Bike Sharing Demand

데이터분석과 시각화, 머신러닝 알고리즘으로 시간당 자전거 대여량을 예측하기

이번 캐글 경진대회는 시간당 자전거 대여량을 예측하는 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 입니다. 워싱턴 D.C 소재의 자전거 대여 스타트업 Capital Bikeshare (https://www.capitalbikeshare.com/)의 데이터를 활용하여, 특정 시간대에 얼마나 많은 사람들이 자전거를 대여하는지 예측하는 것이 목표입니다.

사람들이 자전거를 대여하는데는 많은 요소가 관여되어 있을 겁니다. 가령 시간(새벽보다 낮에 많이 빌리겠죠), 날씨(비가 오면 자전거를 대여하지 않을 겁니다), 근무일(근무 시간에는 자전거를 대여하지 않겠죠) 등. 이런 모든 요소를 조합하여 워싱턴 D.C의 자전거 교통량을 예측해주세요. 이번 경진대회에서는 기존까지 배웠던 프로그래밍 언어와 인공지능&머신러닝 능력 외에도, 자전거 렌탈 시장에 대한 약간의 전문지식, 그리고 일반인의 기초 상식을 총동원 할 수 있습니다.

저번 Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic/) 경진대회와 마찬가지로, 이번에도 프로그래밍 언어 파이썬(Python (https://www.python.org/)), 데이터 분석 패키지 판다스(Pandas (https://pandas.pydata.org/)), 그리고 머신러닝&인공지능 라이브러리인 싸이킷런(scikit-learn (scikit-learn.org))을 사용합니다. 여기에 더불어, 이번에는 데이터 시각화 패키지 matplotlib (https://matplotlib.org/)와 Seaborn (https://seaborn.pydata.org/)을 본격적으로 활용해볼 것입니다.

컬럼 설명

(데이터는 다음의 링크 (https://www.kaggle.com/c/bike-sharing-demand/data)에서 다운받으실 수 있습니다)

- **datetime** - 시간. 연-월-일 시:분:초 로 표현합니다. (가령 2011-01-01 00:00:00은 2011년 1월 1일 0시 0분 0초)
- **season** - 계절. 봄(1), 여름(2), 가을(3), 겨울(4) 순으로 표현합니다.
- **holiday** - 공휴일. 1이면 공휴일이며, 0이면 공휴일이 아닙니다.
- **workingday** - 근무일. 1이면 근무일이며, 0이면 근무일이 아닙니다.
- **weather** - 날씨. 1 ~ 4 사이의 값을 가지며, 구체적으로는 다음과 같습니다.
  - 1: 아주 깨끗한 날씨입니다. 또는 아주 약간의 구름이 끼어있습니다.
  - 2: 약간의 안개와 구름이 끼어있는 날씨입니다.
  - 3: 약간의 눈, 비가 오거나 천둥이 칩니다.
  - 4: 아주 많은 비가 오거나 우박이 내립니다.
- **temp** - 온도. 섭씨(Celsius)로 적혀있습니다.
- **atemp** - 체감 온도. 마찬가지로 섭씨(Celsius)로 적혀있습니다.
- **humidity** - 습도.
- **windspeed** - 풍속.
- **casual** - 비회원(non-registered)의 자전거 대여량.
- **registered** - 회원(registered)의 자전거 대여량.
- **count** - 총 자전거 대여랑. 비회원(casual) + 회원(registered)과 동일합니다.

In [1]:

```
# 파이썬의 데이터 분석 패키지 Pandas(pandas.pydata.org) 를 읽어옵니다.
# Pandas는 쉽게 말해 파이썬으로 엑셀을 다룰 수 있는 툴이라고 보시면 됩니다.
# 이 패키지를 앞으로는 pd라는 축약어로 사용하겠습니다.
import pandas as pd
```

Load Dataset

언제나처럼 모든 데이터 분석의 시작은 주어진 데이터를 읽어오는 것입니다. 판다스(Pandas) (https://pandas.pydata.org/)의 read_csv (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html)를 활용하여 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 경진대회에서 제공하는 두 개의 데이터(train, test)를 읽어오겠습니다. (다운로드 링크 (https://www.kaggle.com/c/bike-sharing-demand/data))

앞서 Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic/) 경진대회와 마찬가지로, 여기에서도 파일의 경로를 지정하는 방법에 주의하셔야 합니다. 만일 read_csv를 실행할 때 (**FileNotFoundError**)라는 이름의 에러가 난다면 경로가 제대로 지정이 되지 않은 것입니다. **파일의 경로를 지정하는 법이 생각나지 않는다면 다음의 링크 (http://88240.tistory.com/122)를 통해 경로를 지정하는 법을 복습한 뒤 다시 시도해주세요.**

In [2]:

```
# 판다스의 read_csv로 train.csv 파일을 읽어옵니다.
# 여기서 datetime은 특별히 날짜로 해석하기 위해 parse_dates 옵션에 넣어줍니다.
# 읽어온 데이터를 train이라는 이름의 변수에 할당합니다.
train = pd.read_csv("data/bike/train.csv", parse_dates=["datetime"])

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# head()로 train 데이터의 상위 5개를 띄웁니다.
train.head()
```

```
(10886, 12)
```

Out[2]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspe
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0

In [3]:

```
# train.csv 파일을 읽어온 방식과 동일하게 test.csv를 읽어옵니다.
# 이후 이 데이터를 test라는 이름의 변수에 저장합니다.
test = pd.read_csv("data/bike/test.csv", parse_dates=["datetime"])

# 마찬가지로 행렬(row, column) 사이즈를 출력하고
print(test.shape)

# 전체 test 데이터에서 상위 5개만 출력합니다.
test.head()
```

(6493, 9)

Out[3]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspee
0	2011-01-20 00:00:00	1	0	1	1	10.66	11.365	56	26.0027
1	2011-01-20 01:00:00	1	0	1	1	10.66	13.635	56	0.0000
2	2011-01-20 02:00:00	1	0	1	1	10.66	13.635	56	0.0000
3	2011-01-20 03:00:00	1	0	1	1	10.66	12.880	56	11.0014
4	2011-01-20 04:00:00	1	0	1	1	10.66	12.880	56	11.0014

Preprocessing

데이터를 읽어왔으면, 이 데이터를 편하게 분석하고 머신러닝 알고리즘에 집어넣기 위해 간단한 전처리(Preprocessing) 작업을 진행하겠습니다.

Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand)는 편리하게도 대부분의 데이터가 전처리 되어있습니다. (가령 season 컬럼은 봄을 spring이라 표현하지 않고 1이라고 표현합니다) 그러므로 Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic/) 경진대회와는 달리 간단한 전처리만 끝내면 바로 머신러닝 모델에 데이터를 집어넣을 수 있습니다.

Parse datetime

먼저 **날짜(datetime)** 컬럼을 전처리 하겠습니다.

날짜 컬럼은 얼핏 보면 여러개의 숫자로 구성되어 있습니다. (ex: 2011-01-01 00:00:00) 하지만 결론적으로 숫자는 아니며, 판다스에서는 문자열(object) 또는 날짜(datetime64)로 인식합니다. (값에 하이픈(-)과 콜론(:)이 있기 때문입니다) 그러므로 날짜(datetime) 컬럼을 사용하기 위해서는 머신러닝 알고리즘이 이해할 수 있는 방식으로 전처리를 해줘야 합니다.

날짜(datetime) 컬럼을 전처리하는 가장 쉬운 방법은 연, 월, 일, 시, 분, 초를 따로 나누는 것입니다. 가령 2011-01-01 00:00:00은 2011년 1월 1일 0시 0분 0초라고 볼 수 있으므로, 2011, 1, 1, 0, 0, 0으로 따로 나누면 총 6개의 숫자가 됩니다. 즉, **날짜(datetime) 컬럼을 여섯개의 다른 컬럼으로 나누어주는 것이 날짜 컬럼을 전처리하는 핵심입니다**.

In [4]:

```
# train 데이터에 연, 월, 일, 시, 분, 초를 나타내는 새로운 컬럼을 생성합니다.
# 각각의 이름을 datetime-year/month/day/hour/minute/second라고 가정합니다.
# 이 컬럼에 날짜(datetime) 컬럼의 dt(datetime의 약자입니다) 옵션을 활용하여 연월일시분초를 따로 넣
어줍니다.
train["datetime-year"] = train["datetime"].dt.year
train["datetime-month"] = train["datetime"].dt.month
train["datetime-day"] = train["datetime"].dt.day
train["datetime-hour"] = train["datetime"].dt.hour
train["datetime-minute"] = train["datetime"].dt.minute
train["datetime-second"] = train["datetime"].dt.second

# dayofweek는 날짜에서 요일(월~일)을 가져오는 기능입니다.
# 값은 0(월), 1(화), 2(수), 3(목), 4(금), 5(토), 6(일) 을 나타냅니다.
train["datetime-dayofweek"] = train["datetime"].dt.dayofweek

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# .head()로 train 데이터의 상위 5개를 띄우되,
# datetime과 이와 연관된 나머지 일곱 개의 컬럼만을 출력합니다.
train[["datetime", "datetime-year", "datetime-month", "datetime-day", "datetime-
hour", "datetime-minute", "datetime-second", "datetime-dayofweek"]].head()
```

```
(10886, 19)
```

Out[4]:

	datetime	datetime-year	datetime-month	datetime-day	datetime-hour	datetime-minute	datetime-second	datetir dayofwe
0	2011-01-01 00:00:00	2011	1	1	0	0	0	5
1	2011-01-01 01:00:00	2011	1	1	1	0	0	5
2	2011-01-01 02:00:00	2011	1	1	2	0	0	5
3	2011-01-01 03:00:00	2011	1	1	3	0	0	5
4	2011-01-01 04:00:00	2011	1	1	4	0	0	5

In [5]:

```
# datetime-dayofweek를 사람이 이해하기 쉬운 표현으로 변경합니다. (Monday ~ Sunday)
# 이를 datetime-dayofweek(humanized)라는 새로운 컬럼에 추가합니다.
train.loc[train["datetime-dayofweek"] == 0, "datetime-dayofweek(humanized)"] =
"Monday"
train.loc[train["datetime-dayofweek"] == 1, "datetime-dayofweek(humanized)"] =
"Tuesday"
train.loc[train["datetime-dayofweek"] == 2, "datetime-dayofweek(humanized)"] =
"Wednesday"
train.loc[train["datetime-dayofweek"] == 3, "datetime-dayofweek(humanized)"] =
"Thursday"
train.loc[train["datetime-dayofweek"] == 4, "datetime-dayofweek(humanized)"] =
"Friday"
train.loc[train["datetime-dayofweek"] == 5, "datetime-dayofweek(humanized)"] =
"Saturday"
train.loc[train["datetime-dayofweek"] == 6, "datetime-dayofweek(humanized)"] =
"Sunday"

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# .head()로 train 데이터의 상위 5개를 띄우되,
# datetime과 datetime-dayofweek, 그리고 datetime-dayofweek(humanized) 컬럼만을 출력합니
다.
train[["datetime", "datetime-dayofweek", "datetime-dayofweek(humanized)"]].head
()
```

```
(10886, 20)
```

Out[5]:

	datetime	datetime-dayofweek	datetime-dayofweek(humanized)
0	2011-01-01 00:00:00	5	Saturday
1	2011-01-01 01:00:00	5	Saturday
2	2011-01-01 02:00:00	5	Saturday
3	2011-01-01 03:00:00	5	Saturday
4	2011-01-01 04:00:00	5	Saturday

In [6]:

```
# test 데이터와 train 데이터와 동일하게 연, 월, 일, 시, 분, 초 컬럼을 생성합니다.
test["datetime-year"] = test["datetime"].dt.year
test["datetime-month"] = test["datetime"].dt.month
test["datetime-day"] = test["datetime"].dt.day
test["datetime-hour"] = test["datetime"].dt.hour
test["datetime-minute"] = test["datetime"].dt.minute
test["datetime-second"] = test["datetime"].dt.second

# dayofweek 컬럼도 train 데이터와 동일하게 생성합니다.
test["datetime-dayofweek"] = test["datetime"].dt.dayofweek

# test 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(test.shape)

# .head()로 test 데이터의 상위 5개를 띄우되,
# datetime과 이와 연관된 나머지 일곱 개의 컬럼만을 출력합니다.
test[["datetime", "datetime-year", "datetime-month", "datetime-day", "datetime-h
our", "datetime-minute", "datetime-second", "datetime-dayofweek"]].head()
```

```
(6493, 16)
```

Out[6]:

	datetime	datetime-year	datetime-month	datetime-day	datetime-hour	datetime-minute	datetime-second	datetir dayofwe
0	2011-01-20 00:00:00	2011	1	20	0	0	0	3
1	2011-01-20 01:00:00	2011	1	20	1	0	0	3
2	2011-01-20 02:00:00	2011	1	20	2	0	0	3
3	2011-01-20 03:00:00	2011	1	20	3	0	0	3
4	2011-01-20 04:00:00	2011	1	20	4	0	0	3

In [7]:

```
# datetime-dayofweek를 사람이 이해하기 쉬운 표현으로 변경합니다. (Monday ~ Sunday)
# 이를 datetime-dayofweek(humanized)라는 새로운 컬럼에 추가합니다.
test.loc[test["datetime-dayofweek"] == 0, "datetime-dayofweek(humanized)"] = "Mo
nday"
test.loc[test["datetime-dayofweek"] == 1, "datetime-dayofweek(humanized)"] = "Tu
esday"
test.loc[test["datetime-dayofweek"] == 2, "datetime-dayofweek(humanized)"] = "We
dnesday"
test.loc[test["datetime-dayofweek"] == 3, "datetime-dayofweek(humanized)"] = "Th
ursday"
test.loc[test["datetime-dayofweek"] == 4, "datetime-dayofweek(humanized)"] = "Fr
iday"
test.loc[test["datetime-dayofweek"] == 5, "datetime-dayofweek(humanized)"] = "Sa
turday"
test.loc[test["datetime-dayofweek"] == 6, "datetime-dayofweek(humanized)"] = "Su
nday"

# test 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(test.shape)

# .head()로 test 데이터의 상위 5개를 띄우되,
# datetime과 datetime-dayofweek, 그리고 datetime-dayofweek(humanized) 컬럼만을 출력합니
다.
test[["datetime", "datetime-dayofweek", "datetime-dayofweek(humanized)"]].head()
```

(6493, 17)

Out[7]:

	datetime	datetime-dayofweek	datetime-dayofweek(humanized)
0	2011-01-20 00:00:00	3	Thursday
1	2011-01-20 01:00:00	3	Thursday
2	2011-01-20 02:00:00	3	Thursday
3	2011-01-20 03:00:00	3	Thursday
4	2011-01-20 04:00:00	3	Thursday

Explore

전처리(Preprocesing)를 끝냈으면 그 다음에는 데이터를 분석해보겠습니다.

주어진 데이터를 시각화나 분석 툴을 통해 다양한 관점에서 이해하는 과정을 탐험적 데이터 분석(Exploratory Data Analysis (https://en.wikipedia.org/wiki/Exploratory_data_analysis))이라고 합니다. 저번 타이타닉 문제와 마찬가지로, 이번에도 파이썬의 데이터 시각화 패키지인 (matplotlib (https://matplotlib.org))와 seaborn (https://seaborn.pydata.org/) 을 활용해서 분석해보겠습니다.

In [8]:

```
# matplotlib로 실행하는 모든 시각화를 자동으로 쥬피터 노트북에 띄웁니다.
# seaborn 도 결국에는 matplotlib를 기반으로 동작하기 때문에, seaborn으로 실행하는 모든 시각화도
 마찬가지로 쥬피터 노트북에 자동적으로 띄워집니다.
%matplotlib inline

# 데이터 시각화 패키지 seaborn을 로딩합니다. 앞으로는 줄여서 sns라고 사용할 것입니다.
import seaborn as sns

# 데이터 시각화 패키지 matplotlib를 로딩합니다. 앞으로는 줄여서 plt라고 사용할 것입니다.
import matplotlib.pyplot as plt
```

datetime

먼저 분석할 컬럼은 **날짜(datetime)** 컬럼입니다. 날짜 컬럼은 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 경진대회의 핵심 컬럼이라고 볼 수 있으며, 이번 경진대회에서 상위 성적을 올리고 싶다면 날짜 컬럼을 완벽하게 이해하는 것이 무엇보다도 중요합니다.

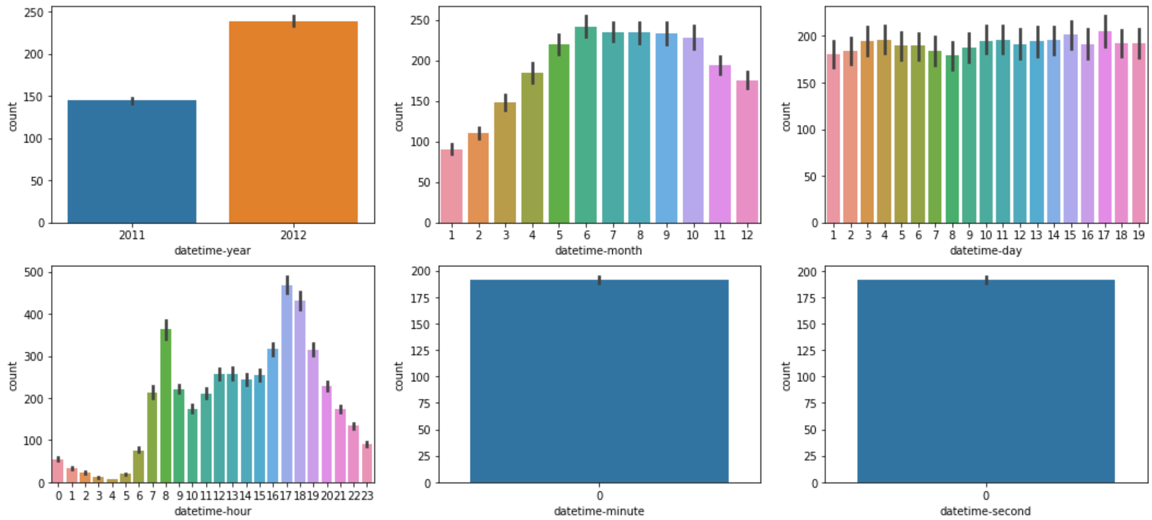
먼저 연/월/일/시/분/초에 따른 자전거 대여량을 시각화 해보겠습니다.

In [9]:

```
# matplotlib의 subplots를 사용합니다. 이 함수는 여러 개의 시각화를 한 화면에 띄울 수 있도록 합니
다.
# 이번에는 2x3으로 총 6개의 시각화를 한 화면에 띄웁니다.
figure, ((ax1, ax2, ax3), (ax4, ax5, ax6)) = plt.subplots(nrows=2, ncols=3)

# 시각화의 전체 사이즈는 18x8로 설정합니다.
figure.set_size_inches(18, 8)

# seaborn의 barplot으로 subplots의 각 구역에
# 연, 월, 일, 시, 분, 초 별 자전거 대여량을 출력합니다.
sns.barplot(data=train, x="datetime-year", y="count", ax=ax1)
sns.barplot(data=train, x="datetime-month", y="count", ax=ax2)
sns.barplot(data=train, x="datetime-day", y="count", ax=ax3)
sns.barplot(data=train, x="datetime-hour", y="count", ax=ax4)
sns.barplot(data=train, x="datetime-minute", y="count", ax=ax5)
sns.barplot(data=train, x="datetime-second", y="count", ax=ax6)
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3902e753c8>
```



위 그림에서 알 수 있는 내용은 다음과 같습니다.

datetime-year

- 2011년도의 자전거 대여량보다 2012년도의 자전거 대여량이 더 높습니다. 이는 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 경진대회를 주최한 Capital Bikeshare (https://www.capitalbikeshare.com/)사가 꾸준히 성장하고 있다고 간주할 수 있습니다.

datetime-month

- 주로 여름(6~8월)에 자전거를 많이 빌리며, 겨울(12~2월)에는 자전거를 많이 빌리지 않습니다.
- 같은 겨울이라도 12월의 자전거 대여량이 1월의 자전거 대여량보다 두 배 가까이 높아 보입니다. 하지만 여기에는 숨겨진 비밀이 있는데, 다음에 나올 다른 시각화에서 자세히 살펴보겠습니다.

datetime-day

- x축을 자세히 보면 1일부터 19일까지밖에 없습니다. 20일은 어디에 있을까요? 바로 test 데이터에 있습니다. 이 시각화에서 알 수 있는 내용은, train 데이터와 test 데이터를 나누는 기준이 되는 컬럼이 바로 **datetime-day**라는 것입니다.
- 이런 경우 **datetime-day**를 feature로 집어넣으면 머신러닝 알고리즘이 과적합(overfitting (https://hyperdot.wordpress.com/2017/02/06/%EA%B3%BC%EC%A0%81%ED%95%A9overfitting/)) 되는 현상이 일어날 수 있습니다. **그러므로 train 데이터와 test 데이터를 나누는 기준이 되는 컬럼이 있으면, 이 컬럼은 feature로 사용하지 않는 것이 좋습니다.**

datetime-hour

- 새벽 시간에는 사람들이 자전거를 빌리지 않으며, 오후 시간에 상대적으로 자전거를 많이 빌립니다.
- 특이하게도 두 부분에서 사람들이 자전거를 특별히 많이 빌리는 현상이 있습니다. 바로 출근 시간(7~9시)과 퇴근 시간(16시~19시) 입니다.
- 물론 출퇴근시간이 아닌 다른 시간대에 자전거를 빌리는 경우도 존재합니다. 이는 다음에 나올 다른 시각화에서 자세히 살펴보겠습니다.

datetime-minute & **datetime-second**

- 이 두 컬럼은 x축이 모두 0으로 되어있습니다. 즉, **datetime-minute**과 **datetime-second**은 기록되고 있지 않다는 사실을 알 수 있습니다. 이 경우에는 feature로 넣어도 큰 의미가 없기 때문에 사용하지 않습니다.

그러므로 이 시각화에서 알 수 있는 결론은, **전체 여섯개의 컬럼 중 datetime-year와 datetime-month, 그리고 datetime-hour만 사용하는 것이 가장 좋다는 사실을 깨달을 수 있습니다.**

datetime-year & datetime-month

다음에는 연-월을 붙여서 시각화해보겠습니다.

이전에는 연/월을 따로 시각화해서 출력하였지만, 이번에는 연-월을 붙여서 2011년 1월부터 2012년 12월까지 총 24개의 경우의 수를 x축으로 놓고 시각화해보고 싶습니다. 먼저 이를 시각화하기에 필요한 **datetime-year_month**라는 새로운 컬럼을 만들어 보겠습니다.

In [10]:

```
# datetime-year와 datetime-month의 형태를 변환합니다.
# 이전까지는 정수형(int)였지만, pandas의 astype을 통해 문자열(str)로 변환합니다.
# 이 결과를 datetime-year(str)와 datetime-month(str)라는 새로운 컬럼에 집어넣습니다.
train["datetime-year(str)"] = train["datetime-year"].astype('str')
train["datetime-month(str)"] = train["datetime-month"].astype('str')

# datetime-year(str)와 datetime-month(str) 문자열 두 개를 붙여서 datetime-year_month라
는 새로운 컬럼을 추가합니다.
# 이 컬럼에는 2011-1부터 2012-12까지의 총 24의 경우의 수가 들어갑니다.
train["datetime-year_month"] = train["datetime-year(str)"] + "-" + train["dateti
me-month(str)"]

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# .head() train 데이터의 상위 5개를 띄우되,
# datetime과 datetime-year_month 두 개의 컬럼만 출력합니다.
train[["datetime", "datetime-year_month"]].head()
```

```
(10886, 23)
```

Out[10]:

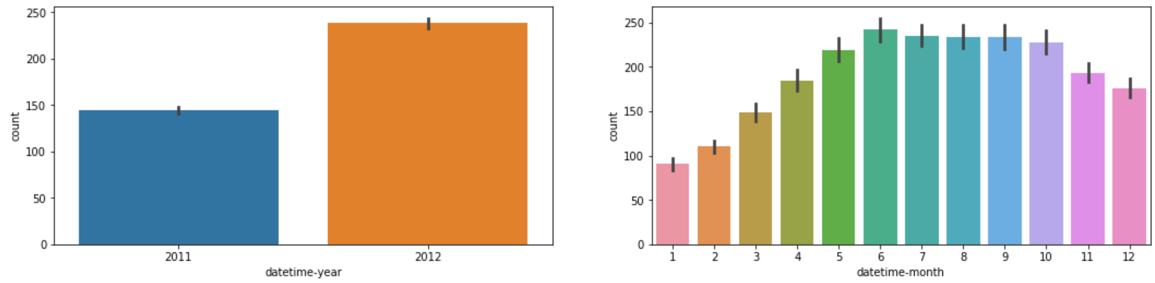
	datetime	datetime-year_month
0	2011-01-01 00:00:00	2011-1
1	2011-01-01 01:00:00	2011-1
2	2011-01-01 02:00:00	2011-1
3	2011-01-01 03:00:00	2011-1
4	2011-01-01 04:00:00	2011-1

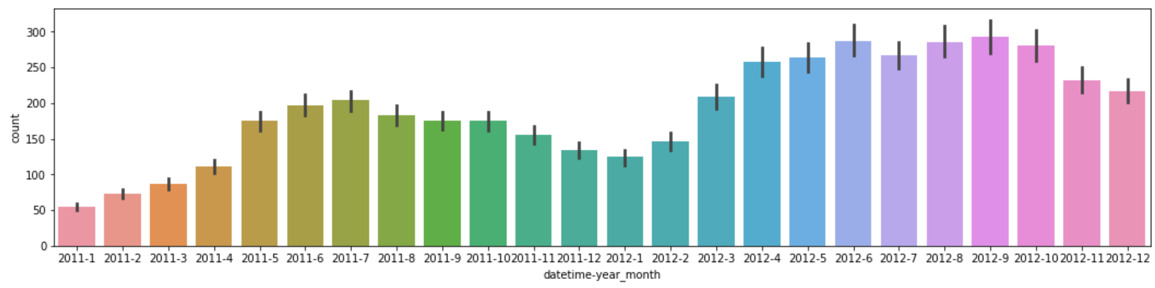
In [11]:

```
# matplotlib의 subplots를 사용합니다. 이 함수는 여러 개의 시각화를 한 화면에 띄울 수 있도록 합니
다.
# 이번에는 1x2로 총 2개의 시각화를 한 화면에 띄웁니다.
figure, (ax1, ax2) = plt.subplots(nrows=1, ncols=2)

# 시각화의 전체 사이즈는 18x4로 설정합니다.
figure.set_size_inches(18, 4)

# seaborn의 barplot으로 subplots의 각 구역에
# 연, 월별 자전거 대여량을 출력합니다.
sns.barplot(data=train, x="datetime-year", y="count", ax=ax1)
sns.barplot(data=train, x="datetime-month", y="count", ax=ax2)

# 다시 한 번 matplotlib의 subplots를 사용합니다.
# 이번에는 1x1로 1개의 시각화만을 출력합니다.
figure, ax3 = plt.subplots(nrows=1, ncols=1)

# 이 시각화의 전체 사이즈는 18x4로 설정합니다.
figure.set_size_inches(18, 4)

# 이번에는 seaborn의 barplot으로 연-월을 붙여서 자전거 대여량을 출력합니다.
sns.barplot(data=train, x="datetime-year_month", y="count", ax=ax3)
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f39024c5e80>
```





위 그림에서 알 수 있는 내용은 다음과 같습니다.

* Capital Bikeshare (https://www.capitalbikeshare.com/)사의 자전거 대여량은 꾸준히 상승하고 있는 추세입니다.
* 우상단 시각화를 보자면, 12월의 자전거 대여량이 1월의 자전거 대여량보다 두 배 가까이 높습니다.
* 하지만 아래의 시각화를 보면, 2011년 12월의 자전거 대여량과 2012년 1월의 자전거 대여량이 큰 차이가 없다는 사실을 발견할 수 있습니다.
* 반면에 2011년 1월의 자전거 대여량과 2012년 12월의 자전거 대여량은 큰 차이가 나는 것을 알 수 있습니다.

즉, 12월이 1월에 비해 자전거 대여량이 두 배 가까이 높은 이유는, 1) Capital Bikeshare (https://www.capitalbikeshare.com/)의 자전거 대여량이 꾸준히 상승하고 있는 추세이며, 2) 이 과정에서 시기상으로 12월이 1월부터 늦게 발생했기 때문입니다. 즉 **자전거를 대여하는 고객 입장에서 12월이라고 자전거를 더 많이 빌려야 할 이유는 없습니다.**

이 점 역시 머신러닝 알고리즘이 과적합(overfitting)될 소지가 다분합니다. 이를 해결할 수 있는 다양한 방법이 있는데,

* **datetime-year_month**를 통채로 One Hot Encoding해서 feature로 사용한다.
* 자전거 대여량이 꾸준히 성장하는 추세에 맞춰서 count를 보정한다.

하지만 제 경험상, 가장 쉽과 빠르게 머신러닝 모델의 정확도를 늘리는 방법은 **datetime-month**를 feature로 사용하지 않는 것입니다. 그러므로 **연/월/일/시/분/초 여섯 개의 컬럼 중 연도(datetime-year)와 시간(datetime-hour), 이렇게 두 개의 컬럼만 사용하도록 하겠습니다.**

datetime-hour

다음에는 **datetime-hour** 컬럼을 분석해보겠습니다.

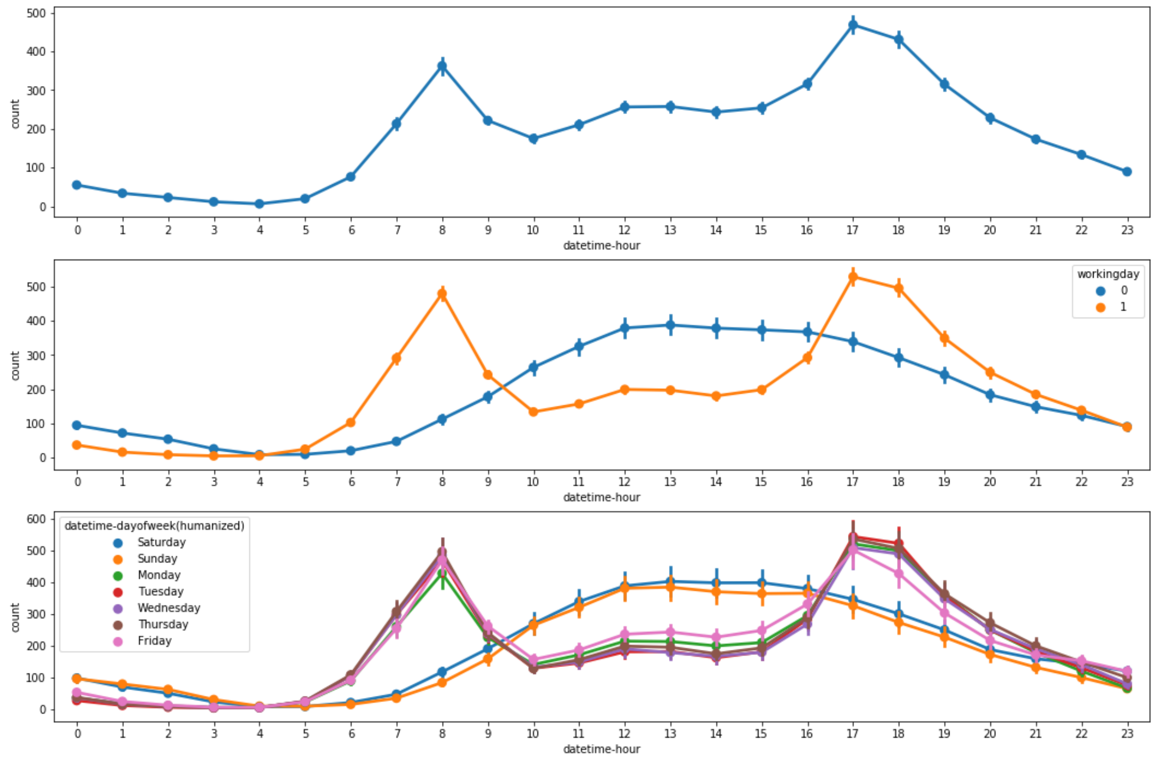
이번에는 **datetime-hour** 컬럼 외에도 두 개의 컬럼을 추가로 분석하겠습니다. 바로 근무일(workingday)와 요일(datetime-dayofweek)입니다.

In [12]:

```
# matplotlib의 subplots를 사용합니다. 이 함수는 여러 개의 시각화를 한 화면에 띄울 수 있도록 합니다.
# 이번에는 3x1로 총 3개의 시각화를 한 화면에 띄웁니다.
figure, (ax1, ax2, ax3) = plt.subplots(nrows=3, ncols=1)

# 시각화의 전체 사이즈는 18x12로 설정합니다.
figure.set_size_inches(18, 12)

# seaborn의 pointplot으로 시각당 자전거 대여량을 시각화합니다.
sns.pointplot(data=train, x="datetime-hour", y="count", ax=ax1)

# 비슷하게 seaborn의 pointplot으로 시각당 자전거 대여량을 시각화합니다.
# 하지만 이번에는 근무일(workingday)에 따른 차이를 보여줍니다.
sns.pointplot(data=train, x="datetime-hour", y="count", hue="workingday", ax=ax2)

# 비슷하게 seaborn의 pointplot으로 시각당 자전거 대여량을 시각화합니다.
# 하지만 이번에는 요일(datetime-dayofweek)에 따른 차이를 보여줍니다.
sns.pointplot(data=train, x="datetime-hour", y="count", hue="datetime-dayofweek(humanized)", ax=ax3)
```

Out[12]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f39027ca5c0>



위 그림에서 알 수 있는 내용은 다음과 같습니다.

- 사람들은 기본적으로 출근 시간(7~9시)과 퇴근 시간(16~19시)에 자전거를 많이 빌립니다.
- 하지만 이는 근무일일 경우(workingday == 1)에만 한정된 이야기입니다. 근무일이 아닐 경우(workingday == 0), 사람들은 출/퇴근시간에 자전거를 빌리지 않고, 오후 시간(10 ~ 16시)에 자전거를 많이 빌리는 것을 확인할 수 있습니다.

이번에는 **요일(datetime-dayofweek)**별 자전거 대여량을 살펴보겠습니다.

- 먼저 금요일을 살펴보면, 다른 주중(월~목)에 비해 퇴근 시간(17~19시)에 상대적으로 자전거를 덜 빌리는 사실을 알 수 있습니다. 이는 추측컨데 모종의 이유로 자전거를 탈 수 없거나(ex: 음주), 다른 교통수단(ex: 버스, 택시)을 대신 사용했다는 것을 알 수 있습니다.
- 반면 금요일은 주중임에도 불구하고 상대적으로 오후 시간(10~16시)의 자전거 대여량이 높은 것을 알 수 있습니다. 그 다음으로 높은 주중은 바로 월요일입니다. 즉, 금요일과 월요일은 주중임에도 불구하고 어느정도 주말의 속성을 가지고 있다는 사실을 알 수 있습니다.
- 이번에는 주말을 살펴보겠습니다. 일요일을 보자면, 토요일에 비해 상대적으로 자전거 대여량이 낮다는 사실을 알 수 있습니다. 이는 추측컨데 월요일의 피로도를 고려해서 토요일에 비해 대외 활동을 덜 가지는 것으로 생각할 수 있습니다.

이 분석을 통해 알 수 있는 사실은, 요일(datetime-dayofweek)을 머신러닝 알고리즘에 feature로 집어넣으면 근무일(workingday)만 집어넣는 것에 비해 더 좋은 성능을 낼 수 있다고 볼 수 있습니다. 그러므로 **요일(datetime-dayofweek)** 컬럼을 feature로 추가하겠습니다.

count

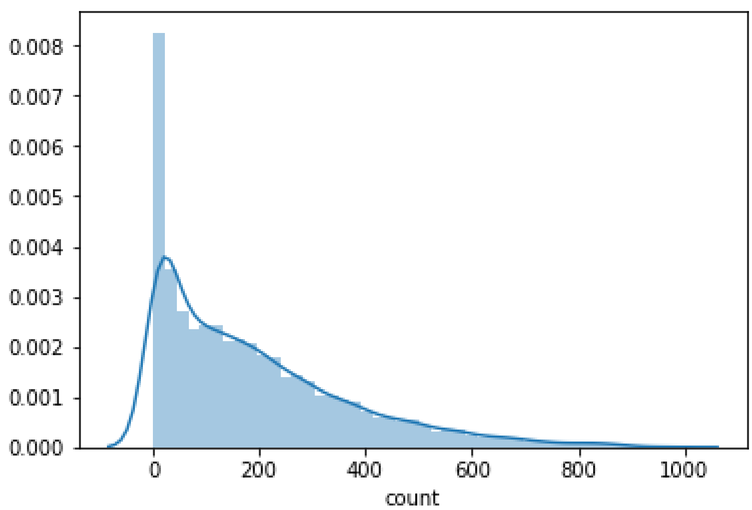
다음으로 분석할 컬럼은 **자전거 대여량(count)**입니다. 자전거 대여량(count)은 최종적으로 우리가 맞춰야 할 label이기 때문에, 다른 컬럼보다도 이 컬럼을 완벽하게 이해하는 것이 중요합니다. 먼저 seaborn의 [distplot (https://seaborn.pydata.org/generated/seaborn.distplot.html)](https://seaborn.pydata.org/generated/seaborn.distplot.html)으로 이 컬럼을 시각화 해보겠습니다.

In [13]:

```
# 자전거 대여량(count)의 분포를 시각화합니다.
sns.distplot(train["count"])
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f390203ebe0>
```



위 그림에서 알 수 있는 내용은 다음과 같습니다.

- 자전거 대여량이 1 ~ 20대인 비중이 굉장히 높습니다.
- 반면에 자전거 대여량이 1,000대에 근접하는 경우도 있습니다. (977대)

위 두 개의 특성이 데이터를 왜곡되게(skewed) 만드는 것 같습니다. 이러한 경우 log transformation (http://onlinestatbook.com/2/transformations/log.html)을 시도해 볼 만 합니다.

In [14]:

```
# numpy라는 패키지를 불러옵니다.
# 이 패키지는 선형대수(linear algebra) 패키지라고 불리는데,
# 현재는 간단하게 '수학 연산을 편하게 해주는 패키지'라고 이해하시면 됩니다.
import numpy as np

# 자전거 대여량(count)에 +1을 한 후 log를 적용합니다.
# 이를 log transformation이라고 합니다.
train["log_count"] = np.log(train["count"] + 1)

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# .head()로 train 데이터의 상위 5개를 띄우되,
# count와 log_count 컬럼만 출력합니다.
train[["count", "log_count"]].head()
```

(10886, 24)

Out[14]:

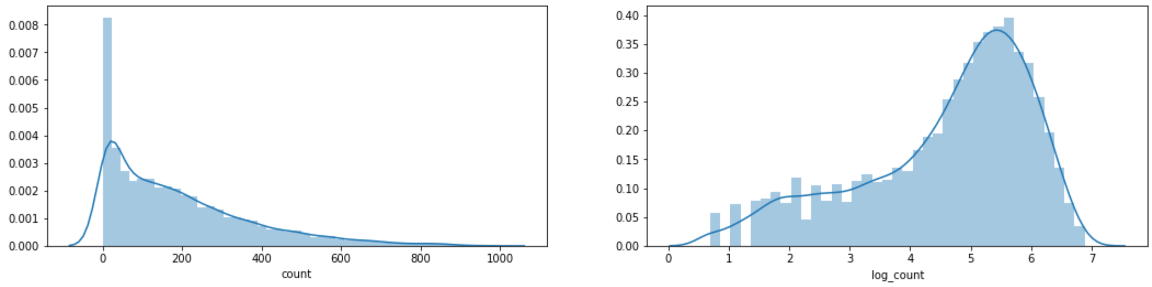
	count	log_count
0	16	2.833213
1	40	3.713572
2	32	3.496508
3	13	2.639057
4	1	0.693147

In [15]:

```
# matplotlib의 subplots를 사용합니다. 이 함수는 여러 개의 시각화를 한 화면에 띄울 수 있도록 합니
다.
# 이번에는 1x2로 총 2개의 시각화를 한 화면에 띄웁니다.
figure, (ax1, ax2) = plt.subplots(nrows=1, ncols=2)

# 시각화의 전체 사이즈는 18x4로 설정합니다.
figure.set_size_inches(18, 4)

# 좌측에는 자전거 대여량(count)의 분포를 시각화합니다.
sns.distplot(train["count"], ax=ax1)

# 우측에는 log transformation한 자전거 대여량(log_count)의 분포를 시각화합니다.
sns.distplot(train["log_count"], ax=ax2)
```

Out[15]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f3902481b00>



비교 결과 좌측보다 우측이 훨씬 더 자연스럽습니다. (=정규 분포 (https://ko.wikipedia.org/wiki/%EC%A0%95%EA%B7%9C_%EB%B6%84%ED%8F%AC)에 가깝게 나옵니다.) 그 즉슨, **자전거 대여량(count)을 그대로 사용하는 것 보다, 이를 log transformation한 버전(log_count)을 사용하는게 더 좋은 정확도를 낼 수 있다고 가정할 수 있습니다.**

다만 이 경우, 캐글에 제출하기 위해서는 log transformation한 버전을 원상복귀 시켜줘야 합니다. 이 경우에는 exp (https://en.wikipedia.org/wiki/Exponential_function)를 사용합니다.

In [16]:

```
# log transformation한 자전거 대여량(log_count)을 다시 exp로 원상복귀 합니다.
# (=자연로그는 exp로 없애버릴 수 있기 때문입니다)
# 이를 count(recover)라는 새로운 컬럼에 대입합니다.
train["count(recover)"] = np.exp(train["log_count"]) - 1

# train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(train.shape)

# .head()로 train 데이터의 상위 5개를 띄우되,
# count, log_count, 그리고 count(recover) 컬럼만 출력합니다.
train[["count", "log_count", "count(recover)"]].head()
```

```
(10886, 25)
```

Out[16]:

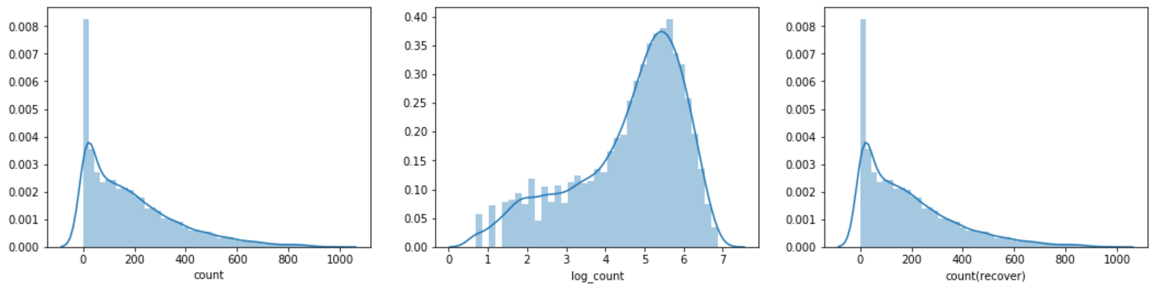
	count	log_count	count(recover)
0	16	2.833213	16.0
1	40	3.713572	40.0
2	32	3.496508	32.0
3	13	2.639057	13.0
4	1	0.693147	1.0

In [17]:

```
# matplotlib의 subplots를 사용합니다. 이 함수는 여러 개의 시각화를 한 화면에 띄울 수 있도록 합니
다.
# 이번에는 1x3로 총 3개의 시각화를 한 화면에 띄웁니다.
figure, (ax1, ax2, ax3) = plt.subplots(nrows=1, ncols=3)

# 시각화의 전체 사이즈는 18x4로 설정합니다.
figure.set_size_inches(18, 4)

# 좌측에는 자전거 대여량(count)의 분포를 시각화합니다.
sns.distplot(train["count"], ax=ax1)

# 가운데에는 log transformation한 자전거 대여량(log_count)의 분포를 시각화합니다.
sns.distplot(train["log_count"], ax=ax2)

# 우측에는 log transformation을 다시 원상복귀한 버전(count(recover))의 분포를 시각화합니다.
sns.distplot(train["count(recover)"], ax=ax3)
```

Out[17]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f3901b70940>



Train

이제 분석을 통해 발견한 인사이트를 활용해보겠습니다.

이전 경진대회와 마찬가지로, 이번에도 머신러닝 알고리즘을 사용하겠습니다. 이번에도 변함없이 [지도학습(Supervised Learning) (http://solarisailab.com/archives/1785)](http://solarisailab.com/archives/1785) 알고리즘을 사용할 계획이기 때문에, 데이터를 Label(맞춰야 하는 정답)과 Feature(Label을 맞추는데 도움이 되는 값들)로 나눌 필요가 있습니다.

이번 경진대회에서는 다음의 컬럼들을 Feature와 Label로 활용할 것입니다.

- **Feature**: 1) 계절(season), 2) 공휴일(holiday), 3) 근무일(workingday), 4) 날씨(weather), 5) 온도(temp), 6) 체감 온도(atemp), 7) 습도(humidity), 8) 풍속(weather), 9) 연도(datetime-year), 10) 시간(datetime-hour), 마지막으로 11) 요일(datetime-dayofweek) 입니다.
- **Label**: log transformation한 자전거 대여량(log_count)을 사용합니다.

이를 통해 train 데이터와 test 데이터를 다음의 세 가지 형태의 값으로 나눌 것입니다.

- **X_train**: train 데이터의 feature 입니다. 줄여서 X_train이라고 부릅니다.
- **X_test**: test 데이터의 feature 입니다. 마찬가지로 줄여서 X_test라고 부릅니다.
- **y_train**: train 데이터의 label 입니다. 마찬가지로 줄여서 y_train이라고 부릅니다.

In [18]:

```
# 총 11개의 컬럼을 feature를 지정합니다.
# 이 11개의 컬럼명을 feature_names라는 이름의 파이썬 리스트(list)로 만들어 변수에 할당합니다.
feature_names = ["season", "holiday", "workingday", "weather",
                 "temp", "atemp", "humidity", "windspeed",
                 "datetime-year", "datetime-hour", "datetime-dayofweek"]
feature_names
```

Out[18]:

```
['season',
 'holiday',
 'workingday',
 'weather',
 'temp',
 'atemp',
 'humidity',
 'windspeed',
 'datetime-year',
 'datetime-hour',
 'datetime-dayofweek']
```

In [19]:

```
# log transformation한 자전거 대여량(log_count)을 label로 지정합니다.
label_name = "log_count"
label_name
```

Out[19]:

```
'log_count'
```

In [20]:

```
# feature_names를 활용해 train 데이터의 feature를 가져옵니다.
# 이를 X_train이라는 이름의 변수에 할당합니다.
X_train = train[feature_names]

# X_train 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(X_train.shape)

# X_train 데이터의 상위 5개를 띄웁니다.
X_train.head()
```

(10886, 11)

Out[20]:

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	datetim yea
0	1	0	0	1	9.84	14.395	81	0.0	2011
1	1	0	0	1	9.02	13.635	80	0.0	2011
2	1	0	0	1	9.02	13.635	80	0.0	2011
3	1	0	0	1	9.84	14.395	75	0.0	2011
4	1	0	0	1	9.84	14.395	75	0.0	2011

In [21]:

```
# feature_names를 활용해 test 데이터의 feature를 가져옵니다.
# 이를 X_test라는 이름의 변수에 할당합니다.
X_test = test[feature_names]

# X_test 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(X_test.shape)

# X_test 데이터의 상위 5개를 띄웁니다.
X_test.head()
```

(6493, 11)

Out[21]:

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	dateti
0	1	0	1	1	10.66	11.365	56	26.0027	2011
1	1	0	1	1	10.66	13.635	56	0.0000	2011
2	1	0	1	1	10.66	13.635	56	0.0000	2011
3	1	0	1	1	10.66	12.880	56	11.0014	2011
4	1	0	1	1	10.66	12.880	56	11.0014	2011

In [22]:

```
# label_name을 활용해 train 데이터의 label을 가져옵니다.
# 이를 y_train이라는 이름의 변수에 할당합니다.
y_train = train[label_name]

# y_train 변수에 할당된 데이터의 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시되나, column이 없기 때문에 (row,) 형태로 표시될 것입니다.
print(y_train.shape)

# y_train 데이터의 상위 5개를 띄웁니다.
y_train.head()
```

```
(10886,)
```

Out[22]:

```
0    2.833213
1    3.713572
2    3.496508
3    2.639057
4    0.693147
Name: log_count, dtype: float64
```

Evaluate

머신러닝 모델을 학습시키기 전에, 측정 공식(Evaluation Metric)을 통해 학습한 모델의 성능이 얼마나 뛰어난지 정량적으로 측정해보겠습니다. 이번 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 경진대회에서 사용하는 측정 공식은 Root Mean Squared Logarithmic Error (RMSLE (https://www.kaggle.com/c/bike-sharing-demand#evaluation)) 입니다.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

이 공식은 정답(a_i, actual)과 예측값(p_i, predict)의 차이가 크면 클수록 **페널티를 덜 주는** 방식으로 동작합니다. (`log(count + 1)`이 그 역할을 합니다)

다만 현재 이미 log transformation한 count(log_count)을 사용하고 있기 때문에, 이 공식을 그대로 사용할 경우 **사실상 `log(count + 1)`를 두 번 하게 되는 셈이 됩니다.** 이를 방지하기 위해, 측정 공식에서 `log(count + 1)`을 제거하도록 하겠습니다.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2}$$

이 공식을 Root Mean Squared Error (RMSE (https://en.wikipedia.org/wiki/Root-mean-square_deviation))라고 합니다. 파이썬과 numpy (http://www.numpy.org/), scikit-learn (http://scikit-learn.org/stable/)으로 **RMSE** 공식을 구현해보겠습니다.

In [23]:

```
# numpy라는 패키지를 불러옵니다.
# 이 패키지는 선형대수(linear algebra) 패키지라고 불리는데,
# 현재는 간단하게 '수학 연산을 편하게 해주는 패키지'라고 이해하시면 됩니다.
import numpy as np

# scikit-learn 패키지의 metrics 모듈에서 make_scorer라는 함수를 가지고 옵니다.
# 이 함수는 파이썬을 구현한 측정 공식을 scikit-learn에서 사용할 수 있도록 변환해 줍니다.
from sklearn.metrics import make_scorer

# RMSE 공식을 구현한 함수를 생성합니다.
# 이 함수는 예측값(predict)과 정답(actual)을 인자로 받습니다.
def rmse(predict, actual):
    # predict와 actual을 numpy array로 변환합니다.
    # 이렇게 하면 수학 연산을 편하게 할 수 있습니다.
    predict = np.array(predict)
    actual = np.array(actual)

    # 공식에 쓰여진대로 predict와 actual을 빼서 차이를 구합니다.
    # 이 차이를 distance라는 이름의 새로운 변수에 할당합니다.
    distance = predict - actual

    # 공식에 쓰여진대로 distance를 제곱합니다.
    # 이 결과를 square_distance라는 이름의 새로운 변수에 할당합니다.
    square_distance = distance ** 2

    # 공식에 쓰여진대로 square_distance의 평균을 구합니다.
    # 이 결과를 mean_square_distance라는 이름의 새로운 변수에 할당합니다.
    mean_square_distance = square_distance.mean()

    # 공식에 쓰여진대로 mean_square_distance에 루트(sqrt)를 씌웁니다.
    # 이 결과를 score라는 이름의 새로운 변수에 할당합니다.
    score = np.sqrt(mean_square_distance)

    # score 변수를 반환합니다.
    return score

# scikit-learn의 make_scorer를 활용하여
# rmse 함수를 scikit-learn의 다른 함수에서 사용할 수 있도록 변환합니다.
# 이 결과를 rmse_score라는 이름의 새로운 변수에 할당합니다.
rmse_score = make_scorer(rmse)
rmse_score
```

Out[23]:

```
make_scorer(rmse)
```

Hyperparameter Tuning

이번에는 머신러닝 모델의 하이퍼패러미터를 튜닝해보겠습니다.

머신러닝 모델에는 다양한 옵션이 있는데, 이 옵션을 통해 모델의 성능을 끌어올릴 수 있습니다. 이 옵션들을 전문용어로 하이퍼패러미터(Hyperparameter)라고 부릅니다. 만일 적절한 하이퍼패러미터를 찾아서 모델에 적용할 수 있다면 모델의 성능을 한 층 더 끌어올릴 수 있습니다. 이를 **하이퍼패러미터 튜닝(Hyperparamter Tuning)**이라고 합니다.

In [53]:

```
# Gradient Boosting Machine 패키지인 XGBoost를 가져옵니다.
# 이를 xgb라는 축약어로 사용합니다.
import xgboost as xgb

# XGBRegressor를 생성합니다.
# 생성한 모델을 출력하면 다양한 하이퍼패러미터(n_estimators, max_depth, etc)들이 있는 것을 확인할 수 있습니다.
xgb.XGBRegressor()
```

Out[53]:

```
XGBRegressor(base_score=0.5, colsample_bylevel=1, colsample_bytree=
1, gamma=0,
       learning_rate=0.1, max_delta_step=0, max_depth=3,
       min_child_weight=1, missing=None, n_estimators=100, nthread=-
1,
       objective='reg:linear', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=0, silent=True, subsample=1)
```

하이어패러미터를 튜닝하는 방법은 크게 두 가지가 있습니다. 첫 번째로 Grid Search라는 방식이고, 두 번째는 Coarse & Finer Search라는 방식입니다. 먼저 Grid Search부터 살펴보겠습니다.

Case 1 - Grid Search

Grid Search는 몇 개의 하이퍼패리미터 후보군을 정한 뒤 이를 계속 조합해가며 가장 좋은 하이퍼패리미터를 찾는 방식입니다. Cross Validation 점수가 가장 좋은 하이퍼패러미터를 가장 좋은 하이퍼패러미터라고 간주합니다. (자세한 사항은 다음의 링크 (http://scikit-learn.org/stable/modules/grid_search.html)를 참고 바랍니다)

보통은 scikit-learn (http://scikit-learn.org/)의 GridSearchCV (http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)를 사용합니다만, 이번에는 Grid Search를 더 자세히 이해하기 위해 이를 직접 구현해보겠습니다.

In [25]:

```
# Gradient Boosting Machine 패키지인 XGBoost를 가져옵니다.
# 이를 xgb라는 축약어로 사용합니다.
import xgboost as xgb

# scikit-learn 패키지의 model_selection 모듈에 있는 cross_val_score 함수를 가지고 옵니다.
# 이 함수가 Cross Validation을 담당합니다.
from sklearn.model_selection import cross_val_score

# n_estimators는 트리의 갯수입니다.
# 너무 낮으면 트리의 갯수가 부족하고, 너무 높으면 트리가 과적합(Overfitting)되는 현상이 있습니다.
# 그러므로 적당한 값을 주는 것이 좋습니다.
n_estimators_list = [100, 300, 1000]

# max_depth의 후보군을 지정합니다. 10 ~ 90 사이에서 10 단위로 지정하겠습니다.
max_depth_list = [50, 75, 100]

# learning_rate의 후보군을 지정합니다. 트리마다의 비중을 나타냅니다.
# 보통은 10의 -n승 단위로 지정합니다.
learning_rate_list = [1.0, 0.1, 0.01, 0.001, 0.0001]

# subsample의 후보군을 지정합니다. 하나의 트리를 만들 때, 사용할 데이터의 비율을 나타냅니다.
# 0.5, 0.75, 1.0을 주겠습니다. (각각 50%, 75%, 100%의 데이터를 사용합니다.)
subsample_list = [0.5, 0.75, 1.0]

# colsample_bytree의 후보군을 지정합니다. 하나의 트리를 만들 때, 사용할 컬럼의 비율을 나타냅니다.
# 0.5, 0.75, 1.0을 주겠습니다. (각각 50%, 75%, 100%의 컬럼을 사용합니다.)
colsample_bytree_list = [0.4, 0.7, 1.0]

# colsample_bylevel의 후보군을 지정합니다. 나무에서 가지를 한 번 칠 때, 사용할 컬럼의 비율을 나타냅
니다.
# 0.5, 0.75, 1.0을 주겠습니다. (각각 50%, 75%, 100%의 컬럼을 사용합니다.)
colsample_bylevel_list = [0.4, 0.7, 1.0]

# hyperparameter 탐색 결과를 리스트로 저장합니다.
hyperparameters_list = []

# 모든 종류의 hyperparameter 후보군을 만듭니다.
for n_estimators in n_estimators_list:
    for max_depth in max_depth_list:
        for learning_rate in learning_rate_list:
            for subsample in subsample_list:
                for colsample_bylevel in colsample_bylevel_list:
                    for colsample_bytree in colsample_bytree_list:
                        # XGBRegressor를 생성합니다. 실행할때는 다음의 옵션이 들어갑니다.
                        # 1) n_estimators. 트리의 갯수입니다. 지정한 갯수만큼 트리를 생성합니
다.
                        # 2) max_depth. 트리의 깊이입니다. 지정한 숫자만큼 트리가 깊게 가지를
 뻗습니다.
                        # 3) learning_rate. 각 트리마다의 비중을 나타냅니다. 너무 작으면 과
적합(overfitting)될 가능성이 있고, 너무 높으면 부적합(underfitting)될 가능성이 있습니다.
                        # 4) subsample. 하나의 트리를 만들 때 사용할 데이터의 비율을 나타냅니
다. 0.0 ~ 1.0 사이의 값을 넣으면 지정한 비율만큼만 랜덤하게 데이터를 사용합니다.
                        # 5) colsample_bytree. 하나의 트리를 만들 때 사용할 feature의 비
율을 나타냅니다. 0.0 ~ 1.0 사이의 값을 넣으면 트리를 만들 때 지정한 비율만큼만 랜덤하게 feature를 사
용합니다.
                        # 6) colsample_bylevel. 트리가 한 번 가지를 칠 때 사용할 feature
의 비율을 나타냅니다. 0.0 ~ 1.0 사이의 값을 넣으면 트리가 가지를 칠 때 지정한 비율만큼만 랜덤하게 fea
ture를 사용합니다.
                        # 생성한 XGBRegressor를 model이라는 이름의 변수에 대입합니다.
```

```
                        model = xgb.XGBRegressor(n_estimators=n_estimators,
                                                 max_depth=max_depth,
                                                 learning_rate=learning_rate,
                                                 subsample=subsample,
                                                 colsample_bytree=colsample_bytr
ee,
                                                 colsample_bylevel=colsample_byl
evel,
                                                 seed=37)

                        # cross_val_score를 실행합니다. 실행할 때는 다음의 옵션이 들어갑니다.
                        # 1) model. 점수를 측정할 머신러닝 모델이 들어갑니다.
                        # 2) X_train. train 데이터의 feature 입니다.
                        # 3) y_train. train 데이터의 label 입니다.
                        # 4) cv. Cross Validation에서 데이터를 조각낼(split) 갯수입니다.
 총 20조각을 내야하기 때문에 20을 대입합니다.
                        # 5) scoring. 점수를 측정할 공식입니다. 앞서 구현한 RMSE를 적용합니
다.
                        # 마지막으로, 이 함수의 실행 결과의 평균(mean)을 구한 뒤 score라는 이
름의 새로운 변수에 할당합니다.
                        score = cross_val_score(model, X_train, y_train, cv=20,
scoring=rmse_score).mean()

                        # hyperparameter 탐색 결과를 딕셔너리화 합니다.
                        hyperparameters = {
                            'score': score,
                            'n_estimators': n_estimators,
                            'max_depth': max_depth,
                            'learning_rate': learning_rate,
                            'subsample': subsample,
                            'colsample_bylevel': colsample_bylevel,
                            'colsample_bytree': colsample_bytree,
                        }

                        # hyperparameter 탐색 결과를 리스트에 저장합니다.
                        hyperparameters_list.append(hyperparameters)

                        # hyperparameter 탐색 결과를 출력합니다.
                        print(f"n_estimators = {n_estimators}, max_depth = {max_
depth:2}, learning_rate = {learning_rate:.6f}, subsample = {subsample:.6f}, cols
ample_bylevel = {colsample_bylevel:.6f}, colsample_bytree = {colsample_bytree:.6
f}, Score = {score:.5f}")


# hyperparameters_list를 Pandas의 DataFrame으로 변환합니다.
hyperparameters_list = pd.DataFrame.from_dict(hyperparameters_list)

# 변환한 hyperparameters_list를 score가 낮은 순으로 정렬합니다.
# (RMSE는 score가 낮을 수록 더 정확도가 높다고 가정합니다)
hyperparameters_list = hyperparameters_list.sort_values(by="score")

# hyperparameters_list 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(hyperparameters_list.shape)

# hyperparameters_list의 상위 5개를 출력합니다.
hyperparameters_list.head()
```

```
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.88140
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.04028
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.74037
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.22020
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.99091
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.68523
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.60839
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.73196
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.66976
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.61069
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.75889
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.58929
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.86022
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.61284
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.53668
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.18265
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.58497
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.51850
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.53155
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.70665
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
```

```
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.51485
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.84865
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.64217
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.47024
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.24395
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.66669
n_estimators = 100, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.47178
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.53103
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.38397
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.35315
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.42191
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37249
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34473
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.52158
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.36707
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34185
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.51322
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.38583
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34562
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.41997
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
```

```
00000, Score = 0.36695
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34069
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.53935
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.37323
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34397
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.52874
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37509
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34661
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.43909
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37413
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34096
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.47844
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.41386
n_estimators = 100, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.35398
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.88250
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.76747
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.67506
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.82334
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.72101
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.65721
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.80464
```

```
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.70569
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.65600
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.87677
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.75870
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.67053
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.80198
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.71315
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.65303
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.80207
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.70080
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.64972
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.88526
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.75469
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.66419
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.81615
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.70964
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.64731
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.75960
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.69044
n_estimators = 100, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.64830
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
```

```
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93626
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92659
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.91260
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.93155
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.92023
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90985
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92866
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.91826
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90992
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93510
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92519
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.91216
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.92912
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.91897
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90897
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92834
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.91773
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90893
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93433
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92385
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
```

```
00000, Score = 3.91177
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.92982
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.91809
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90866
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92437
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.91639
n_estimators = 100, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90895
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27128
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27038
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26896
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27075
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26977
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26872
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27062
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26957
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26869
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27120
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27032
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26889
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27060
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26964
```

```
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26858
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27058
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26951
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26859
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27119
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27003
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26886
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27073
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26965
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26857
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27019
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26937
n_estimators = 100, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26854
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.88140
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.04028
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.74037
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.22020
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.99091
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.68523
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.60839
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
```

```
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.73196
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.66976
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.61069
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.75889
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.58929
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.86022
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.61284
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.53668
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.18265
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.58497
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.51850
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.53155
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.70665
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.51483
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.84865
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.64217
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.47010
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.24395
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.66669
n_estimators = 100, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.47178
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
```

```
00000, Score = 0.53103
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.38397
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.35315
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.42191
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37249
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34473
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.52158
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.36707
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34185
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.51322
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.38583
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34562
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.41997
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.36695
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34069
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.53935
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.37323
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34397
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.52874
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37509
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34661
```

```
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.43909
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37413
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34096
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.47844
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.41386
n_estimators = 100, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.35398
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.88250
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.76747
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.67506
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.82334
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.72101
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.65721
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.80464
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.70569
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.65600
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.87677
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.75870
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.67053
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.80198
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.71315
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
```

```
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.65303
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.80207
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.70080
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.64972
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.88526
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.75469
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 1.66419
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.81615
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 1.70964
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 1.64731
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.75960
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 1.69044
n_estimators = 100, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 1.64830
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93626
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92659
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.91260
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.93155
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.92023
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90985
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92866
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
```

```
00000, Score = 3.91826
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90992
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93510
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92519
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.91216
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.92912
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.91897
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90897
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92834
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.91773
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90893
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.93433
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.92385
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.91177
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.92982
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.91809
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.90866
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.92437
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.91639
n_estimators = 100, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.90895
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27128
```

```
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27038
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26896
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27075
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26977
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26872
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27062
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26957
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26869
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27120
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27032
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26889
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27060
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26964
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26858
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27058
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26951
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26859
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.27119
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.27003
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.26886
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
```

```
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.27073
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.26965
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.26857
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.27019
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.26937
n_estimators = 100, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.26854
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.88140
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.04028
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.74037
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.22020
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.99091
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.68523
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.60839
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.73196
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.66976
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.61069
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.75889
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.58929
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.86022
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.61284
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
```

```
000000, Score = 0.53668
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.18265
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.58497
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.51850
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.53155
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.70665
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.51483
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.84865
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.64217
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.47010
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.24395
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.66669
n_estimators = 100, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.47178
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.53103
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.38397
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.35315
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.42191
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37249
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34473
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.52158
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36707
```

```
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34185
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.51322
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.38583
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34562
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.41997
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.36695
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34069
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.53935
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.37323
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34397
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.52874
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37509
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34661
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.43909
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37413
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34096
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.47844
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.41386
n_estimators = 100, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35398
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.88250
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
```

```
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.76747
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67506
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.82334
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.72101
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65721
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.80464
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.70569
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.65600
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.87677
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.75870
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67053
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.80198
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.71315
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65303
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.80207
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.70080
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.64972
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.88526
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.75469
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.66419
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
```

```
400000, Score = 1.81615
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.70964
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.64731
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.75960
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.69044
n_estimators = 100, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.64830
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93626
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92659
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91260
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.93155
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.92023
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90985
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92866
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91826
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90992
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93510
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92519
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91216
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92912
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91897
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90897
```

```
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92834
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91773
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90893
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93433
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92385
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91177
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92982
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91809
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90866
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92437
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91639
n_estimators = 100, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90895
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.27128
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.27038
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.26896
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.27075
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.26977
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.26872
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.27062
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.26957
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
```

```
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.26869
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.27120
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.27032
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.26889
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.27060
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.26964
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.26858
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.27058
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.26951
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.26859
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.27119
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.27003
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.26886
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.27073
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.26965
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.26857
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.27019
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.26937
n_estimators = 100, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.26854
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.03727
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
```

```
00000, Score = 1.04561
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.74806
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.25190
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.99481
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.69283
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.61467
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.73941
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.67267
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.64444
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.76162
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.59338
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.86105
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.61517
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.53912
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.18252
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.58913
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.52103
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.53749
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.70665
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.51485
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.84864
```

```
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.64217
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.47024
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.24395
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.66669
n_estimators = 300, max_depth = 50, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.47178
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.39132
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37640
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.35418
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.38960
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37248
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34631
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.48697
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.36704
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34348
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.39122
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37948
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34587
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.39368
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.36662
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34112
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
```

```
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.50976
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.37297
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34438
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.38995
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37092
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34658
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.40778
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37358
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34093
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.46860
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.41328
n_estimators = 300, max_depth = 50, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.35397
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.87125
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.58177
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.44646
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.68789
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.49692
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42465
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66506
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48266
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
```

```
00000, Score = 0.42327
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.86018
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.57311
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.44183
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.68430
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.48656
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42404
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66445
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48279
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.42419
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.87578
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.56675
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.43985
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.69381
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.48766
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42366
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66380
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48582
n_estimators = 300, max_depth = 50, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.43369
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29585
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.26528
```

```
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22829
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27742
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24721
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.22036
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.27241
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24596
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.22021
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29594
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.26192
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22646
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27474
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24555
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.21784
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.27122
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24424
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.21876
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29446
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.25987
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22425
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27360
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
```

```
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24382
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.21686
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.26918
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24314
n_estimators = 300, max_depth = 50, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.21789
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19404
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19115
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.18708
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19218
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18929
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18621
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.19183
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18912
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18620
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19385
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19079
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.18681
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19202
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18902
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18584
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
```

```
00000, Score = 4.19170
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18895
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18587
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19389
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19032
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.18662
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19202
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18896
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18584
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.19149
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18882
n_estimators = 300, max_depth = 50, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18563
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 1.03727
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 1.04561
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.74806
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 1.25190
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.99481
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.69283
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.61467
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.73941
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.67267
```

```
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.64444
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.76162
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.59338
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.86105
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.61517
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.53912
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.18252
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.58913
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.52103
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.53749
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.70665
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.51483
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.84864
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.64217
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.47010
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 1.24395
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.66669
n_estimators = 300, max_depth = 75, learning_rate = 1.000000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.47178
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.39132
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37640
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
```

```
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.35418
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.38960
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.37248
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34631
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.48697
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.36704
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34343
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.39122
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37948
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34591
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.39368
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.36662
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34112
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.50976
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.37297
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.34438
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.38995
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.37092
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.34658
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.40778
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
```

```
00000, Score = 0.37358
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.34093
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.46860
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.41328
n_estimators = 300, max_depth = 75, learning_rate = 0.100000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.35397
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.87125
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.58177
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.44646
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.68789
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.49692
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42465
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66506
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48266
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.42327
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.86018
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.57311
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.44183
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.68430
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.48656
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42404
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66445
```

```
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48279
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.42419
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 0.87578
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 0.56675
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 0.43985
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 0.69381
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 0.48766
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 0.42366
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 0.66380
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 0.48582
n_estimators = 300, max_depth = 75, learning_rate = 0.010000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 0.43369
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29585
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.26528
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22829
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27742
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24721
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.22036
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.27241
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24596
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.22021
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
```

```
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29594
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.26192
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22646
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27474
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24555
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.21784
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.27122
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24424
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.21876
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 3.29446
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 3.25987
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 3.22425
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 3.27360
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 3.24382
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 3.21686
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 3.26918
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 3.24314
n_estimators = 300, max_depth = 75, learning_rate = 0.001000, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 3.21789
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19404
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19115
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
```

```
00000, Score = 4.18708
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19218
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18929
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18621
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.19183
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18912
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18620
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19385
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19079
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.18681
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19202
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18902
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18584
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.19170
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18895
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18587
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.4
00000, Score = 4.19389
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.7
00000, Score = 4.19032
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.0
00000, Score = 4.18662
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.4
00000, Score = 4.19202
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.7
00000, Score = 4.18896
```

```
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.0
00000, Score = 4.18584
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.4
00000, Score = 4.19149
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.7
00000, Score = 4.18882
n_estimators = 300, max_depth = 75, learning_rate = 0.000100, subsam
ple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.0
00000, Score = 4.18563
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.03727
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.04561
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.74806
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.25190
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.99481
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.69283
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.61467
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.73941
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.67267
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.64444
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.76162
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.59338
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.86105
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.61517
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.53912
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.18252
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
```

```
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.58913
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.52103
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.53749
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.70665
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.51483
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.84864
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.64217
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.47010
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.24395
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.66669
n_estimators = 300, max_depth = 100, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.47178
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.39132
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37640
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.35418
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.38960
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37248
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34631
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.48697
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36704
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34343
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
```

```
400000, Score = 0.39122
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37948
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34591
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39368
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.36662
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34112
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.50976
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.37297
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34438
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.38995
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37092
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34658
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40778
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37358
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34093
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.46860
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.41328
n_estimators = 300, max_depth = 100, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35397
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.87125
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.58177
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.44646
```

```
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.68789
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.49692
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.42465
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.66506
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.48266
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.42327
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.86018
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.57311
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.44183
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.68430
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.48656
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.42404
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.66445
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.48279
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.42419
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.87578
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.56675
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.43985
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.69381
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.48766
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
```

```
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.42366
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.66380
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.48582
n_estimators = 300, max_depth = 100, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.43369
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.29585
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.26528
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.22829
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.27742
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.24721
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.22036
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.27241
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.24596
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.22021
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.29594
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.26192
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.22646
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.27474
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.24555
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.21784
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.27122
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
```

```
700000, Score = 3.24424
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.21876
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.29446
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.25987
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.22425
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.27360
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.24382
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.21686
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.26918
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.24314
n_estimators = 300, max_depth = 100, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.21789
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.19404
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.19115
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.18708
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.19218
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.18929
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.18621
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.19183
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.18912
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.18620
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.19385
```

```
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.19079
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.18681
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.19202
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.18902
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.18584
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.19170
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.18895
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.18587
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 4.19389
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 4.19032
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 4.18662
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 4.19202
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 4.18896
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 4.18584
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 4.19149
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 4.18882
n_estimators = 300, max_depth = 100, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 4.18563
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.16183
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.05518
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.76923
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
```

```
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.25621
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.01160
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.71913
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.62710
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.76576
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.68825
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.65312
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.76759
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.60073
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.86349
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.62512
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.55728
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.18495
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.60435
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.52858
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.53854
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.70665
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.51485
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.84864
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.64217
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
```

```
000000, Score = 0.47024
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.24395
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.66669
n_estimators = 1000, max_depth = 50, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.47178
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36401
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37708
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.35434
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39023
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37288
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34657
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.48435
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36710
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34360
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36301
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37960
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34608
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39399
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.36677
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34133
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.50752
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.37281
```

```
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34425
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36316
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37092
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34658
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40725
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37358
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34093
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.46806
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.41328
n_estimators = 1000, max_depth = 50, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35397
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50462
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.35701
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33262
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39581
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.34670
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.32932
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.41676
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.34589
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.33061
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50284
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
```

```
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.36157
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33473
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40915
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.34892
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.33178
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.42889
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.35402
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.33556
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50934
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.36659
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33682
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40981
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.35600
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.33555
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.44791
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36241
n_estimators = 1000, max_depth = 50, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35090
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89961
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.77525
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.68269
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
```

```
400000, Score = 1.81981
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.72476
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.66408
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79626
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.71422
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.66129
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89732
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.76453
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67659
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.81464
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.72087
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65783
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79305
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.71003
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.65633
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89328
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.75946
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67169
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.81101
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.71611
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65437
```

```
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79683
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.70776
n_estimators = 1000, max_depth = 50, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.65542
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93577
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92584
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91312
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92957
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.92058
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.91013
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92743
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91882
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.91013
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93530
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92501
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91208
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92932
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91902
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90919
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92702
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91826
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
```

```
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90927
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93542
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92385
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91149
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92862
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91903
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90895
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92731
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91805
n_estimators = 1000, max_depth = 50, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90912
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.16183
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.05518
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.76923
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.25621
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.01160
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.71913
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.62710
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.76576
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.68825
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.65312
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
```

```
700000, Score = 0.76759
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.60073
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.86349
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.62512
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.55728
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.18495
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.60435
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.52858
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.53854
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.70665
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.51483
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.84864
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.64217
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.47010
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.24395
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.66669
n_estimators = 1000, max_depth = 75, learning_rate = 1.000000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.47178
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36401
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37708
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.35434
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39023
```

```
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37288
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34657
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.48435
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36710
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34356
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36301
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37960
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34623
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39399
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.36677
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34133
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.50752
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.37281
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.34424
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.36316
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.37092
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.34658
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40725
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.37358
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.34093
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
```

```
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.46806
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.41328
n_estimators = 1000, max_depth = 75, learning_rate = 0.100000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35397
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50462
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.35701
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33262
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.39581
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.34670
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.32934
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.41676
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.34589
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.33061
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50284
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.36157
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33473
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40915
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.34892
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.33178
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.42889
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.35402
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
```

```
000000, Score = 0.33556
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 0.50934
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 0.36659
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 0.33682
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 0.40981
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 0.35600
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 0.33555
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 0.44791
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 0.36241
n_estimators = 1000, max_depth = 75, learning_rate = 0.010000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 0.35090
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89961
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.77525
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.68269
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.81981
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.72476
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.66408
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79626
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.71422
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.66129
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89732
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.76453
```

```
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67659
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.81464
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.72087
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65783
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79305
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.71003
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.65633
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 1.89328
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 1.75946
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 1.67169
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 1.81101
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 1.71611
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 1.65437
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 1.79683
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 1.70776
n_estimators = 1000, max_depth = 75, learning_rate = 0.001000, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 1.65542
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93577
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92584
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91312
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92957
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
```

```
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.92058
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.91013
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92743
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91882
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.500000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.91013
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93530
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92501
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91208
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92932
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91902
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90919
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
400000, Score = 3.92702
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91826
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 0.750000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90927
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
400000, Score = 3.93542
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 0.
700000, Score = 3.92385
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.400000, colsample_bytree = 1.
000000, Score = 3.91149
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
400000, Score = 3.92862
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 0.
700000, Score = 3.91903
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 0.700000, colsample_bytree = 1.
000000, Score = 3.90895
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
```

```
400000, Score = 3.92731
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 0.
700000, Score = 3.91805
n_estimators = 1000, max_depth = 75, learning_rate = 0.000100, subsa
mple = 1.000000, colsample_bylevel = 1.000000, colsample_bytree = 1.
000000, Score = 3.90912
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 1.16183
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 1.05518
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.76923
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 1.25621
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 1.01160
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.71913
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.62710
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.76576
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.68825
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.65312
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.76759
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.60073
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.86349
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.62512
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.55728
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.18495
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.60435
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.52858
```

```
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.53854
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.70665
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.51483
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.84864
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.64217
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.47010
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.24395
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.66669
n_estimators = 1000, max_depth = 100, learning_rate = 1.000000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.47178
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.36401
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.37708
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.35434
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.39023
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.37288
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.34657
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.48435
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.36710
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.34356
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.36301
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.37960
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
```

```
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.34623
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.39399
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.36677
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.34133
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.50752
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.37281
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.34424
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.36316
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.37092
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.34658
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.40725
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.37358
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.34093
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.46806
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.41328
n_estimators = 1000, max_depth = 100, learning_rate = 0.100000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.35397
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.50462
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.35701
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.33262
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.39581
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
```

```
 0.700000, Score = 0.34670
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.32934
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.41676
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.34589
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.33061
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.50284
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.36157
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.33473
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.40915
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.34892
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.33178
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.42889
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.35402
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.33556
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 0.50934
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 0.36659
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 0.33682
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 0.40981
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 0.35600
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 0.33555
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 0.44791
```

```
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 0.36241
n_estimators = 1000, max_depth = 100, learning_rate = 0.010000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 0.35090
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 1.89961
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 1.77525
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 1.68269
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 1.81981
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 1.72476
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 1.66408
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.79626
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 1.71422
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 1.66129
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 1.89732
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 1.76453
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 1.67659
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 1.81464
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 1.72087
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 1.65783
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.79305
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 1.71003
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 1.65633
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
```

```
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 1.89328
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 1.75946
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 1.67169
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 1.81101
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 1.71611
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 1.65437
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 1.79683
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 1.70776
n_estimators = 1000, max_depth = 100, learning_rate = 0.001000, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 1.65542
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 3.93577
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 3.92584
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 3.91312
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 3.92957
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 3.92058
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 3.91013
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 3.92743
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 3.91882
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.500000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 3.91013
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 3.93530
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 3.92501
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.400000, colsample_bytree =
```

```
 1.000000, Score = 3.91208
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 3.92932
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 3.91902
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 3.90919
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 3.92702
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 3.91826
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 0.750000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 3.90927
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.400000, Score = 3.93542
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 0.700000, Score = 3.92385
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.400000, colsample_bytree =
 1.000000, Score = 3.91149
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.400000, Score = 3.92862
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 0.700000, Score = 3.91903
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 0.700000, colsample_bytree =
 1.000000, Score = 3.90895
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.400000, Score = 3.92731
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 0.700000, Score = 3.91805
n_estimators = 1000, max_depth = 100, learning_rate = 0.000100, subs
ample = 1.000000, colsample_bylevel = 1.000000, colsample_bytree =
 1.000000, Score = 3.90912
(1215, 7)
```

Out[25]:

	colsample_bylevel	colsample_bytree	learning_rate	max_depth	n_estimators	
869	0.7	1.0	0.01	50	1000	(
1139	0.7	1.0	0.01	100	1000	(
1004	0.7	1.0	0.01	75	1000	(
1007	1.0	1.0	0.01	75	1000	(
1142	1.0	1.0	0.01	100	1000	(

Case 2 - Coarse & Finer Search

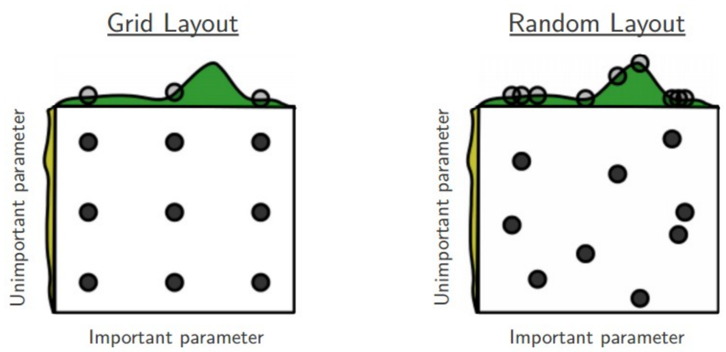
Grid Search는 굉장히 좋은 방식이지만 한 가지 큰 단점을 갖고 있습니다. 바로 **Grid Search로는 거의 대부분의 경우 가장 좋은 하이퍼패러미터를 찾을 수 없다는 사실**입니다.

가령 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 문제에서 가장 적합한 max_depth가 83이라고 한다면, 위 Grid Search에서는 찾을 수 없습니다. 왜냐하면 위 코드에서 max_depth의 후보군을 다음과 같이 지정해놓았기 때문입니다.

```
# 이 후보군에는 max_depth가 83인 경우가 없습니다.
max_depth_list = [50, 75, 100]
```

그러므로 Grid Search로는 가장 좋은 하이퍼패러미터에 근접한 다른 하이퍼패러미터는 찾을 수 있지만, 가장 좋은 하이퍼패러미터를 찾는 것은 어렵습니다.

그렇다면 어떻게 하면 가장 좋은 하이퍼패러미터를 찾을 수 있을까요? 답은 간단합니다. 이론상으로 존재 가능한 모든 하이퍼패러미터 범위에서 랜덤하게 찾아서 Cross Validation을 해보면 됩니다. 이 방식을 랜덤 서치(Random Search)라고 합니다.



위 그림과 같이, Grid Search를 활용하면 가장 좋은 성능을 내는 하이퍼패러미터를 찾기 어렵습니다. 이런 경우는 Random Search를 사용합니다.

(see Random Search for Hyper-Parameter Optimization (http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf))

하지만 랜덤 서치(Random Search)는 현실적으로 시간이 오래 걸리기 때문에, 랜덤 서치(Random Search)를 응용한 다른 하이퍼패러미터 튜닝 방식을 사용하겠습니다. 바로 **Coarse & Finer Search** 입니다.

Coarse & Finer Search는 크게 1) Coarse Search와 2) Finer Search로 동작합니다

먼저 **Coarse Search**에서는 Random Search를 하되, 이론상으로 존재 가능한 모든 하이퍼패러미터 범위를 집어넣습니다. 이렇게 Random Search를 하면 가장 좋은 하이퍼패러미터를 찾는 것은 어렵지만, **좋지 않은 하이퍼패러미터를 정렬해서 후순위로 놓을 수 있습니다.**

이를 통해 좋지 않은 하이퍼패러미터를 버린 뒤 다시 한 번 Random Search를 하는 것을 **Finer Search**라고 합니다.

Coarse Search

In [26]:

```
# Gradient Boosting Machine 패키지인 XGBoost를 가져옵니다.
# 이를 xgb라는 축약어로 사용합니다.
import xgboost as xgb

# scikit-learn 패키지의 model_selection 모듈에 있는 cross_val_score 함수를 가지고 옵니다.
from sklearn.model_selection import cross_val_score

# 랜덤 서치를 반복할 횟수입니다.
# 보통 100번을 반복합니다.
num_epoch = 100

# hyperparameter 탐색 결과를 리스트로 저장합니다.
coarse_hyperparameters_list = []

# num_epoch 횟수만큼 랜덤 서치를 반복합니다.
for epoch in range(num_epoch):
    # 10에서 100 사이의 정수형(int) 값을 랜덤하게 생성하여 n_estimators 변수에 할당합니다.
    n_estimators = np.random.randint(low=100, high=1000)

    # 2에서 100 사이의 정수형(int) 값을 랜덤하게 생성하여 max_depth 변수에 할당합니다.
    max_depth = np.random.randint(low=2, high=100)

    # 1.0에서 1-e10(10의 -10승)사이의 실수형(float) 값을 랜덤하게 생성하여  learning_rate 변
수에 할당합니다.
    learning_rate = 10 ** -np.random.uniform(low=0, high=10)

    # 0.1에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 subsample 변수에 할당합니다.
    subsample = np.random.uniform(low=0.1, high=1.0)

    # 0.4에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bytree 변수에 할당합니
다.
    colsample_bytree = np.random.uniform(low=0.4, high=1.0)

    # 0.4에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bylevel 변수에 할당합니
다.
    colsample_bylevel = np.random.uniform(low=0.4, high=1.0)

    # XGBRegressor를 생성합니다. 실행할때는 다음의 옵션이 들어갑니다.
    # 1) n_estimators. 트리의 갯수입니다. 지정한 갯수만큼 트리를 생성합니다.
    # 2) max_depth. 트리의 깊이입니다. 지정한 숫자만큼 트리가 깊게 가지를 뻗습니다.
    # 3) learning_rate. 각 트리마다의 비중을 나타냅니다. 너무 작으면 과적합(overfitting)될 가
능성이 있고, 너무 높으면 부적합(underfitting)될 가능성이 있습니다.
    # 4) subsample. 하나의 트리를 만들 때 사용할 데이터의 비율을 나타냅니다. 0.0 ~ 1.0 사이의 값
을 넣으면 지정한 비율만큼만 랜덤하게 데이터를 사용합니다.
    # 5) colsample_bytree. 하나의 트리를 만들 때 사용할 feature의 비율을 나타냅니다. 0.0 ~
 1.0 사이의 값을 넣으면 트리를 만들 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
    # 6) colsample_bylevel. 트리가 한 번 가지를 칠 때 사용할 feature의 비율을 나타냅니다. 0.0
 ~ 1.0 사이의 값을 넣으면 트리가 가지를 칠 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
    # 생성한 XGBRegressor를 model이라는 이름의 변수에 대입합니다.
    model = xgb.XGBRegressor(n_estimators=n_estimators,
                             max_depth=max_depth,
                             learning_rate=learning_rate,
                             subsample=subsample,
                             colsample_bylevel=colsample_bylevel,
                             colsample_bytree=colsample_bytree,
                             seed=37)

    # cross_val_score를 실행합니다. 실행할 때는 다음의 옵션이 들어갑니다.
    # 1) model. 점수를 측정할 머신러닝 모델이 들어갑니다.
```

```
    # 2) X_train. train 데이터의 feature 입니다.
    # 3) y_train. train 데이터의 label 입니다.
    # 4) cv. Cross Validation에서 데이터를 조각낼(split) 갯수입니다. 총 20조각을 내야하기 때문
에 20을 대입합니다.
    # 5) scoring. 점수를 측정할 공식입니다. 앞서 구현한 RMSE를 적용합니다.
    # 마지막으로, 이 함수의 실행 결과의 평균(mean)을 구한 뒤 score라는 이름의 새로운 변수에 할당합니
다.
    score = cross_val_score(model, X_train, y_train, cv=20, scoring=rmse_score).
mean()

    # hyperparameter 탐색 결과를 딕셔너리화 합니다.
    hyperparameters = {
        'epoch': epoch,
        'score': score,
        'n_estimators': n_estimators,
        'max_depth': max_depth,
        'learning_rate': learning_rate,
        'subsample': subsample,
        'colsample_bylevel': colsample_bylevel,
        'colsample_bytree': colsample_bytree,
    }

    # hyperparameter 탐색 결과를 리스트에 저장합니다.
    coarse_hyperparameters_list.append(hyperparameters)

    # hyperparameter 탐색 결과를 출력합니다.
    print(f"{epoch:2} n_estimators = {n_estimators}, max_depth = {max_depth:2},
 learning_rate = {learning_rate:.10f}, subsample = {subsample:.6f}, colsample_by
level = {colsample_bylevel:.6f}, colsample_bytree = {colsample_bytree:.6f}, Scor
e = {score:.5f}")


# coarse_hyperparameters_list를 Pandas의 DataFrame으로 변환합니다.
coarse_hyperparameters_list = pd.DataFrame.from_dict(coarse_hyperparameters_list
)

# 변환한 coarse_hyperparameters_list를 score가 낮은 순으로 정렬합니다.
# (RMSE는 score가 낮을 수록 더 정확도가 높다고 가정합니다)
coarse_hyperparameters_list = coarse_hyperparameters_list.sort_values(by="score"
)

# coarse_hyperparameters_list 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(coarse_hyperparameters_list.shape)

# coarse_hyperparameters_list의 상위 10개를 출력합니다.
coarse_hyperparameters_list.head(10)
```

```
 0 n_estimators = 699, max_depth = 24, learning_rate = 0.2322858010,
subsample = 0.345621, colsample_bylevel = 0.768295, colsample_bytree
= 0.477353, Score = 0.45069
 1 n_estimators = 932, max_depth = 27, learning_rate = 0.0209459632,
subsample = 0.289088, colsample_bylevel = 0.863977, colsample_bytree
= 0.890338, Score = 0.32996
 2 n_estimators = 138, max_depth = 68, learning_rate = 0.0000000047,
subsample = 0.733244, colsample_bylevel = 0.943907, colsample_bytree
= 0.656607, Score = 4.31058
 3 n_estimators = 747, max_depth = 18, learning_rate = 0.0049878620,
subsample = 0.715803, colsample_bylevel = 0.612042, colsample_bytree
= 0.848196, Score = 0.37242
 4 n_estimators = 149, max_depth = 49, learning_rate = 0.2041003999,
subsample = 0.124723, colsample_bylevel = 0.886809, colsample_bytree
= 0.988393, Score = 0.44243
 5 n_estimators = 636, max_depth = 16, learning_rate = 0.0000000030,
subsample = 0.504256, colsample_bylevel = 0.547654, colsample_bytree
= 0.631089, Score = 4.31057
 6 n_estimators = 181, max_depth = 13, learning_rate = 0.0791185412,
subsample = 0.778489, colsample_bylevel = 0.897664, colsample_bytree
= 0.930401, Score = 0.34003
 7 n_estimators = 418, max_depth = 17, learning_rate = 0.1830706502,
subsample = 0.310375, colsample_bylevel = 0.500772, colsample_bytree
= 0.672945, Score = 0.39840
 8 n_estimators = 503, max_depth = 21, learning_rate = 0.0000722441,
subsample = 0.378579, colsample_bylevel = 0.739344, colsample_bytree
= 0.938420, Score = 4.16130
 9 n_estimators = 634, max_depth = 54, learning_rate = 0.0000013401,
subsample = 0.403305, colsample_bylevel = 0.673181, colsample_bytree
= 0.993229, Score = 4.30703
10 n_estimators = 716, max_depth = 33, learning_rate = 0.0001939084,
subsample = 0.781217, colsample_bylevel = 0.695672, colsample_bytree
= 0.775315, Score = 3.77404
11 n_estimators = 213, max_depth = 45, learning_rate = 0.0000000108,
subsample = 0.572285, colsample_bylevel = 0.443255, colsample_bytree
= 0.795412, Score = 4.31057
12 n_estimators = 446, max_depth = 83, learning_rate = 0.0640465370,
subsample = 0.150685, colsample_bylevel = 0.937346, colsample_bytree
= 0.814871, Score = 0.35610
13 n_estimators = 747, max_depth = 88, learning_rate = 0.0002217402,
subsample = 0.726182, colsample_bylevel = 0.946189, colsample_bytree
= 0.903247, Score = 3.67416
14 n_estimators = 489, max_depth = 57, learning_rate = 0.0201989867,
subsample = 0.475406, colsample_bylevel = 0.774680, colsample_bytree
= 0.580041, Score = 0.36350
15 n_estimators = 839, max_depth = 45, learning_rate = 0.5442394865,
subsample = 0.995695, colsample_bylevel = 0.478000, colsample_bytree
= 0.753065, Score = 0.42932
16 n_estimators = 356, max_depth = 85, learning_rate = 0.0000000038,
subsample = 0.373146, colsample_bylevel = 0.897079, colsample_bytree
= 0.583530, Score = 4.31057
17 n_estimators = 935, max_depth = 75, learning_rate = 0.9807720629,
subsample = 0.893219, colsample_bylevel = 0.637073, colsample_bytree
= 0.439554, Score = 0.74638
18 n_estimators = 696, max_depth = 43, learning_rate = 0.0000000009,
subsample = 0.653544, colsample_bylevel = 0.522975, colsample_bytree
= 0.716282, Score = 4.31058
19 n_estimators = 495, max_depth = 80, learning_rate = 0.0000030272,
subsample = 0.973307, colsample_bylevel = 0.785242, colsample_bytree
= 0.815694, Score = 4.30437
20 n_estimators = 884, max_depth = 61, learning_rate = 0.0000023313,
```

```
subsample = 0.399694, colsample_bylevel = 0.523210, colsample_bytree
= 0.775448, Score = 4.30210
21 n_estimators = 687, max_depth = 73, learning_rate = 0.0006810143,
subsample = 0.829407, colsample_bylevel = 0.934895, colsample_bytree
= 0.439556, Score = 2.81349
22 n_estimators = 364, max_depth = 34, learning_rate = 0.0000103534,
subsample = 0.172567, colsample_bylevel = 0.859149, colsample_bytree
= 0.628469, Score = 4.29529
23 n_estimators = 912, max_depth = 12, learning_rate = 0.0000019339,
subsample = 0.621734, colsample_bylevel = 0.406301, colsample_bytree
= 0.690087, Score = 4.30345
24 n_estimators = 362, max_depth = 79, learning_rate = 0.0000005338,
subsample = 0.354466, colsample_bylevel = 0.789681, colsample_bytree
= 0.670166, Score = 4.30979
25 n_estimators = 868, max_depth = 59, learning_rate = 0.0229278034,
subsample = 0.404547, colsample_bylevel = 0.468473, colsample_bytree
= 0.869623, Score = 0.33691
26 n_estimators = 965, max_depth = 15, learning_rate = 0.0000004128,
subsample = 0.939578, colsample_bylevel = 0.725404, colsample_bytree
= 0.909890, Score = 4.30891
27 n_estimators = 836, max_depth = 89, learning_rate = 0.0000037724,
subsample = 0.726318, colsample_bylevel = 0.683683, colsample_bytree
= 0.500918, Score = 4.29781
28 n_estimators = 323, max_depth = 26, learning_rate = 0.0000000392,
subsample = 0.978342, colsample_bylevel = 0.940089, colsample_bytree
= 0.642947, Score = 4.31053
29 n_estimators = 787, max_depth =  7, learning_rate = 0.0000995481,
subsample = 0.381461, colsample_bylevel = 0.785578, colsample_bytree
= 0.470377, Score = 4.00591
30 n_estimators = 189, max_depth = 68, learning_rate = 0.0000167665,
subsample = 0.877068, colsample_bylevel = 0.672164, colsample_bytree
= 0.827566, Score = 4.29737
31 n_estimators = 201, max_depth = 50, learning_rate = 0.0000001114,
subsample = 0.433854, colsample_bylevel = 0.569141, colsample_bytree
= 0.681251, Score = 4.31049
32 n_estimators = 959, max_depth = 71, learning_rate = 0.0056328430,
subsample = 0.714654, colsample_bylevel = 0.987466, colsample_bytree
= 0.682794, Score = 0.36501
33 n_estimators = 889, max_depth = 90, learning_rate = 0.0004388320,
subsample = 0.382027, colsample_bylevel = 0.661549, colsample_bytree
= 0.797469, Score = 2.97803
34 n_estimators = 284, max_depth = 11, learning_rate = 0.0000000002,
subsample = 0.602027, colsample_bylevel = 0.456392, colsample_bytree
= 0.542921, Score = 4.31058
35 n_estimators = 173, max_depth = 95, learning_rate = 0.0003143775,
subsample = 0.368995, colsample_bylevel = 0.516508, colsample_bytree
= 0.960402, Score = 4.09000
36 n_estimators = 647, max_depth = 55, learning_rate = 0.0000000361,
subsample = 0.828360, colsample_bylevel = 0.513000, colsample_bytree
= 0.449799, Score = 4.31049
37 n_estimators = 232, max_depth = 81, learning_rate = 0.0000003053,
subsample = 0.595217, colsample_bylevel = 0.729356, colsample_bytree
= 0.902786, Score = 4.31028
38 n_estimators = 365, max_depth = 93, learning_rate = 0.0358787616,
subsample = 0.898861, colsample_bylevel = 0.908478, colsample_bytree
= 0.566676, Score = 0.38425
39 n_estimators = 779, max_depth = 84, learning_rate = 0.0147565408,
subsample = 0.306529, colsample_bylevel = 0.403381, colsample_bytree
= 0.623426, Score = 0.35509
40 n_estimators = 318, max_depth = 90, learning_rate = 0.0000000408,
subsample = 0.241957, colsample_bylevel = 0.550880, colsample_bytree
```

```
= 0.645874, Score = 4.31053
41 n_estimators = 885, max_depth = 87, learning_rate = 0.0000048304,
subsample = 0.150566, colsample_bylevel = 0.530487, colsample_bytree
= 0.646527, Score = 4.29327
42 n_estimators = 774, max_depth =  4, learning_rate = 0.3656108884,
subsample = 0.999501, colsample_bylevel = 0.943843, colsample_bytree
= 0.491711, Score = 0.34537
43 n_estimators = 682, max_depth = 89, learning_rate = 0.0000037424,
subsample = 0.590769, colsample_bylevel = 0.545058, colsample_bytree
= 0.546107, Score = 4.30020
44 n_estimators = 629, max_depth = 29, learning_rate = 0.0000204281,
subsample = 0.930338, colsample_bylevel = 0.578545, colsample_bytree
= 0.630497, Score = 4.25849
45 n_estimators = 829, max_depth = 60, learning_rate = 0.0009029055,
subsample = 0.608412, colsample_bylevel = 0.524489, colsample_bytree
= 0.696356, Score = 2.15942
46 n_estimators = 810, max_depth = 89, learning_rate = 0.0000000492,
subsample = 0.272125, colsample_bylevel = 0.980866, colsample_bytree
= 0.917275, Score = 4.31041
47 n_estimators = 403, max_depth =  5, learning_rate = 0.0399934453,
subsample = 0.651052, colsample_bylevel = 0.467062, colsample_bytree
= 0.835914, Score = 0.32159
48 n_estimators = 857, max_depth = 10, learning_rate = 0.0000002220,
subsample = 0.868509, colsample_bylevel = 0.453792, colsample_bytree
= 0.434522, Score = 4.30983
49 n_estimators = 703, max_depth = 45, learning_rate = 0.0044924140,
subsample = 0.721587, colsample_bylevel = 0.635406, colsample_bytree
= 0.867280, Score = 0.42002
50 n_estimators = 426, max_depth = 27, learning_rate = 0.1464556134,
subsample = 0.543915, colsample_bylevel = 0.473488, colsample_bytree
= 0.733188, Score = 0.37602
51 n_estimators = 699, max_depth =  7, learning_rate = 0.0262003173,
subsample = 0.513865, colsample_bylevel = 0.631298, colsample_bytree
= 0.971718, Score = 0.32117
52 n_estimators = 400, max_depth = 76, learning_rate = 0.0000000044,
subsample = 0.244632, colsample_bylevel = 0.507006, colsample_bytree
= 0.634038, Score = 4.31057
53 n_estimators = 192, max_depth = 89, learning_rate = 0.0000179771,
subsample = 0.916516, colsample_bylevel = 0.892634, colsample_bytree
= 0.900298, Score = 4.29620
54 n_estimators = 814, max_depth =  3, learning_rate = 0.1425177568,
subsample = 0.250722, colsample_bylevel = 0.681453, colsample_bytree
= 0.840537, Score = 0.33576
55 n_estimators = 501, max_depth = 21, learning_rate = 0.0000000450,
subsample = 0.145558, colsample_bylevel = 0.755875, colsample_bytree
= 0.601645, Score = 4.31049
56 n_estimators = 331, max_depth = 93, learning_rate = 0.1181079429,
subsample = 0.334909, colsample_bylevel = 0.854929, colsample_bytree
= 0.558321, Score = 0.40052
57 n_estimators = 123, max_depth = 29, learning_rate = 0.0002807910,
subsample = 0.561204, colsample_bylevel = 0.599664, colsample_bytree
= 0.627916, Score = 4.17255
58 n_estimators = 582, max_depth = 55, learning_rate = 0.0000000001,
subsample = 0.992588, colsample_bylevel = 0.960817, colsample_bytree
= 0.538212, Score = 4.31058
59 n_estimators = 377, max_depth = 55, learning_rate = 0.7438589701,
subsample = 0.356307, colsample_bylevel = 0.685136, colsample_bytree
= 0.598843, Score = 1.10299
60 n_estimators = 918, max_depth =  9, learning_rate = 0.0006800793,
subsample = 0.378812, colsample_bylevel = 0.869626, colsample_bytree
= 0.854009, Score = 2.37986
```

```
61 n_estimators = 490, max_depth = 29, learning_rate = 0.0000047022,
subsample = 0.810455, colsample_bylevel = 0.510392, colsample_bytree
= 0.761150, Score = 4.30107
62 n_estimators = 710, max_depth = 19, learning_rate = 0.0027408782,
subsample = 0.991736, colsample_bylevel = 0.564205, colsample_bytree
= 0.718276, Score = 0.83501
63 n_estimators = 761, max_depth = 85, learning_rate = 0.0192320397,
subsample = 0.748591, colsample_bylevel = 0.691748, colsample_bytree
= 0.589797, Score = 0.36735
64 n_estimators = 728, max_depth = 60, learning_rate = 0.1412635723,
subsample = 0.695288, colsample_bylevel = 0.710799, colsample_bytree
= 0.593679, Score = 0.42517
65 n_estimators = 316, max_depth = 27, learning_rate = 0.1471100928,
subsample = 0.145800, colsample_bylevel = 0.648763, colsample_bytree
= 0.605091, Score = 0.42332
66 n_estimators = 587, max_depth = 67, learning_rate = 0.0000000074,
subsample = 0.420016, colsample_bylevel = 0.650798, colsample_bytree
= 0.498415, Score = 4.31056
67 n_estimators = 945, max_depth = 19, learning_rate = 0.1531371146,
subsample = 0.548602, colsample_bylevel = 0.731302, colsample_bytree
= 0.496670, Score = 0.43069
68 n_estimators = 461, max_depth = 58, learning_rate = 0.0000000199,
subsample = 0.857297, colsample_bylevel = 0.849168, colsample_bytree
= 0.674343, Score = 4.31054
69 n_estimators = 506, max_depth = 57, learning_rate = 0.0000000130,
subsample = 0.330269, colsample_bylevel = 0.442864, colsample_bytree
= 0.888845, Score = 4.31055
70 n_estimators = 519, max_depth = 15, learning_rate = 0.0000000345,
subsample = 0.984410, colsample_bylevel = 0.412342, colsample_bytree
= 0.678033, Score = 4.31051
71 n_estimators = 562, max_depth = 88, learning_rate = 0.0002139609,
subsample = 0.208272, colsample_bylevel = 0.731471, colsample_bytree
= 0.878480, Score = 3.84127
72 n_estimators = 134, max_depth = 30, learning_rate = 0.0003101267,
subsample = 0.552828, colsample_bylevel = 0.732190, colsample_bytree
= 0.838838, Score = 4.14081
73 n_estimators = 633, max_depth = 58, learning_rate = 0.0000264021,
subsample = 0.443676, colsample_bylevel = 0.990289, colsample_bytree
= 0.972268, Score = 4.24121
74 n_estimators = 473, max_depth = 63, learning_rate = 0.0000002064,
subsample = 0.448687, colsample_bylevel = 0.608312, colsample_bytree
= 0.462555, Score = 4.31018
75 n_estimators = 380, max_depth = 84, learning_rate = 0.0000199118,
subsample = 0.454644, colsample_bylevel = 0.499235, colsample_bytree
= 0.507499, Score = 4.28029
76 n_estimators = 132, max_depth = 75, learning_rate = 0.0043060072,
subsample = 0.208642, colsample_bylevel = 0.651713, colsample_bytree
= 0.876266, Score = 2.51666
77 n_estimators = 429, max_depth =  6, learning_rate = 0.0069815001,
subsample = 0.964905, colsample_bylevel = 0.952162, colsample_bytree
= 0.947500, Score = 0.44311
78 n_estimators = 347, max_depth = 47, learning_rate = 0.0093663548,
subsample = 0.139800, colsample_bylevel = 0.932781, colsample_bytree
= 0.838805, Score = 0.41824
79 n_estimators = 766, max_depth = 93, learning_rate = 0.0000000118,
subsample = 0.903499, colsample_bylevel = 0.840421, colsample_bytree
= 0.792772, Score = 4.31054
80 n_estimators = 742, max_depth = 71, learning_rate = 0.0888087065,
subsample = 0.884192, colsample_bylevel = 0.527225, colsample_bytree
= 0.982032, Score = 0.34298
81 n_estimators = 338, max_depth = 82, learning_rate = 0.0000000320,
```

```
subsample = 0.282353, colsample_bylevel = 0.992051, colsample_bytree
= 0.792277, Score = 4.31053
82 n_estimators = 433, max_depth = 73, learning_rate = 0.0000001165,
subsample = 0.248062, colsample_bylevel = 0.754374, colsample_bytree
= 0.428163, Score = 4.31038
83 n_estimators = 743, max_depth = 50, learning_rate = 0.0302944539,
subsample = 0.321904, colsample_bylevel = 0.551960, colsample_bytree
= 0.774964, Score = 0.33755
84 n_estimators = 976, max_depth = 74, learning_rate = 0.0000000002,
subsample = 0.441274, colsample_bylevel = 0.543013, colsample_bytree
= 0.557282, Score = 4.31058
85 n_estimators = 816, max_depth = 24, learning_rate = 0.1196689432,
subsample = 0.576076, colsample_bylevel = 0.960542, colsample_bytree
= 0.880688, Score = 0.35638
86 n_estimators = 358, max_depth = 29, learning_rate = 0.0000000147,
subsample = 0.773707, colsample_bylevel = 0.554307, colsample_bytree
= 0.758348, Score = 4.31056
87 n_estimators = 139, max_depth = 33, learning_rate = 0.0000108395,
subsample = 0.261408, colsample_bylevel = 0.865924, colsample_bytree
= 0.579585, Score = 4.30444
88 n_estimators = 813, max_depth = 19, learning_rate = 0.0000137022,
subsample = 0.936285, colsample_bylevel = 0.743037, colsample_bytree
= 0.481291, Score = 4.26563
89 n_estimators = 359, max_depth = 34, learning_rate = 0.5547208781,
subsample = 0.880080, colsample_bylevel = 0.827680, colsample_bytree
= 0.937673, Score = 0.45891
90 n_estimators = 479, max_depth = 99, learning_rate = 0.0000000174,
subsample = 0.516692, colsample_bylevel = 0.729623, colsample_bytree
= 0.909218, Score = 4.31054
91 n_estimators = 548, max_depth =  9, learning_rate = 0.0000000037,
subsample = 0.658963, colsample_bylevel = 0.899043, colsample_bytree
= 0.989653, Score = 4.31057
92 n_estimators = 100, max_depth = 81, learning_rate = 0.0038433268,
subsample = 0.173474, colsample_bylevel = 0.762391, colsample_bytree
= 0.952989, Score = 2.98189
93 n_estimators = 870, max_depth = 20, learning_rate = 0.0000178511,
subsample = 0.589696, colsample_bylevel = 0.615853, colsample_bytree
= 0.515258, Score = 4.24814
94 n_estimators = 319, max_depth =  6, learning_rate = 0.0000000352,
subsample = 0.931856, colsample_bylevel = 0.640521, colsample_bytree
= 0.729642, Score = 4.31053
95 n_estimators = 371, max_depth = 65, learning_rate = 0.3309241398,
subsample = 0.259872, colsample_bylevel = 0.928881, colsample_bytree
= 0.499068, Score = 0.60782
96 n_estimators = 189, max_depth = 15, learning_rate = 0.1458215699,
subsample = 0.560602, colsample_bylevel = 0.998627, colsample_bytree
= 0.884252, Score = 0.36210
97 n_estimators = 236, max_depth = 84, learning_rate = 0.0000000095,
subsample = 0.516492, colsample_bylevel = 0.429003, colsample_bytree
= 0.505152, Score = 4.31057
98 n_estimators = 965, max_depth = 21, learning_rate = 0.0000005806,
subsample = 0.675574, colsample_bylevel = 0.414357, colsample_bytree
= 0.790556, Score = 4.30827
99 n_estimators = 810, max_depth = 84, learning_rate = 0.0000032186,
subsample = 0.138561, colsample_bylevel = 0.617605, colsample_bytree
= 0.724864, Score = 4.29998
(100, 8)
```

Out[26]:

	colsample_bylevel	colsample_bytree	epoch	learning_rate	max_depth	n_estimat
51	0.631298	0.971718	51	0.026200	7	699
47	0.467062	0.835914	47	0.039993	5	403
1	0.863977	0.890338	1	0.020946	27	932
54	0.681453	0.840537	54	0.142518	3	814
25	0.468473	0.869623	25	0.022928	59	868
83	0.551960	0.774964	83	0.030294	50	743
6	0.897664	0.930401	6	0.079119	13	181
80	0.527225	0.982032	80	0.088809	71	742
42	0.943843	0.491711	42	0.365611	4	774
39	0.403381	0.623426	39	0.014757	84	779

Coarse Search가 끝났으면, 상위 5 ~ 10개의 결과만 출력한 뒤 이 결과를 낸 하이퍼패러미터 범위만 남겨놓고 다시 한 번 Random Search를 합니다. 이를 Finer Search라고 합니다.

가령 위 Coarse Search를 통해, 다음의 하이퍼패러미터가 상위 5 ~ 10개 안에 들었다고 가정하겠습니다.

- n_estimators = 300 ~ 1,000
- max_depth = 2 ~ 60
- learning_rate = 1.0 ~ 0.01
- subsample = 0.2 ~ 0.7
- colsample_bytree = 0.7 ~ 1.0
- colsample_bylevel = 0.4 ~ 1.0

이제 위 코드를 그대로 사용하되, 다음의 부분만 수정한 뒤 다시 한 번 Random Search를 하겠습니다.

```
# 300에서 1000 사이의 정수형(int) 값을 랜덤하게 생성하여 n_estimators 변수에 할당합니다.
n_estimators = np.random.randint(low=300, high=1000)

# 2에서 60 사이의 정수형(int) 값을 랜덤하게 생성하여 max_depth 변수에 할당합니다.
max_depth = np.random.randint(low=2, high=60)

# 1.0에서 1-e2(10의 -2승)사이의 실수형(float) 값을 랜덤하게 생성하여  learning_rate
 변수에 할당합니다.
learning_rate = 10 ** -np.random.uniform(low=0, high=2)

# 0.2에서 0.7사이의 실수형(float) 값을 랜덤하게 생성하여 subsample 변수에 할당합니다.
subsample = np.random.uniform(low=0.2, high=0.7)

# 0.7에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bytree 변수에 할당합
니다.
colsample_bytree = np.random.uniform(low=0.7, high=1.0)

# 0.4에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bylevel 변수에 할당
합니다.
colsample_bylevel = np.random.uniform(low=0.4, high=1.0)
```

Finer Search

In [38]:

```
# Gradient Boosting Machine 패키지인 XGBoost를 가져옵니다.
# 이를 xgb라는 축약어로 사용합니다.
import xgboost as xgb

# scikit-learn 패키지의 model_selection 모듈에 있는 cross_val_score 함수를 가지고 옵니다.
from sklearn.model_selection import cross_val_score

# 랜덤 서치를 반복할 횟수입니다.
# 보통 100번을 반복합니다.
num_epoch = 100

# hyperparameter 탐색 결과를 리스트로 저장합니다.
finer_hyperparameters_list = []

# num_epoch 횟수만큼 랜덤 서치를 반복합니다.
for epoch in range(num_epoch):
    # 300에서 1000 사이의 정수형(int) 값을 랜덤하게 생성하여 n_estimators 변수에 할당합니다.
    n_estimators = np.random.randint(low=300, high=1000)

    # 2에서 60 사이의 정수형(int) 값을 랜덤하게 생성하여 max_depth 변수에 할당합니다.
    max_depth = np.random.randint(low=2, high=60)

    # 1.0에서 1-e2(10의 -2승)사이의 실수형(float) 값을 랜덤하게 생성하여  learning_rate 변수
에 할당합니다.
    learning_rate = 10 ** -np.random.uniform(low=0, high=2)

    # 0.2에서 0.7사이의 실수형(float) 값을 랜덤하게 생성하여 subsample 변수에 할당합니다.
    subsample = np.random.uniform(low=0.2, high=0.7)

    # 0.7에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bytree 변수에 할당합니
다.
    colsample_bytree = np.random.uniform(low=0.7, high=1.0)

    # 0.4에서 1.0사이의 실수형(float) 값을 랜덤하게 생성하여 colsample_bylevel 변수에 할당합니
다.
    colsample_bylevel = np.random.uniform(low=0.4, high=1.0)

    # XGBRegressor를 생성합니다. 실행할때는 다음의 옵션이 들어갑니다.
    # 1) n_estimators. 트리의 갯수입니다. 지정한 갯수만큼 트리를 생성합니다.
    # 2) max_depth. 트리의 깊이입니다. 지정한 숫자만큼 트리가 깊게 가지를 뻗습니다.
    # 3) learning_rate. 각 트리마다의 비중을 나타냅니다. 너무 작으면 과적합(overfitting)될 가
능성이 있고, 너무 높으면 부적합(underfitting)될 가능성이 있습니다.
    # 4) subsample. 하나의 트리를 만들 때 사용할 데이터의 비율을 나타냅니다. 0.0 ~ 1.0 사이의 값
을 넣으면 지정한 비율만큼만 랜덤하게 데이터를 사용합니다.
    # 5) colsample_bytree. 하나의 트리를 만들 때 사용할 feature의 비율을 나타냅니다. 0.0 ~
 1.0 사이의 값을 넣으면 트리를 만들 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
    # 6) colsample_bylevel. 트리가 한 번 가지를 칠 때 사용할 feature의 비율을 나타냅니다. 0.0
 ~ 1.0 사이의 값을 넣으면 트리가 가지를 칠 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
    # 생성한 XGBRegressor를 model이라는 이름의 변수에 대입합니다.
    model = xgb.XGBRegressor(n_estimators=n_estimators,
                             max_depth=max_depth,
                             learning_rate=learning_rate,
                             subsample=subsample,
                             colsample_bylevel=colsample_bylevel,
                             colsample_bytree=colsample_bytree,
                             seed=37)

    # cross_val_score를 실행합니다. 실행할 때는 다음의 옵션이 들어갑니다.
    # 1) model. 점수를 측정할 머신러닝 모델이 들어갑니다.
```

```
    # 2) X_train. train 데이터의 feature 입니다.
    # 3) y_train. train 데이터의 label 입니다.
    # 4) cv. Cross Validation에서 데이터를 조각낼(split) 갯수입니다. 총 20조각을 내야하기 때문
에 20을 대입합니다.
    # 5) scoring. 점수를 측정할 공식입니다. 앞서 구현한 RMSE를 적용합니다.
    # 마지막으로, 이 함수의 실행 결과의 평균(mean)을 구한 뒤 score라는 이름의 새로운 변수에 할당합니
다.
    score = cross_val_score(model, X_train, y_train, cv=20, scoring=rmse_score).
mean()

    # hyperparameter 탐색 결과를 딕셔너리화 합니다.
    hyperparameters = {
        'epoch': epoch,
        'score': score,
        'n_estimators': n_estimators,
        'max_depth': max_depth,
        'learning_rate': learning_rate,
        'subsample': subsample,
        'colsample_bylevel': colsample_bylevel,
        'colsample_bytree': colsample_bytree,
    }

    # hyperparameter 탐색 결과를 리스트에 저장합니다.
    finer_hyperparameters_list.append(hyperparameters)

    # hyperparameter 탐색 결과를 출력합니다.
    print(f"{epoch:2} n_estimators = {n_estimators}, max_depth = {max_depth:2},
 learning_rate = {learning_rate:.10f}, subsample = {subsample:.6f}, colsample_by
level = {colsample_bylevel:.6f}, colsample_bytree = {colsample_bytree:.6f}, Scor
e = {score:.5f}")

# finer_hyperparameters_list를 Pandas의 DataFrame으로 변환합니다.
finer_hyperparameters_list = pd.DataFrame.from_dict(finer_hyperparameters_list)

# 변환한 finer_hyperparameters_list를 score가 낮은 순으로 정렬합니다.
# (RMSE는 score가 낮을 수록 더 정확도가 높다고 가정합니다)
finer_hyperparameters_list = finer_hyperparameters_list.sort_values(by="score")

# finer_hyperparameters_list 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(finer_hyperparameters_list.shape)

# finer_hyperparameters_list의 상위 10개를 출력합니다.
finer_hyperparameters_list.head(10)
```

```
 0 n_estimators = 826, max_depth = 26, learning_rate = 0.0140914189,
subsample = 0.604440, colsample_bylevel = 0.803958, colsample_bytree
= 0.958276, Score = 0.33072
 1 n_estimators = 693, max_depth = 19, learning_rate = 0.0204163085,
subsample = 0.281243, colsample_bylevel = 0.842756, colsample_bytree
= 0.922510, Score = 0.32868
 2 n_estimators = 578, max_depth = 21, learning_rate = 0.0367275403,
subsample = 0.673859, colsample_bylevel = 0.694686, colsample_bytree
= 0.775183, Score = 0.34194
 3 n_estimators = 556, max_depth = 35, learning_rate = 0.0206988792,
subsample = 0.311399, colsample_bylevel = 0.730884, colsample_bytree
= 0.920663, Score = 0.32868
 4 n_estimators = 786, max_depth = 11, learning_rate = 0.0352740882,
subsample = 0.394929, colsample_bylevel = 0.497259, colsample_bytree
= 0.946960, Score = 0.33906
 5 n_estimators = 462, max_depth =  8, learning_rate = 0.0217484971,
subsample = 0.601801, colsample_bylevel = 0.435530, colsample_bytree
= 0.907742, Score = 0.32755
 6 n_estimators = 315, max_depth = 57, learning_rate = 0.0194657250,
subsample = 0.313923, colsample_bylevel = 0.845876, colsample_bytree
= 0.966426, Score = 0.32730
 7 n_estimators = 943, max_depth = 21, learning_rate = 0.0697425019,
subsample = 0.388384, colsample_bylevel = 0.547449, colsample_bytree
= 0.899230, Score = 0.34735
 8 n_estimators = 791, max_depth = 35, learning_rate = 0.0125012079,
subsample = 0.224484, colsample_bylevel = 0.677926, colsample_bytree
= 0.709716, Score = 0.33583
 9 n_estimators = 511, max_depth = 35, learning_rate = 0.0178360559,
subsample = 0.387557, colsample_bylevel = 0.623901, colsample_bytree
= 0.816696, Score = 0.33536
10 n_estimators = 898, max_depth = 21, learning_rate = 0.0324601521,
subsample = 0.264827, colsample_bylevel = 0.875968, colsample_bytree
= 0.713651, Score = 0.34599
11 n_estimators = 892, max_depth = 20, learning_rate = 0.0910118678,
subsample = 0.343863, colsample_bylevel = 0.896878, colsample_bytree
= 0.891140, Score = 0.34970
12 n_estimators = 406, max_depth = 14, learning_rate = 0.0121356539,
subsample = 0.526529, colsample_bylevel = 0.546882, colsample_bytree
= 0.752584, Score = 0.34846
13 n_estimators = 683, max_depth = 22, learning_rate = 0.6734879598,
subsample = 0.278793, colsample_bylevel = 0.832358, colsample_bytree
= 0.743327, Score = 0.82619
14 n_estimators = 565, max_depth = 56, learning_rate = 0.4208953969,
subsample = 0.501257, colsample_bylevel = 0.686633, colsample_bytree
= 0.919102, Score = 0.43834
15 n_estimators = 895, max_depth = 55, learning_rate = 0.0229405212,
subsample = 0.245841, colsample_bylevel = 0.720467, colsample_bytree
= 0.926856, Score = 0.33425
16 n_estimators = 441, max_depth =  5, learning_rate = 0.1557808729,
subsample = 0.201518, colsample_bylevel = 0.604204, colsample_bytree
= 0.789971, Score = 0.35362
17 n_estimators = 838, max_depth = 40, learning_rate = 0.3109060394,
subsample = 0.405133, colsample_bylevel = 0.874428, colsample_bytree
= 0.816343, Score = 0.39935
18 n_estimators = 426, max_depth = 56, learning_rate = 0.0980670795,
subsample = 0.535718, colsample_bylevel = 0.775129, colsample_bytree
= 0.742462, Score = 0.35147
19 n_estimators = 372, max_depth = 30, learning_rate = 0.1709669643,
subsample = 0.592810, colsample_bylevel = 0.496650, colsample_bytree
= 0.802357, Score = 0.36893
20 n_estimators = 636, max_depth =  9, learning_rate = 0.9526417597,
```

```
subsample = 0.398907, colsample_bylevel = 0.844800, colsample_bytree
= 0.733590, Score = 1.43410
21 n_estimators = 481, max_depth = 49, learning_rate = 0.5854043195,
subsample = 0.497239, colsample_bylevel = 0.782827, colsample_bytree
= 0.719362, Score = 0.56249
22 n_estimators = 864, max_depth = 22, learning_rate = 0.2735823453,
subsample = 0.335316, colsample_bylevel = 0.762807, colsample_bytree
= 0.792830, Score = 0.40885
23 n_estimators = 916, max_depth = 51, learning_rate = 0.0100133400,
subsample = 0.459323, colsample_bylevel = 0.695348, colsample_bytree
= 0.747032, Score = 0.33249
24 n_estimators = 640, max_depth = 23, learning_rate = 0.0499691162,
subsample = 0.685181, colsample_bylevel = 0.472997, colsample_bytree
= 0.925317, Score = 0.33838
25 n_estimators = 738, max_depth = 45, learning_rate = 0.0407032864,
subsample = 0.296325, colsample_bylevel = 0.894751, colsample_bytree
= 0.963906, Score = 0.33906
26 n_estimators = 933, max_depth = 29, learning_rate = 0.1886915092,
subsample = 0.477641, colsample_bylevel = 0.693674, colsample_bytree
= 0.958425, Score = 0.36226
27 n_estimators = 385, max_depth = 54, learning_rate = 0.0254351149,
subsample = 0.266720, colsample_bylevel = 0.831264, colsample_bytree
= 0.867055, Score = 0.32644
28 n_estimators = 701, max_depth = 18, learning_rate = 0.0146441339,
subsample = 0.284567, colsample_bylevel = 0.791547, colsample_bytree
= 0.716776, Score = 0.33701
29 n_estimators = 388, max_depth = 43, learning_rate = 0.0232468029,
subsample = 0.526676, colsample_bylevel = 0.558939, colsample_bytree
= 0.901665, Score = 0.33353
30 n_estimators = 715, max_depth = 57, learning_rate = 0.2204953869,
subsample = 0.455188, colsample_bylevel = 0.712494, colsample_bytree
= 0.994391, Score = 0.37042
31 n_estimators = 982, max_depth = 21, learning_rate = 0.0394635889,
subsample = 0.567685, colsample_bylevel = 0.850578, colsample_bytree
= 0.998446, Score = 0.33515
32 n_estimators = 554, max_depth = 34, learning_rate = 0.0167896270,
subsample = 0.425722, colsample_bylevel = 0.684372, colsample_bytree
= 0.971310, Score = 0.32618
33 n_estimators = 639, max_depth = 56, learning_rate = 0.0342627969,
subsample = 0.537765, colsample_bylevel = 0.614150, colsample_bytree
= 0.786516, Score = 0.34602
34 n_estimators = 345, max_depth = 42, learning_rate = 0.0156133893,
subsample = 0.688611, colsample_bylevel = 0.469881, colsample_bytree
= 0.799938, Score = 0.36028
35 n_estimators = 855, max_depth =  8, learning_rate = 0.0285197555,
subsample = 0.402216, colsample_bylevel = 0.547196, colsample_bytree
= 0.854189, Score = 0.32445
36 n_estimators = 538, max_depth = 44, learning_rate = 0.1027719906,
subsample = 0.343363, colsample_bylevel = 0.456526, colsample_bytree
= 0.729607, Score = 0.36177
37 n_estimators = 866, max_depth = 14, learning_rate = 0.4914480158,
subsample = 0.452086, colsample_bylevel = 0.571139, colsample_bytree
= 0.720022, Score = 0.51907
38 n_estimators = 565, max_depth = 10, learning_rate = 0.0267918529,
subsample = 0.510642, colsample_bylevel = 0.511964, colsample_bytree
= 0.940773, Score = 0.32947
39 n_estimators = 378, max_depth =  3, learning_rate = 0.1240507309,
subsample = 0.219938, colsample_bylevel = 0.678010, colsample_bytree
= 0.999816, Score = 0.33688
40 n_estimators = 839, max_depth = 12, learning_rate = 0.0385669369,
subsample = 0.266342, colsample_bylevel = 0.801249, colsample_bytree
```

```
= 0.961586, Score = 0.33678
41 n_estimators = 437, max_depth = 11, learning_rate = 0.0233644625,
subsample = 0.321998, colsample_bylevel = 0.561755, colsample_bytree
= 0.960961, Score = 0.32762
42 n_estimators = 923, max_depth = 28, learning_rate = 0.0725801194,
subsample = 0.334403, colsample_bylevel = 0.619528, colsample_bytree
= 0.731837, Score = 0.35865
43 n_estimators = 789, max_depth = 49, learning_rate = 0.9031698804,
subsample = 0.335432, colsample_bylevel = 0.493866, colsample_bytree
= 0.828910, Score = 1.23522
44 n_estimators = 475, max_depth = 13, learning_rate = 0.3570459630,
subsample = 0.580113, colsample_bylevel = 0.769912, colsample_bytree
= 0.752050, Score = 0.42210
45 n_estimators = 469, max_depth = 21, learning_rate = 0.0102538360,
subsample = 0.316122, colsample_bylevel = 0.555959, colsample_bytree
= 0.741816, Score = 0.35384
46 n_estimators = 502, max_depth = 38, learning_rate = 0.0111374803,
subsample = 0.358754, colsample_bylevel = 0.505624, colsample_bytree
= 0.973068, Score = 0.32991
47 n_estimators = 678, max_depth = 25, learning_rate = 0.0217827608,
subsample = 0.523825, colsample_bylevel = 0.808594, colsample_bytree
= 0.875922, Score = 0.33293
48 n_estimators = 926, max_depth = 39, learning_rate = 0.0417623298,
subsample = 0.538666, colsample_bylevel = 0.614637, colsample_bytree
= 0.715228, Score = 0.34953
49 n_estimators = 848, max_depth = 35, learning_rate = 0.0199923328,
subsample = 0.649909, colsample_bylevel = 0.618475, colsample_bytree
= 0.963445, Score = 0.33089
50 n_estimators = 660, max_depth = 53, learning_rate = 0.1744916417,
subsample = 0.429841, colsample_bylevel = 0.894033, colsample_bytree
= 0.908646, Score = 0.36765
51 n_estimators = 323, max_depth = 53, learning_rate = 0.2742895948,
subsample = 0.317018, colsample_bylevel = 0.618159, colsample_bytree
= 0.861279, Score = 0.42262
52 n_estimators = 575, max_depth =  3, learning_rate = 0.0504267387,
subsample = 0.655626, colsample_bylevel = 0.572267, colsample_bytree
= 0.911976, Score = 0.33836
53 n_estimators = 646, max_depth =  6, learning_rate = 0.4478130499,
subsample = 0.677061, colsample_bylevel = 0.863480, colsample_bytree
= 0.964276, Score = 0.40714
54 n_estimators = 805, max_depth =  2, learning_rate = 0.1437046472,
subsample = 0.641922, colsample_bylevel = 0.468121, colsample_bytree
= 0.877805, Score = 0.38581
55 n_estimators = 864, max_depth = 38, learning_rate = 0.0257201746,
subsample = 0.582692, colsample_bylevel = 0.791519, colsample_bytree
= 0.905579, Score = 0.33344
56 n_estimators = 497, max_depth =  3, learning_rate = 0.0308144687,
subsample = 0.286670, colsample_bylevel = 0.416176, colsample_bytree
= 0.941399, Score = 0.42473
57 n_estimators = 359, max_depth = 38, learning_rate = 0.0148173959,
subsample = 0.656984, colsample_bylevel = 0.856412, colsample_bytree
= 0.706486, Score = 0.36368
58 n_estimators = 653, max_depth = 23, learning_rate = 0.0689673741,
subsample = 0.578014, colsample_bylevel = 0.440388, colsample_bytree
= 0.709141, Score = 0.36057
59 n_estimators = 936, max_depth = 28, learning_rate = 0.0139820517,
subsample = 0.638780, colsample_bylevel = 0.727529, colsample_bytree
= 0.852536, Score = 0.33051
60 n_estimators = 961, max_depth = 42, learning_rate = 0.0494575173,
subsample = 0.443794, colsample_bylevel = 0.435667, colsample_bytree
= 0.842521, Score = 0.34863
```

```
61 n_estimators = 941, max_depth = 37, learning_rate = 0.0795945844,
subsample = 0.639516, colsample_bylevel = 0.814327, colsample_bytree
= 0.714473, Score = 0.37009
62 n_estimators = 316, max_depth = 25, learning_rate = 0.8171440833,
subsample = 0.503361, colsample_bylevel = 0.870639, colsample_bytree
= 0.811066, Score = 0.68252
63 n_estimators = 608, max_depth = 20, learning_rate = 0.0122782175,
subsample = 0.615252, colsample_bylevel = 0.513641, colsample_bytree
= 0.788186, Score = 0.33737
64 n_estimators = 326, max_depth =  8, learning_rate = 0.0466216280,
subsample = 0.684720, colsample_bylevel = 0.679378, colsample_bytree
= 0.901412, Score = 0.32392
65 n_estimators = 732, max_depth = 34, learning_rate = 0.1545358630,
subsample = 0.466161, colsample_bylevel = 0.600334, colsample_bytree
= 0.730037, Score = 0.36813
66 n_estimators = 484, max_depth =  2, learning_rate = 0.5748964666,
subsample = 0.622408, colsample_bylevel = 0.518681, colsample_bytree
= 0.773039, Score = 0.35794
67 n_estimators = 612, max_depth = 32, learning_rate = 0.1991645342,
subsample = 0.272800, colsample_bylevel = 0.874424, colsample_bytree
= 0.922642, Score = 0.38911
68 n_estimators = 373, max_depth = 26, learning_rate = 0.8023382663,
subsample = 0.627920, colsample_bylevel = 0.430611, colsample_bytree
= 0.800935, Score = 0.53953
69 n_estimators = 338, max_depth = 45, learning_rate = 0.1151012629,
subsample = 0.572554, colsample_bylevel = 0.749211, colsample_bytree
= 0.804116, Score = 0.35725
70 n_estimators = 375, max_depth = 10, learning_rate = 0.0912986042,
subsample = 0.632452, colsample_bylevel = 0.457623, colsample_bytree
= 0.995132, Score = 0.34907
71 n_estimators = 654, max_depth = 17, learning_rate = 0.3985748388,
subsample = 0.610973, colsample_bylevel = 0.404381, colsample_bytree
= 0.996095, Score = 0.42605
72 n_estimators = 876, max_depth = 52, learning_rate = 0.3130205122,
subsample = 0.352322, colsample_bylevel = 0.614967, colsample_bytree
= 0.947566, Score = 0.42314
73 n_estimators = 840, max_depth = 11, learning_rate = 0.0827328462,
subsample = 0.549684, colsample_bylevel = 0.652479, colsample_bytree
= 0.931786, Score = 0.34405
74 n_estimators = 863, max_depth = 11, learning_rate = 0.2574509349,
subsample = 0.306930, colsample_bylevel = 0.620301, colsample_bytree
= 0.804716, Score = 0.42218
75 n_estimators = 876, max_depth = 44, learning_rate = 0.2955270232,
subsample = 0.686000, colsample_bylevel = 0.527141, colsample_bytree
= 0.725916, Score = 0.41859
76 n_estimators = 494, max_depth = 48, learning_rate = 0.0183072819,
subsample = 0.662914, colsample_bylevel = 0.446013, colsample_bytree
= 0.717283, Score = 0.35233
77 n_estimators = 985, max_depth =  3, learning_rate = 0.0718847944,
subsample = 0.285630, colsample_bylevel = 0.790567, colsample_bytree
= 0.925351, Score = 0.32769
78 n_estimators = 618, max_depth = 21, learning_rate = 0.0132333903,
subsample = 0.260413, colsample_bylevel = 0.831675, colsample_bytree
= 0.813305, Score = 0.33106
79 n_estimators = 578, max_depth =  2, learning_rate = 0.5614038526,
subsample = 0.470569, colsample_bylevel = 0.406692, colsample_bytree
= 0.960557, Score = 0.36013
80 n_estimators = 453, max_depth = 39, learning_rate = 0.2792367957,
subsample = 0.428404, colsample_bylevel = 0.791115, colsample_bytree
= 0.846710, Score = 0.39964
81 n_estimators = 341, max_depth =  5, learning_rate = 0.0286532695,
```

```
subsample = 0.575951, colsample_bylevel = 0.524058, colsample_bytree
= 0.748711, Score = 0.33779
82 n_estimators = 704, max_depth = 38, learning_rate = 0.0212829787,
subsample = 0.262961, colsample_bylevel = 0.856746, colsample_bytree
= 0.988876, Score = 0.33093
83 n_estimators = 457, max_depth = 38, learning_rate = 0.0356493505,
subsample = 0.406002, colsample_bylevel = 0.482871, colsample_bytree
= 0.903674, Score = 0.33723
84 n_estimators = 525, max_depth = 33, learning_rate = 0.2113503066,
subsample = 0.651353, colsample_bylevel = 0.720116, colsample_bytree
= 0.765657, Score = 0.37206
85 n_estimators = 884, max_depth = 38, learning_rate = 0.0806559339,
subsample = 0.578291, colsample_bylevel = 0.514824, colsample_bytree
= 0.898813, Score = 0.34954
86 n_estimators = 542, max_depth = 48, learning_rate = 0.0897100196,
subsample = 0.672743, colsample_bylevel = 0.768915, colsample_bytree
= 0.761242, Score = 0.35483
87 n_estimators = 657, max_depth = 44, learning_rate = 0.0216305583,
subsample = 0.600683, colsample_bylevel = 0.493772, colsample_bytree
= 0.998632, Score = 0.33300
88 n_estimators = 550, max_depth = 11, learning_rate = 0.0118268256,
subsample = 0.414797, colsample_bylevel = 0.468924, colsample_bytree
= 0.920813, Score = 0.32904
89 n_estimators = 860, max_depth = 55, learning_rate = 0.0146506694,
subsample = 0.533267, colsample_bylevel = 0.703759, colsample_bytree
= 0.924415, Score = 0.32928
90 n_estimators = 876, max_depth = 30, learning_rate = 0.1434107525,
subsample = 0.518028, colsample_bylevel = 0.426024, colsample_bytree
= 0.959858, Score = 0.35903
91 n_estimators = 403, max_depth = 51, learning_rate = 0.2341096236,
subsample = 0.348113, colsample_bylevel = 0.444010, colsample_bytree
= 0.947844, Score = 0.39720
92 n_estimators = 859, max_depth = 10, learning_rate = 0.0111051573,
subsample = 0.362207, colsample_bylevel = 0.870122, colsample_bytree
= 0.980396, Score = 0.32133
93 n_estimators = 815, max_depth = 51, learning_rate = 0.0161951484,
subsample = 0.214376, colsample_bylevel = 0.407823, colsample_bytree
= 0.740139, Score = 0.33541
94 n_estimators = 560, max_depth = 41, learning_rate = 0.1017657375,
subsample = 0.579904, colsample_bylevel = 0.827769, colsample_bytree
= 0.747174, Score = 0.35405
95 n_estimators = 597, max_depth = 14, learning_rate = 0.0184553028,
subsample = 0.597537, colsample_bylevel = 0.747206, colsample_bytree
= 0.975083, Score = 0.32881
96 n_estimators = 349, max_depth = 28, learning_rate = 0.0265883410,
subsample = 0.392322, colsample_bylevel = 0.667323, colsample_bytree
= 0.948886, Score = 0.32822
97 n_estimators = 909, max_depth = 47, learning_rate = 0.0362947251,
subsample = 0.337482, colsample_bylevel = 0.654701, colsample_bytree
= 0.864936, Score = 0.33585
98 n_estimators = 378, max_depth =  2, learning_rate = 0.0272153858,
subsample = 0.257277, colsample_bylevel = 0.802665, colsample_bytree
= 0.703375, Score = 0.54857
99 n_estimators = 369, max_depth = 22, learning_rate = 0.0345952079,
subsample = 0.478914, colsample_bylevel = 0.768151, colsample_bytree
= 0.711363, Score = 0.34617
(100, 8)
```

Out[38]:

	colsample_bylevel	colsample_bytree	epoch	learning_rate	max_depth	n_estimat
92	0.870122	0.980396	92	0.011105	10	859
64	0.679378	0.901412	64	0.046622	8	326
35	0.547196	0.854189	35	0.028520	8	855
32	0.684372	0.971310	32	0.016790	34	554
27	0.831264	0.867055	27	0.025435	54	385
6	0.845876	0.966426	6	0.019466	57	315
5	0.435530	0.907742	5	0.021748	8	462
41	0.561755	0.960961	41	0.023364	11	437
77	0.790567	0.925351	77	0.071885	3	985
96	0.667323	0.948886	96	0.026588	28	349

탐색 결과 다음의 하이퍼패러미터가 가장 좋은 하이퍼패러미터라는 사실을 발견할 수 있습니다.

In [72]:

```
# 가장 score가 낮게 나온(=좋은 정확도가 나온) 하이퍼패러미터를 가져옵니다.
# 이를 best_hyperparameters라는 이름의 변수에 저장합니다.
best_hyperparameters = finer_hyperparameters_list.iloc[0]

# best_hyperparameters에서 n_estimators 하이퍼패러미터만 가져옵니다.
# 이를 best_n_estimators라는 이름의 변수에 저장합니다.
# 주의: n_estimators는 무조건 정수형 값(int)이어야 하기 때문에, 정수형으로 타입 변환을 해줍니다.
best_n_estimators = int(best_hyperparameters["n_estimators"])

# best_hyperparameters에서 max_depth 하이퍼패러미터만 가져옵니다.
# 이를 best_max_depth라는 이름의 변수에 저장합니다.
# 주의: max_depth는 무조건 정수형 값(int)이어야 하기 때문에, 정수형으로 타입 변환을 해줍니다.
best_max_depth = int(best_hyperparameters["max_depth"])

# best_hyperparameters에서 learning_rate 하이퍼패러미터만 가져옵니다.
# 이를 best_learning_rate라는 이름의 변수에 저장합니다.
best_learning_rate = best_hyperparameters["learning_rate"]

# best_hyperparameters에서 subsample 하이퍼패러미터만 가져옵니다.
# 이를 best_subsample라는 이름의 변수에 저장합니다.
best_subsample = best_hyperparameters["subsample"]

# best_hyperparameters에서 colsample_bytree 하이퍼패러미터만 가져옵니다.
# 이를 best_colsample_bytree라는 이름의 변수에 저장합니다.
best_colsample_bytree = best_hyperparameters["colsample_bytree"]

# best_hyperparameters에서 colsample_bylevel 하이퍼패러미터만 가져옵니다.
# 이를 best_colsample_bylevel라는 이름의 변수에 저장합니다.
best_colsample_bylevel = best_hyperparameters["colsample_bylevel"]

# best_hyperparameters들을 출력합니다.
print(f"n_estimators(best) = {best_n_estimators}, max_depth(best) = {best_max_depth}, learning_rate(best) = {best_learning_rate:.6f}, subsample(best) = {best_subsample:.6f}, colsample_bytree(best) = {best_colsample_bytree:.6f}, colsample_bylevel(best) = {best_colsample_bylevel:.6f}")
```

```
n_estimators(best) = 859, max_depth(best) = 10, learning_rate(best)
= 0.011105, subsample(best) = 0.362207, colsample_bytree(best) = 0.9
80396, colsample_bylevel(best) = 0.870122
```

Use Gradient Boosting Machine

Hyperparameter Tuning으로 만족스러운 하이퍼패러미터를 찾았다면, 이제 이 하이퍼패러미터를 활용하여 머신러닝 모델을 학습할 시간입니다.

이번에 사용할 알고리즘은 그래디언트 부스팅 머신(Gradient Boosting Machine)입니다. 그래디언트 부스팅 머신은 의사결정나무(Decision Tree)에 그래디언트 부스팅(Boosting Machine)이라는 알고리즘을 적용한 모델인데, 구조화된 데이터(Structured Data)에 한해서는 가장 강력한 머신러닝 알고리즘이라고 불리우고 있습니다. 알고리즘의 동작 원리는 다음과 같습니다.

1. 의사결정나무(Decision Tree)를 하나 학습합니다.
2. 1번에서 학습한 의사결정나무를 통해, 학습(train)데이터를 예측합니다. 그리고 예측값과 정답의 차이(residual)를 계산합니다.
3. 위 차이(residual)를 보정하는 또 하나의 의사결정나무(Decision Tree)를 학습합니다. 두 번째 의사결정나무에서는 차이를 입력값으로 받고, 차이를 보정하기 위해서는 얼만큼의 보정값이 필요한지를 예측합니다.
4. 위 방식을 끊임없이 반복합니다.

이러한 방식을 거치면 의사결정나무(Decision Tree)보다 더 강력한 알고리즘을 구현할 수 있습니다. 자세한 설명은 다음의 링크들을 참고해주세요.

- A Kaggle Master Explains Gradient Boosting (http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/)
- Gradient Boosting from scratch (https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d)
- A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning (https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/)

이번에는 가장 강력한 그래디언트 부스팅 머신(Gradient Boosting Machine) 구현체중 하나인 XGBoost (https://github.com/dmlc/xgboost)를 사용하겠습니다. XGBoost의 회귀(Regression)용 머신러닝 모델인 `XGBRegressor`를 가져올텐데, 이 `XGBRegressor`에는 크게 두 가지 기능이 있습니다.

- **fit**: 머신러닝 알고리즘을 학습시킵니다. 전문용어로 fitting한다고 하기 때문에 fit이라는 표현을 사용합니다. fit을 하기 위해서는 train 데이터가 필요하며, 정확히는 train 데이터의 feature(X_train)와 label(y_train)이 필요합니다.
- **predict**: **fit**이 끝나면, 이후에 **predict**를 통해 예측을 할 수 있습니다. predict를 하기 위해서는 test 데이터가 필요하며, 정확히는 test 데이터의 feature(X_test)가 필요합니다.

In [73]:

```
# Gradient Boosting Machine 패키지인 XGBoost를 가져옵니다.
# 이를 xgb라는 축약어로 사용합니다.
import xgboost as xgb

# 주의: 혹시 하이퍼패러미터 튜닝을 하는데 시간이 너무 오래 걸린다면,
# 이를 대신해서 다음의 하이퍼패러미터를 사용해주세요. (아래 줄의 주석을 풀면 됩니다)
# best_n_estimators = 859
# best_max_depth = 10
# best_learning_rate = 0.011105
# best_subsample = 0.362207
# best_colsample_bytree = 0.980396
# best_colsample_bylevel = 0.870122

# XGBRegressor를 생성합니다. 실행할때는 다음의 옵션이 들어갑니다.
# 1) n_estimators. 트리의 갯수입니다. 지정한 갯수만큼 트리를 생성합니다.
# 2) max_depth. 트리의 깊이입니다. 지정한 숫자만큼 트리가 깊게 가지를 뻗습니다.
# 3) learning_rate. 각 트리마다의 비중을 나타냅니다. 너무 작으면 과적합(overfitting)될 가능성이
 있고, 너무 높으면 부적합(underfitting)될 가능성이 있습니다.
# 4) subsample. 하나의 트리를 만들 때 사용할 데이터의 비율을 나타냅니다. 0.0 ~ 1.0 사이의 값을 넣
으면 지정한 비율만큼만 랜덤하게 데이터를 사용합니다.
# 5) colsample_bytree. 하나의 트리를 만들 때 사용할 feature의 비율을 나타냅니다. 0.0 ~ 1.0 사
이의 값을 넣으면 트리를 만들 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
# 6) colsample_bylevel. 트리가 한 번 가지를 칠 때 사용할 feature의 비율을 나타냅니다. 0.0 ~
 1.0 사이의 값을 넣으면 트리가 가지를 칠 때 지정한 비율만큼만 랜덤하게 feature를 사용합니다.
# 생성한 XGBRegressor를 model이라는 이름의 변수에 대입합니다.
model = xgb.XGBRegressor(n_estimators=best_n_estimators,
                         max_depth=best_max_depth,
                         learning_rate=best_learning_rate,
                         subsample=best_subsample,
                         colsample_bytree=best_colsample_bytree,
                         colsample_bylevel=best_colsample_bylevel,
                         seed=37)

model
```

Out[73]:

```
XGBRegressor(base_score=0.5, colsample_bylevel=0.87012248132277636,
       colsample_bytree=0.98039571556042127, gamma=0,
       learning_rate=0.011105157251072018, max_delta_step=0, max_dep
th=10,
       min_child_weight=1, missing=None, n_estimators=859, nthread=-
1,
       objective='reg:linear', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=37, silent=True,
       subsample=0.36220688378150445)
```

Fit

이제 앞서 설명한 머신러닝 모델을 학습시켜보겠습니다. 머신러닝 모델을 학습시킬때는 `fit` 함수를 사용합니다. 학습을 할 때는 1) train 데이터의 feature인 `X_train`, 그리고 2) train 데이터의 label인 `y_train`이 필요합니다.

In [74]:

```
# XGBRegressor를 학습(fitting)합니다.
# 학습에는 fit 이라는 기능을 사용하며, train 데이터의 feature(X_train)와 label(y_train)을 집어넣습니다.
model.fit(X_train, y_train)
```

Out[74]:

```
XGBRegressor(base_score=0.5, colsample_bylevel=0.87012248132277636,
       colsample_bytree=0.98039571556042127, gamma=0,
       learning_rate=0.011105157251072018, max_delta_step=0, max_dep
th=10,
       min_child_weight=1, missing=None, n_estimators=859, nthread=-
1,
       objective='reg:linear', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=37, silent=True,
       subsample=0.36220688378150445)
```

Predict

머신러닝 모델이 성공적으로 학습이 되었다면, 남은 것은 이 모델를 활용해 test 데이터에 있는 자전거 대여량을 예측하는 것입니다. 예측은 `model.predict`로 할 수 있으며, 이 때 test 데이터의 feature인 `X_test`가 필요합니다.

In [75]:

```
# fit이 끝났으면, predict라는 기능을 사용하여 log transformation한 자전거 대여량(log_count)을 예측합니다.
# log_predictions의 실행이 끝나면 test 데이터의 log transformation한 자전거 대여량(log_count)을 반환하며,
# 이를 predictions라는 이름의 변수에 할당합니다.
log_predictions = model.predict(X_test)

# log_predictions 변수에 할당된 데이터의 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시되나, column이 없기 때문에 (row,) 형태로 표시될 것입니다.
print(log_predictions.shape)

# log_predictions 변수를 출력합니다.
log_predictions
```

```
(6493,)
```

Out[75]:

```
array([ 2.71734762,  1.71947825,  1.47986841, ...,  4.71064758,
        4.44729996,  3.79223394], dtype=float32)
```

앞서 데이터를 분석할 때 설명한대로, **머신러닝 모델에서 예측한 것은 자전거 대여량(count)이 아닌 log transformation 한 자전거 대여량(log_count)입니다.** 이를 다시 자전거 대여량(count)으로 원상복구 하기 위해 exp (https://en.wikipedia.org/wiki/Exponential_function)를 사용하겠습니다.

In [76]:

```
# log transformation한 자전거 대여량(log_count)을 다시 exp로 원상복귀 합니다.
# (=자연로그는 exp로 없애버릴 수 있습니다)
# 이를 predictions라는 새로운 변수에 할당합니다.
predictions = np.exp(log_predictions) - 1

# predictions 변수에 할당된 데이터의 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시되나, column이 없기 때문에 (row,) 형태로 표시될 것입니다.
print(predictions.shape)

# predictions 변수를 출력합니다.
predictions
```

```
(6493,)
```

Out[76]:

```
array([  14.14011192,    4.58161545,    3.39236784, ...,  110.124099
73,
         84.39605713,   43.3553772 ], dtype=float32)
```

Submit

머신러닝 모델의 fit과 predict 를 통해 우리는 test 데이터에 있는 자전거 대여량(count)을 예측하였습니다. 이제 우리에게 남은 건 이를 캐글(kaggle (http://kaggle.com/))이 권장하는 제출(submission) 포멧에 맞게 정리한 뒤 파일로 저장하는 것입니다.

캐글의 Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand) 경진대회에서는 **sampleSubmission.csv**라는 제출 포멧을 제공합니다. (다운로드 링크 (https://www.kaggle.com/c/bike-sharing-demand/data)) 우리는 우리가 예측한 값을 이 제출 포멧에 맞게 집어넣고 저장할 것입니다.

In [77]:

```
# 캐글이 제공하는 제출 포멧(sampleSubmission.csv)을 읽어옵니다.
# 이를 submission 이라는 이름의 변수에 할당합니다.
submission = pd.read_csv("data/bike/sampleSubmission.csv")

# submission 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(submission.shape)

# submission 데이터의 상위 5개를 띄웁니다.
submission.head()
```

(6493, 2)

Out[77]:

	datetime	count
0	2011-01-20 00:00:00	0
1	2011-01-20 01:00:00	0
2	2011-01-20 02:00:00	0
3	2011-01-20 03:00:00	0
4	2011-01-20 04:00:00	0

In [78]:

```
# 제출 포멧(submission)의 자전거 대여량(count) 컬럼에 우리의 예측값(predictions)를 집어넣습니
다.
# 두 데이터 모두 길이가 6493개로 동일하기 때문에, 등호(=)를 통해 쉽게 예측값을 넣을 수 있습니다.
submission["count"] = predictions

# submission 변수에 할당된 데이터의 행렬 사이즈를 출력합니다.
# 출력은 (row, column) 으로 표시됩니다.
print(submission.shape)

# submission 데이터의 상위 5개를 띄웁니다.
submission.head()
```

(6493, 2)

Out[78]:

	datetime	count
0	2011-01-20 00:00:00	14.140112
1	2011-01-20 01:00:00	4.581615
2	2011-01-20 02:00:00	3.392368
3	2011-01-20 03:00:00	2.394218
4	2011-01-20 04:00:00	1.855429

In [79]:

```
# 마지막으로 submission 변수에 들어간 값을 csv 형식의 데이터로 저장합니다.
submission.to_csv("data/bike/xgboost_0.37486.csv", index=False)
```

이제 캐글의 제출 페이지(Late Submission) (https://www.kaggle.com/c/bike-sharing-demand/submit)로 이동해 **xgboost_0.37486.csv** 파일을 제출하면 점수를 확인할 수 있습니다.

In [ ]:

```
```