined_entropy(col1, col2, res1);
    call entropy(col2, res2);
    set res = res1 - res2;
end
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

```
[13]: []
```

计算 2015 年数据集 happiness 列和 healthy_life_expectancy 列条件熵

```
[14]: %%sql

call conditional_entropy('happiness', 'healthy_life_expectancy', @a);
select @a;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
1 rows affected.
```

```
[14]: [(0.0087738037109375,)]
```

相对熵

根据相对熵公式 DKL(p||q)，当 q(x) 为 0 时会产生除零错误，因此为避免该错误，在 q(x) 值为 0

时，给 q(x) 赋一个很小的值 eps，在下列函数中取 eps=1e-10

[15]:
```
%%sql

drop procedure if exists relative_entropy
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

[15]:
```
[]
```

输入 dataset1 dataset2 为数据集名称，输入 col 为列名，输出 res 为结果

[16]:
```
%%sql

create procedure relative_entropy(in dataset1 varchar(30), in dataset2 varchar(30), in col varchar(30), out res float)
begin
    declare length float;
    declare res1 float;
    declare res2 float;
    declare eps float;
    # calculate distribution
    set @s1 = concat('create view e1 as select ', col, ' as x1, count(*) as c1 from ', dataset1, ' group by ', col);
    prepare stmt from @s1;
    execute stmt;
    deallocate prepare stmt;
    set @s2 = concat('create view e2 as select ', col, ' as x2, count(*) as c2 from ', dataset2, ' group by ', col);
    prepare stmt from @s2;
    execute stmt;
    deallocate prepare stmt;
    create view e as select * from e1 left join e2 on e1.x1=e2.x2;
    set length = (select count(*) from e);
    # calculate entropy
    set @eps = 1e-10;
    select sum((e.c1 / length) * LOG((e.c1 / length) / (e.c2 / length))) from e␣
 ↪where e.x2 is not null into res1;
    select sum((e.c1 / length) * LOG((e.c1 / length) / @eps)) from e where e.x2␣
 ↪is null into res2;
    set res = res1 + res2;

    drop view e;
    drop view e1;
    drop view e2;
end
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
```

[16]:

```
[]
```

计算 2015 年和 2016 年数据集 happiness 分布的相对熵

[17]:

```
%%sql

call relative_entropy('TheWorldHappinessReport2015',␣
 ↪'TheWorldHappinessReport2016', 'happiness', @a);
select @a;
```

```
 * mysql://stu1900011117:***@162.105.146.37:43306
0 rows affected.
1 rows affected.
```

[17]:

```
[(16.719417572021484,)]
```

[ ]: