实习四：基于SQL实现机器学习的基本概念

> 成员：罗逸龙2000017781 占可盈2000013136 范居令2000012942

In [ ]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

In [ ]:
```
import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu2000017781:stu2000017781@162.105.146.37:43306
```

In [ ]:
```
%sql use stu2000017781;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
```

Out[ ]:
```
[]
```

练习一：发现数据中隐含的辛普森悖论

任务一：使用纯SQL生成报表，并判断是否存在辛普森悖论

首先需要根据要求生成数据集

In [ ]:
```
%%sql

DROP TABLE IF EXISTS application;

CREATE TABLE application (
  id INT,
  sex VARCHAR(10),
  department VARCHAR(20),
  acception VARCHAR(10)
);

INSERT INTO application (id, sex, department, acception)
VALUES
    (1, 'Female', 'Business School', 'Accepted'),
    (2, 'Female', 'Business School', 'Accepted'),
    (3, 'Female', 'Business School', 'Accepted'),
    (4, 'Female', 'Business School', 'Accepted'),
    (5, 'Female', 'Business School', 'Accepted'),
    (6, 'Female', 'Business School', 'Accepted'),
    (7, 'Female', 'Business School', 'Accepted'),
    (8, 'Female', 'Business School', 'Accepted'),
    (9, 'Female', 'Business School', 'Accepted'),
    (10, 'Female', 'Business School', 'Accepted'),
    (11, 'Female', 'Business School', 'Accepted'),
```

```
    (12, 'Female', 'Business School', 'Accepted'),
    (13, 'Female', 'Business School', 'Accepted'),
    (14, 'Female', 'Business School', 'Accepted'),
    (15, 'Female', 'Business School', 'Accepted'),
    (16, 'Female', 'Business School', 'Accepted'),
    (17, 'Female', 'Business School', 'Accepted'),
    (18, 'Female', 'Business School', 'Accepted'),
    (19, 'Female', 'Business School', 'Accepted'),
    (20, 'Female', 'Business School', 'Accepted'),
    (21, 'Female', 'Business School', 'Accepted'),
    (22, 'Female', 'Business School', 'Accepted'),
    (23, 'Female', 'Business School', 'Accepted'),
    (24, 'Female', 'Business School', 'Accepted'),
    (25, 'Female', 'Business School', 'Accepted'),
    (26, 'Female', 'Business School', 'Accepted'),
    (27, 'Female', 'Business School', 'Accepted'),
    (28, 'Female', 'Business School', 'Accepted'),
    (29, 'Female', 'Business School', 'Accepted'),
    (30, 'Female', 'Business School', 'Accepted'),
    (31, 'Female', 'Business School', 'Accepted'),
    (32, 'Female', 'Business School', 'Accepted'),
    (33, 'Female', 'Business School', 'Accepted'),
    (34, 'Female', 'Business School', 'Accepted'),
    (35, 'Female', 'Business School', 'Accepted'),
    (36, 'Female', 'Business School', 'Accepted'),
    (37, 'Female', 'Business School', 'Accepted'),
    (38, 'Female', 'Business School', 'Accepted'),
    (39, 'Female', 'Business School', 'Accepted'),
    (40, 'Female', 'Business School', 'Accepted'),
    (41, 'Female', 'Business School', 'Accepted'),
    (42, 'Female', 'Business School', 'Accepted'),
    (43, 'Female', 'Business School', 'Accepted'),
    (44, 'Female', 'Business School', 'Accepted'),
    (45, 'Female', 'Business School', 'Accepted'),
    (46, 'Female', 'Business School', 'Accepted'),
    (47, 'Female', 'Business School', 'Accepted'),
    (48, 'Female', 'Business School', 'Accepted'),
    (49, 'Female', 'Business School', 'Accepted'),
    (50, 'Female', 'Business School', 'Rejected'),
    (51, 'Female', 'Business School', 'Rejected'),
    (52, 'Female', 'Business School', 'Rejected'),
    (53, 'Female', 'Business School', 'Rejected'),
    (54, 'Female', 'Business School', 'Rejected'),
    (55, 'Female', 'Business School', 'Rejected'),
    (56, 'Female', 'Business School', 'Rejected'),
    (57, 'Female', 'Business School', 'Rejected'),
    (58, 'Female', 'Business School', 'Rejected'),
    (59, 'Female', 'Business School', 'Rejected'),
    (60, 'Female', 'Business School', 'Rejected'),
    (61, 'Female', 'Business School', 'Rejected'),
    (62, 'Female', 'Business School', 'Rejected'),
    (63, 'Female', 'Business School', 'Rejected'),
    (64, 'Female', 'Business School', 'Rejected'),
    (65, 'Female', 'Business School', 'Rejected'),
    (66, 'Female', 'Business School', 'Rejected'),
    (67, 'Female', 'Business School', 'Rejected'),
    (68, 'Female', 'Business School', 'Rejected'),
```

```
    (69, 'Female', 'Business School', 'Rejected'),
    (70, 'Female', 'Business School', 'Rejected'),
    (71, 'Female', 'Business School', 'Rejected'),
    (72, 'Female', 'Business School', 'Rejected'),
    (73, 'Female', 'Business School', 'Rejected'),
    (74, 'Female', 'Business School', 'Rejected'),
    (75, 'Female', 'Business School', 'Rejected'),
    (76, 'Female', 'Business School', 'Rejected'),
    (77, 'Female', 'Business School', 'Rejected'),
    (78, 'Female', 'Business School', 'Rejected'),
    (79, 'Female', 'Business School', 'Rejected'),
    (80, 'Female', 'Business School', 'Rejected'),
    (81, 'Female', 'Business School', 'Rejected'),
    (82, 'Female', 'Business School', 'Rejected'),
    (83, 'Female', 'Business School', 'Rejected'),
    (84, 'Female', 'Business School', 'Rejected'),
    (85, 'Female', 'Business School', 'Rejected'),
    (86, 'Female', 'Business School', 'Rejected'),
    (87, 'Female', 'Business School', 'Rejected'),
    (88, 'Female', 'Business School', 'Rejected'),
    (89, 'Female', 'Business School', 'Rejected'),
    (90, 'Female', 'Business School', 'Rejected'),
    (91, 'Female', 'Business School', 'Rejected'),
    (92, 'Female', 'Business School', 'Rejected'),
    (93, 'Female', 'Business School', 'Rejected'),
    (94, 'Female', 'Business School', 'Rejected'),
    (95, 'Female', 'Business School', 'Rejected'),
    (96, 'Female', 'Business School', 'Rejected'),
    (97, 'Female', 'Business School', 'Rejected'),
    (98, 'Female', 'Business School', 'Rejected'),
    (99, 'Female', 'Business School', 'Rejected'),
    (100, 'Female', 'Business School', 'Rejected');

INSERT INTO application (id, sex, department, acception)
VALUES
    (101, 'Male', 'Business School', 'Accepted'),
    (102, 'Male', 'Business School', 'Accepted'),
    (103, 'Male', 'Business School', 'Accepted'),
    (104, 'Male', 'Business School', 'Accepted'),
    (105, 'Male', 'Business School', 'Accepted'),
    (106, 'Male', 'Business School', 'Accepted'),
    (107, 'Male', 'Business School', 'Accepted'),
    (108, 'Male', 'Business School', 'Accepted'),
    (109, 'Male', 'Business School', 'Accepted'),
    (110, 'Male', 'Business School', 'Accepted'),
    (111, 'Male', 'Business School', 'Accepted'),
    (112, 'Male', 'Business School', 'Accepted'),
    (113, 'Male', 'Business School', 'Accepted'),
    (114, 'Male', 'Business School', 'Accepted'),
    (115, 'Male', 'Business School', 'Accepted'),
    (116, 'Male', 'Business School', 'Rejected'),
    (117, 'Male', 'Business School', 'Rejected'),
    (118, 'Male', 'Business School', 'Rejected'),
    (119, 'Male', 'Business School', 'Rejected'),
    (120, 'Male', 'Business School', 'Rejected');

INSERT INTO application (id, sex, department, acception)
```

```
VALUES
    (121, 'Female', 'Law School', 'Accepted'),
    (122, 'Female', 'Law School', 'Rejected'),
    (123, 'Female', 'Law School', 'Rejected'),
    (124, 'Female', 'Law School', 'Rejected'),
    (125, 'Female', 'Law School', 'Rejected'),
    (126, 'Female', 'Law School', 'Rejected'),
    (127, 'Female', 'Law School', 'Rejected'),
    (128, 'Female', 'Law School', 'Rejected'),
    (129, 'Female', 'Law School', 'Rejected'),
    (130, 'Female', 'Law School', 'Rejected'),
    (131, 'Female', 'Law School', 'Rejected'),
    (132, 'Female', 'Law School', 'Rejected'),
    (133, 'Female', 'Law School', 'Rejected'),
    (134, 'Female', 'Law School', 'Rejected'),
    (135, 'Female', 'Law School', 'Rejected'),
    (136, 'Female', 'Law School', 'Rejected'),
    (137, 'Female', 'Law School', 'Rejected'),
    (138, 'Female', 'Law School', 'Rejected'),
    (139, 'Female', 'Law School', 'Rejected'),
    (140, 'Female', 'Law School', 'Rejected');

-- 插入法学院男生申请数据
INSERT INTO application (id, sex, department, acception)
VALUES
    (141, 'Male', 'Law School', 'Accepted'),
    (142, 'Male', 'Law School', 'Accepted'),
    (143, 'Male', 'Law School', 'Accepted'),
    (144, 'Male', 'Law School', 'Accepted'),
    (145, 'Male', 'Law School', 'Accepted'),
    (146, 'Male', 'Law School', 'Accepted'),
    (147, 'Male', 'Law School', 'Accepted'),
    (148, 'Male', 'Law School', 'Accepted'),
    (149, 'Male', 'Law School', 'Accepted'),
    (150, 'Male', 'Law School', 'Accepted'),
    (151, 'Male', 'Law School', 'Rejected'),
    (152, 'Male', 'Law School', 'Rejected'),
    (153, 'Male', 'Law School', 'Rejected'),
    (154, 'Male', 'Law School', 'Rejected'),
    (155, 'Male', 'Law School', 'Rejected'),
    (156, 'Male', 'Law School', 'Rejected'),
    (157, 'Male', 'Law School', 'Rejected'),
    (158, 'Male', 'Law School', 'Rejected'),
    (159, 'Male', 'Law School', 'Rejected'),
    (160, 'Male', 'Law School', 'Rejected'),
    (161, 'Male', 'Law School', 'Rejected'),
    (162, 'Male', 'Law School', 'Rejected'),
    (163, 'Male', 'Law School', 'Rejected'),
    (164, 'Male', 'Law School', 'Rejected'),
    (165, 'Male', 'Law School', 'Rejected'),
    (166, 'Male', 'Law School', 'Rejected'),
    (167, 'Male', 'Law School', 'Rejected'),
    (168, 'Male', 'Law School', 'Rejected'),
    (169, 'Male', 'Law School', 'Rejected'),
    (170, 'Male', 'Law School', 'Rejected'),
    (171, 'Male', 'Law School', 'Rejected'),
    (172, 'Male', 'Law School', 'Rejected'),
```

```
    (173, 'Male', 'Law School', 'Rejected'),
    (174, 'Male', 'Law School', 'Rejected'),
    (175, 'Male', 'Law School', 'Rejected'),
    (176, 'Male', 'Law School', 'Rejected'),
    (177, 'Male', 'Law School', 'Rejected'),
    (178, 'Male', 'Law School', 'Rejected'),
    (179, 'Male', 'Law School', 'Rejected'),
    (180, 'Male', 'Law School', 'Rejected'),
    (181, 'Male', 'Law School', 'Rejected'),
    (182, 'Male', 'Law School', 'Rejected'),
    (183, 'Male', 'Law School', 'Rejected'),
    (184, 'Male', 'Law School', 'Rejected'),
    (185, 'Male', 'Law School', 'Rejected'),
    (186, 'Male', 'Law School', 'Rejected'),
    (187, 'Male', 'Law School', 'Rejected'),
    (188, 'Male', 'Law School', 'Rejected'),
    (189, 'Male', 'Law School', 'Rejected'),
    (190, 'Male', 'Law School', 'Rejected'),
    (191, 'Male', 'Law School', 'Rejected'),
    (192, 'Male', 'Law School', 'Rejected'),
    (193, 'Male', 'Law School', 'Rejected'),
    (194, 'Male', 'Law School', 'Rejected'),
    (195, 'Male', 'Law School', 'Rejected'),
    (196, 'Male', 'Law School', 'Rejected'),
    (197, 'Male', 'Law School', 'Rejected'),
    (198, 'Male', 'Law School', 'Rejected'),
    (199, 'Male', 'Law School', 'Rejected'),
    (200, 'Male', 'Law School', 'Rejected'),
    (201, 'Male', 'Law School', 'Rejected'),
    (202, 'Male', 'Law School', 'Rejected'),
    (203, 'Male', 'Law School', 'Rejected'),
    (204, 'Male', 'Law School', 'Rejected'),
    (205, 'Male', 'Law School', 'Rejected'),
    (206, 'Male', 'Law School', 'Rejected'),
    (207, 'Male', 'Law School', 'Rejected'),
    (208, 'Male', 'Law School', 'Rejected'),
    (209, 'Male', 'Law School', 'Rejected'),
    (210, 'Male', 'Law School', 'Rejected'),
    (211, 'Male', 'Law School', 'Rejected'),
    (212, 'Male', 'Law School', 'Rejected'),
    (213, 'Male', 'Law School', 'Rejected'),
    (214, 'Male', 'Law School', 'Rejected'),
    (215, 'Male', 'Law School', 'Rejected'),
    (216, 'Male', 'Law School', 'Rejected'),
    (217, 'Male', 'Law School', 'Rejected'),
    (218, 'Male', 'Law School', 'Rejected'),
    (219, 'Male', 'Law School', 'Rejected'),
    (220, 'Male', 'Law School', 'Rejected'),
    (221, 'Male', 'Law School', 'Rejected'),
    (222, 'Male', 'Law School', 'Rejected'),
    (223, 'Male', 'Law School', 'Rejected'),
    (224, 'Male', 'Law School', 'Rejected'),
    (225, 'Male', 'Law School', 'Rejected'),
    (226, 'Male', 'Law School', 'Rejected'),
    (227, 'Male', 'Law School', 'Rejected'),
    (228, 'Male', 'Law School', 'Rejected'),
    (229, 'Male', 'Law School', 'Rejected'),
```

```
    (230, 'Male', 'Law School', 'Rejected'),
    (231, 'Male', 'Law School', 'Rejected'),
    (232, 'Male', 'Law School', 'Rejected'),
    (233, 'Male', 'Law School', 'Rejected'),
    (234, 'Male', 'Law School', 'Rejected'),
    (235, 'Male', 'Law School', 'Rejected'),
    (236, 'Male', 'Law School', 'Rejected'),
    (237, 'Male', 'Law School', 'Rejected'),
    (238, 'Male', 'Law School', 'Rejected'),
    (239, 'Male', 'Law School', 'Rejected'),
    (240, 'Male', 'Law School', 'Rejected');
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
100 rows affected.
20 rows affected.
20 rows affected.
100 rows affected.
```

Out[ ]: []

In [ ]:

```
%%sql

SELECT
    '商学院' AS 学院,
    COUNT(CASE WHEN sex = 'Female' THEN 1 END) AS 女生申请人数,
    COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1 END) A
    ROUND(COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1
    COUNT(CASE WHEN sex = 'Male' THEN 1 END) AS 男生申请人数,
    COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 END) AS
    ROUND(COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 EN
    COUNT(*) AS 合计申请人数,
    COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) AS 合计录取人数,
    ROUND(COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) / COUNT(*) * 1
FROM application
WHERE department = 'Business School'

UNION

SELECT
    '法学院' AS 学院,
    COUNT(CASE WHEN sex = 'Female' THEN 1 END) AS 女生申请人数,
    COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1 END) A
    ROUND(COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1
    COUNT(CASE WHEN sex = 'Male' THEN 1 END) AS 男生申请人数,
    COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 END) AS
    ROUND(COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 EN
    COUNT(*) AS 合计申请人数,
    COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) AS 合计录取人数,
    ROUND(COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) / COUNT(*) * 1
FROM application
WHERE department = 'Law School'

UNION

SELECT
    '合计' AS 学院,
    COUNT(CASE WHEN sex = 'Female' THEN 1 END) AS 女生申请人数,
    COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1 END) A
    ROUND(COUNT(CASE WHEN sex = 'Female' AND acception = 'Accepted' THEN 1
    COUNT(CASE WHEN sex = 'Male' THEN 1 END) AS 男生申请人数,
    COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 END) AS
    ROUND(COUNT(CASE WHEN sex = 'Male' AND acception = 'Accepted' THEN 1 EN
    COUNT(*) AS 合计申请人数,
    COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) AS 合计录取人数,
    ROUND(COUNT(CASE WHEN acception = 'Accepted' THEN 1 END) / COUNT(*) * 1
FROM application;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
3 rows affected.
```

Out[ ]:

学院	女生申请人数	女生录取人数	女生录取率	男生申请人数	男生录取人数	男生录取率	合计申请人数	合计录取人数	合计录取率
商学院	100	49	49.00	20	15	75.00	120	64	53.33
法学院	20	1	5.00	100	10	10.00	120	11	9.17
合计	120	50	41.67	120	25	20.83	240	75	31.25

可以看到，无论是商学院还是法学院,男生的录取率均高于女生的录取率，但是在整体上，女生的录取率高于男生的录取率，这说明**存在**辛普森悖论。

任务二：

下面计算条件概率，判断是否存在辛普森悖论

In [ ]:

```
%%sql

# P(yes|男生)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male') AS
FROM application
WHERE sex = 'Male' AND acception = 'Accepted';
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_male
0.2083

In [ ]:

```
%%sql

# P(yes|女生)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female') A
FROM application
WHERE sex = 'Female' AND acception = 'Accepted';
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_female
0.4167

In [ ]:
```
%%sql

# P(no|男生)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male') AS
FROM application
WHERE sex = 'Male' AND acception = 'Rejected';
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_no_male
0.7917

In [ ]:
```
%%sql

# P(no|女生)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female') A
FROM application
WHERE sex = 'Female' AND acception = 'Rejected';
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_no_female
0.5833

In [ ]:
```
%%sql

# P(yes|<男生,商学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male' AND
FROM application
WHERE sex = 'Male' AND department = 'Business school' AND acception = 'Acce
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_male_business
0.7500

In [ ]:
```
%%sql

# P(yes|<女生,商学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female' AN
FROM application
WHERE sex = 'Female' AND department = 'Business school' AND acception = 'Ac
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_female_business
0.4900

In [ ]:
```
%%sql

# P(yes|<男生,法学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male' AND
FROM application
WHERE sex = 'Male' AND department = 'Law school' AND acception = 'Accepted'
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_female_business
0.1000

In [ ]:
```
%%sql

# P(yes|<女生,法学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female' AN
FROM application
WHERE sex = 'Female' AND department = 'Law school' AND acception = 'Accepte
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_female_law
0.0500

In [ ]:
```
%%sql

# P(no|<男生,商学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male' AND
FROM application
WHERE sex = 'Male' AND department = 'Business school' AND acception = 'Reje
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_yes_female_law
0.2500

In [ ]:
```
%%sql

# P(no|<女生,商学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female' AN
FROM application
WHERE sex = 'Female' AND department = 'Business school' AND acception = 'Re
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_no_female_business
0.5100

In [ ]:
```
%%sql

# P(no|<男生,法学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Male' AND
FROM application
WHERE sex = 'Male' AND department = 'Law school' AND acception = 'Rejected'
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_no_male_law
0.9000

In [ ]:
```
%%sql

# P(no|<女生,法学院>)
SELECT COUNT(*) / (SELECT COUNT(*) FROM application WHERE sex = 'Female' AN
FROM application
WHERE sex = 'Female' AND department = 'Law school' AND acception = 'Rejecte
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
1 rows affected.
```

Out[ ]:

P_no_female_law
0.9500

由计算结果我们可以再次看到，P(yes|<男生,商学院>)高于P(yes|<女生,商学院>)，P(yes|<男生,法学院>)高于P(yes|<女生,法学院>)，但是P(yes|<男生>)却低于P(yes|<女生>)，这说明**存在**辛普森悖论。当然我们把yes换成no，P(no|<男生,商学院>)低于P(no|<女生,商学院>)，P(no|<男生,法学院>)低于P(no|<女生,法学院>)，但是P(no|<男生>)却高于P(no|<女生>)，这也说明**存在**辛普森悖论。

练习二：KNN分类

这里我们使用威斯康辛乳腺癌数据集

任务一：属性值的预处理

In [ ]:

```
%%sql

# 首先需要在数据库中建表
DROP TABLE IF EXISTS breast_cancer;

CREATE TABLE breast_cancer (
    id INTEGER PRIMARY KEY,
    radius_mean REAL,
    texture_mean REAL,
    perimeter_mean REAL,
    area_mean REAL,
    smoothness_mean REAL,
    compactness_mean REAL,
    concavity_mean REAL,
    concave_points_mean REAL,
    symmetry_mean REAL,
    fractal_dimension_mean REAL,
    radius_se REAL,
    texture_se REAL,
    perimeter_se REAL,
    area_se REAL,
    smoothness_se REAL,
    compactness_se REAL,
    concavity_se REAL,
    concave_points_se REAL,
    symmetry_se REAL,
    fractal_dimension_se REAL,
    radius_worst REAL,
    texture_worst REAL,
    perimeter_worst REAL,
    area_worst REAL,
    smoothness_worst REAL,
    compactness_worst REAL,
    concavity_worst REAL,
    concave_points_worst REAL,
    symmetry_worst REAL,
    fractal_dimension_worst REAL,
    diagnosis BOOLEAN
);
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
```

Out[ ]: []

In [ ]:

```
user_name = 'stu2000017781'
password = 'stu2000017781'
db_name = 'stu2000017781'

db = pymysql.connect(host='162.105.146.37',user=user_name, password=passwor
port=43306,db=db_name)
cursor = db.cursor()
```

In [ ]:

```
#导入乳腺癌数据
from sklearn.datasets import load_breast_cancer
import pandas as pd
import numpy as np

data = load_breast_cancer()

x = data.data
y = data.target
# name = ['平均半径','平均纹理','平均周长','平均面积','平均光滑度',
#         '平均紧凑度','平均凹度','平均凹点','平均对称','平均分形维数',
#         '半径误差','纹理误差','周长误差','面积误差','平滑度误差',
#         '紧凑度误差','凹度误差','凹点误差','对称误差','分形维数误差',
#         '最差半径','最差纹理','最差的边界','最差的区域','最差的平滑度',
#         '最差的紧凑性','最差的凹陷','最差的凹点','最差的对称性','最差的分形维数',
#         '患病否']
insert_query="INSERT INTO breast_cancer (id,radius_mean,texture_mean,perime
 VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,
for i in range(x.shape[0]):
    values=(i, x[i][0],x[i][1],x[i][2],x[i][3],x[i][4],x[i][5],x[i][6],x[i
    try:
        cursor.execute(insert_query, values)
        db.commit()
    except:
        db.rollback()
```

In [ ]:

```
%%sql

select * from breast_cancer limit 10;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
10 rows affected.
```

Out[ ]:

id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactn
0	17.99	10.38	122.8	1001.0	0.1184	
1	20.57	17.77	132.9	1326.0	0.08474	
2	19.69	21.25	130.0	1203.0	0.1096	
3	11.42	20.38	77.58	386.1	0.1425	
4	20.29	14.34	135.1	1297.0	0.1003	
5	12.45	15.7	82.57	477.1	0.1278	
6	18.25	19.98	119.6	1040.0	0.09463	
7	13.71	20.83	90.2	577.9	0.1189	
8	13.0	21.82	87.5	519.8	0.1273	
9	12.46	24.04	83.97	475.9	0.1186	

为了将除了id与diagnosis之外的属性值转化为[0, 1]之间的数值，另一方面数据的极差相较于数据于最小值的差值不大，因此我们使用最小-最大规范化。

In [ ]:

```
%%sql

SET @min_radius_mean = (SELECT MIN(radius_mean) from breast_cancer);
SET @max_radius_mean = (SELECT MAX(radius_mean) from breast_cancer);
SET @min_texture_mean = (SELECT MIN(texture_mean) from breast_cancer);
SET @max_texture_mean = (SELECT MAX(texture_mean) from breast_cancer);
SET @min_perimeter_mean = (SELECT MIN(perimeter_mean) from breast_cancer);
SET @max_perimeter_mean = (SELECT MAX(perimeter_mean) from breast_cancer);
SET @min_area_mean = (SELECT MIN(area_mean) from breast_cancer);
SET @max_area_mean = (SELECT MAX(area_mean) from breast_cancer);
SET @min_smoothness_mean = (SELECT MIN(smoothness_mean) from breast_cancer)
SET @max_smoothness_mean = (SELECT MAX(smoothness_mean) from breast_cancer)
SET @min_compactness_mean = (SELECT MIN(compactness_mean) from breast_cance
SET @max_compactness_mean = (SELECT MAX(compactness_mean) from breast_cance
SET @min_concavity_mean = (SELECT MIN(concavity_mean) from breast_cancer);
SET @max_concavity_mean = (SELECT MAX(concavity_mean) from breast_cancer);
SET @min_concave_points_mean = (SELECT MIN(concave_points_mean) from breast
SET @max_concave_points_mean = (SELECT MAX(concave_points_mean) from breast
SET @min_symmetry_mean = (SELECT MIN(symmetry_mean) from breast_cancer);
SET @max_symmetry_mean = (SELECT MAX(symmetry_mean) from breast_cancer);
SET @min_fractal_dimension_mean = (SELECT MIN(fractal_dimension_mean) from
SET @max_fractal_dimension_mean = (SELECT MAX(fractal_dimension_mean) from
SET @min_radius_se = (SELECT MIN(radius_se) from breast_cancer);
SET @max_radius_se = (SELECT MAX(radius_se) from breast_cancer);
SET @min_texture_se = (SELECT MIN(texture_se) from breast_cancer);
SET @max_texture_se = (SELECT MAX(texture_se) from breast_cancer);
SET @min_perimeter_se = (SELECT MIN(perimeter_se) from breast_cancer);
SET @max_perimeter_se = (SELECT MAX(perimeter_se) from breast_cancer);
SET @min_area_se = (SELECT MIN(area_se) from breast_cancer);
SET @max_area_se = (SELECT MAX(area_se) from breast_cancer);
SET @min_smoothness_se = (SELECT MIN(smoothness_se) from breast_cancer);
SET @max_smoothness_se = (SELECT MAX(smoothness_se) from breast_cancer);
SET @min_compactness_se = (SELECT MIN(compactness_se) from breast_cancer);
SET @max_compactness_se = (SELECT MAX(compactness_se) from breast_cancer);
SET @min_concavity_se = (SELECT MIN(concavity_se) from breast_cancer);
SET @max_concavity_se = (SELECT MAX(concavity_se) from breast_cancer);
SET @min_concave_points_se = (SELECT MIN(concave_points_se) from breast_can
SET @max_concave_points_se = (SELECT MAX(concave_points_se) from breast_can
SET @min_symmetry_se = (SELECT MIN(symmetry_se) from breast_cancer);
SET @max_symmetry_se = (SELECT MAX(symmetry_se) from breast_cancer);
SET @min_fractal_dimension_se = (SELECT MIN(fractal_dimension_se) from brea
SET @max_fractal_dimension_se = (SELECT MAX(fractal_dimension_se) from brea
SET @min_radius_worst = (SELECT MIN(radius_worst) from breast_cancer);
SET @max_radius_worst = (SELECT MAX(radius_worst) from breast_cancer);
SET @min_texture_worst = (SELECT MIN(texture_worst) from breast_cancer);
SET @max_texture_worst = (SELECT MAX(texture_worst) from breast_cancer);
SET @min_perimeter_worst = (SELECT MIN(perimeter_worst) from breast_cancer)
SET @max_perimeter_worst = (SELECT MAX(perimeter_worst) from breast_cancer)
SET @min_area_worst = (SELECT MIN(area_worst) from breast_cancer);
SET @max_area_worst = (SELECT MAX(area_worst) from breast_cancer);
SET @min_smoothness_worst = (SELECT MIN(smoothness_worst) from breast_cance
SET @max_smoothness_worst = (SELECT MAX(smoothness_worst) from breast_cance
SET @min_compactness_worst = (SELECT MIN(compactness_worst) from breast_can
```

```
SET @max_compactness_worst = (SELECT MAX(compactness_worst) from breast_can
SET @min_concavity_worst = (SELECT MIN(concavity_worst) from breast_cancer
SET @max_concavity_worst = (SELECT MAX(concavity_worst) from breast_cancer
SET @min_concave_points_worst = (SELECT MIN(concave_points_worst) from brea
SET @max_concave_points_worst = (SELECT MAX(concave_points_worst) from brea
SET @min_symmetry_worst = (SELECT MIN(symmetry_worst) from breast_cancer);
SET @max_symmetry_worst = (SELECT MAX(symmetry_worst) from breast_cancer);
SET @min_fractal_dimension_worst = (SELECT MIN(fractal_dimension_worst) fro
SET @max_fractal_dimension_worst = (SELECT MAX(fractal_dimension_worst) fro
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

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

Out[ ]: []

In [ ]:

```
%%sql

UPDATE breast_cancer
SET
    radius_mean = (radius_mean - @min_radius_mean) / (@max_radius_mean - @r
    texture_mean = (texture_mean - @min_texture_mean) / (@max_texture_mean
    perimeter_mean = (perimeter_mean - @min_perimeter_mean) / (@max_perimet
    area_mean = (area_mean - @min_area_mean) / (@max_area_mean - @min_area_
    smoothness_mean = (smoothness_mean - @min_smoothness_mean) / (@max_smoo
    compactness_mean = (compactness_mean - @min_compactness_mean) / (@max_c
    concavity_mean = (concavity_mean - @min_concavity_mean) / (@max_concavi
    concave_points_mean = (concave_points_mean - @min_concave_points_mean)
    symmetry_mean = (symmetry_mean - @min_symmetry_mean) / (@max_symmetry_m
    fractal_dimension_mean = (fractal_dimension_mean - @min_fractal_dimensi
    radius_se = (radius_se - @min_radius_se) / (@max_radius_se - @min_radiu
    texture_se = (texture_se - @min_texture_se) / (@max_texture_se - @min_t
    perimeter_se = (perimeter_se - @min_perimeter_se) / (@max_perimeter_se
    area_se = (area_se - @min_area_se) / (@max_area_se - @min_area_se),
    smoothness_se = (smoothness_se - @min_smoothness_se) / (@max_smoothness
    compactness_se = (compactness_se - @min_compactness_se) / (@max_compact
    concavity_se = (concavity_se - @min_concavity_se) / (@max_concavity_se
    concave_points_se = (concave_points_se - @min_concave_points_se) / (@ma
    symmetry_se = (symmetry_se - @min_symmetry_se) / (@max_symmetry_se - @m
    fractal_dimension_se = (fractal_dimension_se - @min_fractal_dimension_s
    radius_worst = (radius_worst - @min_radius_worst) / (@max_radius_worst
    texture_worst = (texture_worst - @min_texture_worst) / (@max_texture_wo
    perimeter_worst = (perimeter_worst - @min_perimeter_worst) / (@max_peri
    area_worst = (area_worst - @min_area_worst) / (@max_area_worst - @min_a
    smoothness_worst = (smoothness_worst - @min_smoothness_worst) / (@max_s
    compactness_worst = (compactness_worst - @min_compactness_worst) / (@ma
    concavity_worst = (concavity_worst - @min_concavity_worst) / (@max_conc
    concave_points_worst = (concave_points_worst - @min_concave_points_wors
    symmetry_worst = (symmetry_worst - @min_symmetry_worst) / (@max_symmetr
    fractal_dimension_worst = (fractal_dimension_worst - @min_fractal_dimer
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
569 rows affected.
```

Out[ ]: []

In [ ]:
```
%%sql

select * from breast_cancer limit 10;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
10 rows affected.
```

Out[ ]:

id	radius_mean	texture_mean	perimeter_mean	area_m
0	0.5210374366983767	0.022658099425092997	0.5459885287817012	0.3637327677624
1	0.6431444933503716	0.2725735542779844	0.6157832907193699	0.5015906680805
2	0.6014955748024045	0.39026039905309434	0.5957432105590491	0.4494167550371
3	0.21009039708457572	0.36083868785931683	0.23350148573008084	0.10290562036055
4	0.6298925647214729	0.15657761244504562	0.6309861101513371	0.4892895015906
5	0.25883856311231007	0.20257017247210005	0.26798424435077045	0.141505832449628
6	0.5333427989966397	0.3473114643219479	0.5238753368806579	0.3802757158006
7	0.3184722419423542	0.37605681433885685	0.320710386289821	0.1842629904559
8	0.2848691372047897	0.40953669259384506	0.3020523806233156	0.15961823966065
9	0.259311846277628	0.48461278322624274	0.277658765807477	0.1409968186638

任务二：数据集的划分

In [ ]:
```
%%sql

ALTER TABLE breast_cancer
ADD COLUMN random_number DOUBLE;

UPDATE breast_cancer
SET random_number = RAND();
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
569 rows affected.
```

Out[ ]:
```
[]
```

In [ ]:
```
%%sql

ALTER TABLE breast_cancer
ADD COLUMN is_train BOOLEAN;

UPDATE breast_cancer
SET is_train = IF(random_number < 0.7, TRUE, FALSE);
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
569 rows affected.
```

Out[ ]: []

任务三：实现KNN算法

In [ ]:
```
%%sql

DROP TABLE IF EXISTS breast_cancer_distance;

CREATE TABLE breast_cancer_distance(
    id_from INT,
    id_to INT,
    distance DOUBLE,
    is_diagnosis BOOLEAN
)
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
```

Out[ ]: []

In [ ]:
```
%%sql

# 计算测试集与训练集之间的距离
INSERT INTO breast_cancer_distance(id_from, id_to, distance, is_diagnosis)
SELECT bc1.id, bc2.id, SQRT(POW(bc1.radius_mean-bc2.radius_mean, 2) + POW(b
    POW(bc1.area_mean-bc2.area_mean, 2) + POW(bc1.smoothness_mean-bc2.smoot
    + POW(bc1.concave_points_mean-bc2.concave_points_mean, 2) + POW(bc1.sym
    + POW(bc1.texture_se-bc2.texture_se, 2) + POW(bc1.perimeter_se-bc2.peri
    POW(bc1.concavity_se-bc2.concavity_se, 2) + POW(bc1.concave_points_se-b
    + POW(bc1.radius_worst-bc2.radius_worst, 2) + POW(bc1.texture_worst-bc2
    + POW(bc1.compactness_worst-bc2.compactness_worst, 2) + POW(bc1.concavi
FROM breast_cancer AS bc1, breast_cancer AS bc2
WHERE (NOT bc1.is_train) AND bc2.is_train
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
66420 rows affected.
```

Out[ ]: []

In [ ]:
```
%%sql

SELECT * FROM breast_cancer_distance LIMIT 10;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
10 rows affected.
```

Out[ ]:

id_from	id_to	distance	is_diagnosis
551	3	2.0615665374362266	0
546	3	2.3673580363398194	0
541	3	1.797292671285451	0
537	3	1.7014808802258243	0
532	3	2.2821060246488094	0
531	3	1.9791727527006795	0
529	3	2.107046237852332	0
527	3	2.2488835842504864	0
526	3	1.9048131953122636	0
523	3	1.9123822621797186	0

这里比较难受的一个点是因为当前的MySQL Server不支持LIMIT字句后面跟随一个变量， 因此我们后续选取K值只能手动选取。

In [ ]:

```
%%sql

ALTER TABLE breast_cancer
ADD COLUMN predict_true BOOLEAN;

UPDATE breast_cancer
SET predict_true = IF(diagnosis = (SELECT is_diagnosis
FROM
        (SELECT bcd.id_to, bcd.is_diagnosis
        FROM breast_cancer_distance AS bcd
        WHERE (breast_cancer.id = bcd.id_from)
        ORDER BY distance ASC
        LIMIT 13) as t
GROUP BY is_diagnosis
ORDER BY COUNT(*) DESC
LIMIT 1), TRUE, FALSE);
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
569 rows affected.
```

Out[ ]:

```
[]
```

In [ ]:

```
%%sql

SELECT * FROM breast_cancer LIMIT 30;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
30 rows affected.
```

Out[ ]:

id	radius_mean	texture_mean	perimeter_mean	area_
0	0.5210374366983767	0.022658099425092997	0.5459885287817012	0.363732767762
1	0.6431444933503716	0.2725735542779844	0.6157832907193699	0.501590668080
2	0.6014955748024045	0.39026039905309434	0.5957432105590491	0.44941675503
3	0.21009039708457572	0.36083868785931683	0.23350148573008084	0.1029056203605
4	0.6298925647214729	0.15657761244504562	0.6309861101513371	0.489289501590
5	0.25883856311231007	0.20257017247210005	0.26798424435077045	0.1415058324496
6	0.5333427989966397	0.3473114643219479	0.5238753368806579	0.380275715800
7	0.3184722419423542	0.37605681433885685	0.320710386289821	0.184262990455
8	0.2848691372047897	0.40953669259384506	0.3020523806233156	0.1596182396606
9	0.259311846277628	0.48461278322624274	0.277658765807477	0.140996818663
10	0.42780065313076815	0.4575583361515048	0.40709004215327205	0.277539766702
11	0.41644185716314075	0.2766317213391951	0.4133093773754405	0.270413573700
12	0.5768848502058783	0.5103145079472439	0.6123281044848318	0.415482502651
13	0.4197548393203654	0.4815691579303347	0.41400041462234816	0.2711346765641
14	0.3194188082729898	0.43625295908014877	0.3442056526846797	0.184432661717
15	0.3577547446637323	0.6029759891782212	0.36583511851288786	0.2185790031813
16	0.3643807089781817	0.3523841731484612	0.3520834772994264	0.2294803817603
17	0.43300676794926407	0.37098410551234356	0.4444060534862829	0.2779639448568
18	0.607174972786218	0.42069665201217443	0.5957432105590491	0.47359490986
19	0.31042642813195137	0.15725397362191404	0.3017759657245525	0.179342523860
20	0.28865540252733213	0.2029083530605343	0.28912998410614327	0.1597030752916
21	0.11940934260968337	0.09232330064254307	0.1143666643632092	0.0553128313891
22	0.3956173978891571	0.1538721677375718	0.4057079676594568	0.2379215270413
23	0.6710682001041224	0.45079472438282037	0.6454978923363969	0.534676564156
24	0.45761749254579015	0.3946567467027392	0.4575357611775275	0.322841993637
25	0.4808083676463629	0.22624281366249568	0.498997995991984	0.32627783669
26	0.3596478773250036	0.39972945552925265	0.37053417179185955	0.2126405090137
27	0.5503809929480808	0.3564423402096719	0.541151268053348	0.403181336161
28	0.39372426522788595	0.5262089956036523	0.40501693041254927	0.249798515376
29	0.5011595437550287	0.18058843422387555	0.4920876235229079	0.344262990455

In [ ]:
```
%%sql

SET @num_sum = (SELECT COUNT(*) FROM breast_cancer WHERE NOT is_train);
SET @num_true = (SELECT COUNT(*) FROM breast_cancer WHERE predict_true AND

SELECT @num_true / @num_sum;
```

```
 * mysql://stu2000017781:***@162.105.146.37:43306
0 rows affected.
0 rows affected.
1 rows affected.
```

Out[ ]:

@num_true / @num_sum
0.9634

任务四：讨论最佳K

下面我们将实验不同k值的结果，找出最优的k值。 正如前文所提到的，由于MySQL Server不支持LIMIT字句后面跟随一个变量，因此我们只能手动选取k值。为了节约篇幅，我们直接给出实验结果而不修改代码（实际上代码修改很简单，只需要修改LIMIT之后的常数即可）。 在knn中，我们一般选择k值为奇数来避免出现平票的情况，另一方面经验认为k=5效果比较好，综上我们选取了1,3,5,7,9,11,13这七个k值进行实验。

K	1	3	5	7	9	11	13
Acurracy	0.9630	0.9809	0.9706	0.9778	0.9632	0.9689	0.9634

可以看到在威斯康辛乳腺癌数据集上，当k=3时，准确率最高，为0.9809。

与scikit-learn的结果对比

下面我们用scikit-learn中的KNN算法进行实验，与我们自己实现的KNN算法进行对比。

In [ ]:
```
name = ['平均半径','平均纹理','平均周长','平均面积','平均光滑度',
        '平均紧凑度','平均凹度','平均凹点','平均对称','平均分形维数',
        '半径误差','纹理误差','周长误差','面积误差','平滑度误差',
        '紧凑度误差','凹度误差','凹点误差','对称误差','分形维数误差',
        '最差半径','最差纹理','最差的边界','最差的区域','最差的平滑度',
        '最差的紧凑性','最差的凹陷','最差的凹点','最差的对称性','最差的分形维数',
        '患病否']

breast_cancer = pd.DataFrame(np.concatenate((x,y.reshape(-1,1)),axis=1),col
breast_cancer.head()
```

Out[ ]:

	平均半径	平均纹理	平均周长	平均面积	平均光滑度	平均紧凑度	平均凹度	平均凹点	平均对称	平均分形维数	...	
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	1
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	2
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	1(

5 rows × 31 columns

In [ ]:

```
#划分训练集和测试集
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3,random_state
```

In [ ]:

```
#归一化
from sklearn.preprocessing import MinMaxScaler as mms

mms_01 = mms().fit(xtrain) #求训练集最大/最小值
mms_02 = mms().fit(xtest) #求测试集最大/最小值

#转化
x_train = mms_01.transform(xtrain)
x_test = mms_02.transform(xtest)
```

In [ ]:

```
#导入包
from sklearn.neighbors import KNeighborsClassifier

#建立模型
neighbors = [1, 3, 5, 7, 9, 11, 13]
for nn in neighbors:
    clf = KNeighborsClassifier(n_neighbors=nn)
    clf = clf.fit(x_train,ytrain)
    print(clf.score(x_test,ytest))
```

```
0.9415204678362573
0.9649122807017544
0.9766081871345029
0.9766081871345029
0.9766081871345029
0.9649122807017544
0.9707602339181286
```

scikit-learn中的KNN算法的结果如下：

K	1	3	5	7	9	11	13
Acurracy	0.9415	0.9649	0.9766	0.9766	0.9766	0.9649	0.9707

可以看到在威斯康辛乳腺癌数据集上，当k=5或7时，准确率最高，为0.9766。 总体来说，无论是scikit-learn中的knn算法还是我们自己使用SQL实现的knn,在预测的准确率上都十分相近，均位于0.96-0.98之间。这在一定程度上说明了我们自己用SQL实现的knn算法的正确性。