1. Exploratory Data Analysis

November 21, 2018

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

0.1 Load Data

```
In [2]: !ls ../data/raw
```

```
TBSC2-20181112T130549Z-001.zip tubes2_HeartDisease_train.csv
description.xlsx               ~$description.xlsx
tubes2_HeartDisease_test.csv
```

0.1.1 Initialize Path Constants

```
In [3]: RAW_DATA_PATH = '../data/raw'
```

0.1.2 Load CSV File

```
In [4]: train_df = pd.read_csv('{}/tubes2_HeartDisease_train.csv'.format(RAW_DATA_PATH))
        test_df = pd.read_csv('{}/tubes2_HeartDisease_test.csv'.format(RAW_DATA_PATH))
```

```
In [5]: train_df.head()
```

Out[5]:

	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	\
0	54	1	4	125	216	0	0	140	0	
1	55	1	4	158	217	0	0	110	1	
2	54	0	3	135	304	1	0	170	0	
3	48	0	3	120	195	0	0	125	0	
4	50	1	4	120	0	0	1	156	1	

	Column10	Column11	Column12	Column13	Column14
0	0	?	?	?	1
1	2.5	2	?	?	1
2	0	1	0	3	0
3	0	?	?	?	0
4	0	1	?	6	3

```
In [6]: test_df.head()
```

```
Out[6]:    Column1  Column2  Column3 Column4 Column5 Column6  Column7 Column8 Column9  \
        0       60        1        2     160     267       1        1     157       0
        1       61        1        4     148     203       0        0     161       0
        2       54        1        4     130     242       0        0      91       1
        3       48        1        4     120     260       0        0     115       0
        4       57        0        1     130     308       0        0      98       0

          Column10 Column11 Column12 Column13
        0      0.5        2        ?        ?
        1        0        1        1        7
        2        1        2        ?        ?
        3        2        2        ?        ?
        4        1        2        ?        ?
```

0.2 Rename Column Names

So it's easier to read..

```
In [7]: from copy import deepcopy

        test_columns_replacement = {
            'Column1': 'age',
            'Column2': 'sex',
            'Column3': 'chest_pain_type',
            'Column4': 'resting_blood_pressure',
            'Column5': 'serum_cholestrol',
            'Column6': 'fasting_blood_sugar',
            'Column7': 'resting_ECG',
            'Column8': 'max_heart_rate_achieved',
            'Column9': 'excercise_induced_angina',
            'Column10': 'ST_depression',
            'Column11': 'peak_exercise_ST_segment',
            'Column12': 'num_of_major_vessels',
            'Column13': 'thal',
        }

        train_columns_replacement = test_columns_replacement.copy()
        train_columns_replacement['Column14'] = 'heart_disease_diagnosis'

        train_df = train_df.rename(columns=train_columns_replacement)

        test_df = test_df.rename(columns=test_columns_replacement)

In [8]: train_df.head()
```

```
Out[8]:    age  sex  chest_pain_type resting_blood_pressure serum_cholestrol  \
        0   54    1                4                    125              216
```

```
1   55    1               4                   158              217
2   54    0               3                   135              304
3   48    0               3                   120              195
4   50    1               4                   120                0

  fasting_blood_sugar resting_ECG max_heart_rate_achieved  \
0                   0           0                     140
1                   0           0                     110
2                   1           0                     170
3                   0           0                     125
4                   0           1                     156

  excercise_induced_angina ST_depression peak_exercise_ST_segment  \
0                        0             0                        ?
1                        1           2.5                        2
2                        0             0                        1
3                        0             0                        ?
4                        1             0                        1

  num_of_major_vessels thal  heart_disease_diagnosis
0                    ?    ?                        1
1                    ?    ?                        1
2                    0    3                        0
3                    ?    ?                        0
4                    ?    6                        3
```

In [9]:
```
test_df.head()
```

Out[9]:
```
   age  sex  chest_pain_type resting_blood_pressure serum_cholestrol  \
0   60    1                2                   160              267
1   61    1                4                   148              203
2   54    1                4                   130              242
3   48    1                4                   120              260
4   57    0                1                   130              308

  fasting_blood_sugar  resting_ECG max_heart_rate_achieved  \
0                   1           1                     157
1                   0           0                     161
2                   0           0                      91
3                   0           0                     115
4                   0           0                      98

  excercise_induced_angina ST_depression peak_exercise_ST_segment  \
0                        0           0.5                        2
1                        0             0                        1
2                        1             1                        2
3                        0             2                        2
4                        0             1                        2
```

```
  num_of_major_vessels thal
0                    ?    ?
1                    1    7
2                    ?    ?
3                    ?    ?
4                    ?    ?
```

0.3 Exploratory Data Analysis

0.3.1 Check For Null Values

Nan values null

```
In [10]: train_df.isnull().sum()
```

```
Out[10]: age                         0
         sex                         0
         chest_pain_type             0
         resting_blood_pressure      0
         serum_cholestrol            0
         fasting_blood_sugar         0
         resting_ECG                 1
         max_heart_rate_achieved     0
         excercise_induced_angina    0
         ST_depression               0
         peak_exercise_ST_segment    0
         num_of_major_vessels        0
         thal                        0
         heart_disease_diagnosis     0
         dtype: int64
```

```
In [11]: test_df.isnull().sum()
```

```
Out[11]: age                         0
         sex                         0
         chest_pain_type             0
         resting_blood_pressure      0
         serum_cholestrol            0
         fasting_blood_sugar         0
         resting_ECG                 0
         max_heart_rate_achieved     0
         excercise_induced_angina    0
         ST_depression               0
         peak_exercise_ST_segment    0
         num_of_major_vessels        0
         thal                        0
         dtype: int64
```

Dataset specific null type Column with values '?' in the dataset is null values a well

```
In [12]: def pad_text(text, target_length):
             assert(len(text) <= target_length)
             return text + (' ' * (target_length - len(text)))

         def print_data_null_encoded(data):
             for column in data.columns:
                 null_values = data[column].apply(lambda x: x == '?')
                 print('{}: {} ({} %)'.format(pad_text(column, 25),
                                              sum(null_values),
                                              sum(null_values) * 100 / data.shape[0]))
```

```
In [13]: print('==== Train Data ====')
         print_data_null_encoded(train_df)

         print('\n\n')

         print('==== Test Data ====')
         print_data_null_encoded(test_df)
```

```
==== Train Data ====
age                       : 0 (0.0 %)
sex                       : 0 (0.0 %)
chest_pain_type           : 0 (0.0 %)
resting_blood_pressure    : 47 (6.033376123234916 %)
serum_cholestrol          : 24 (3.0808729139922977 %)
fasting_blood_sugar       : 78 (10.012836970474968 %)
resting_ECG               : 1 (0.12836970474967907 %)
max_heart_rate_achieved   : 44 (5.648267008985879 %)
excercise_induced_angina  : 44 (5.648267008985879 %)
ST_depression             : 49 (6.290115532734275 %)
peak_exercise_ST_segment  : 262 (33.632862644415916 %)
num_of_major_vessels      : 514 (65.98202824133504 %)
thal                      : 408 (52.374839537869065 %)
heart_disease_diagnosis   : 0 (0.0 %)



==== Test Data ====
age                       : 0 (0.0 %)
sex                       : 0 (0.0 %)
chest_pain_type           : 0 (0.0 %)
resting_blood_pressure    : 12 (8.51063829787234 %)
serum_cholestrol          : 6 (4.25531914893617 %)
fasting_blood_sugar       : 12 (8.51063829787234 %)
resting_ECG               : 0 (0.0 %)
max_heart_rate_achieved   : 11 (7.801418439716312 %)
```

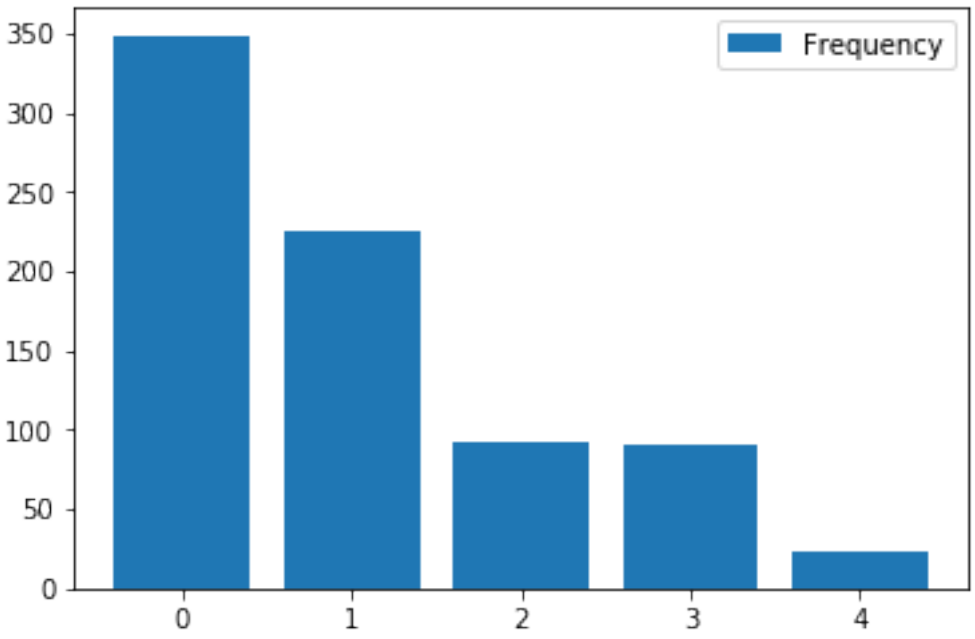
```
excercise_induced_angina : 11 (7.801418439716312 %)
ST_depression            : 13 (9.21985815602837 %)
peak_exercise_ST_segment : 47 (33.333333333333336 %)
num_of_major_vessels     : 97 (68.79432624113475 %)
thal                     : 78 (55.319148936170215 %)
```

The above data shows that some of the columns (*num_of_major_vessels*,*thal*) majority values are null. Null values cannot be used in the model and need to be solved. There are several methods to solve this null value problem. One of the methods is imputing null values to some value that could represent the data well.

0.3.2 Attribute Analysis

Heart Disease Diagnosis

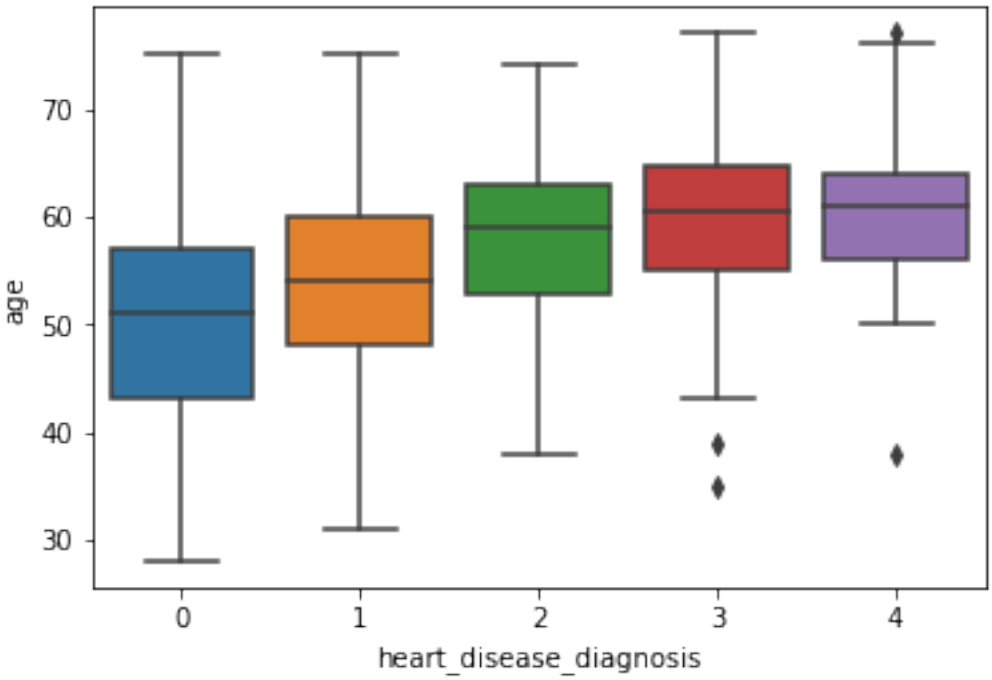
```
In [14]: label_count = {}
         label = train_df['heart_disease_diagnosis']

         for i in range(5):
             label_count[str(i)] = label.loc[label ==i].count()

         plt.bar(label_count.keys(), [label_count[str(i)] for i in range(5)], label="Frequency"
         leg = plt.legend()
         plt.show()
```



The figure above shows the distribution of the labels of the data. We can see that it declines as the label values increases. This implies that the worse the diagnosis is, the occurence of the diagnosis is much rarer.

Age

```
In [15]: sns.boxplot(x="heart_disease_diagnosis", y="age", data=train_df)
         plt.show()
```

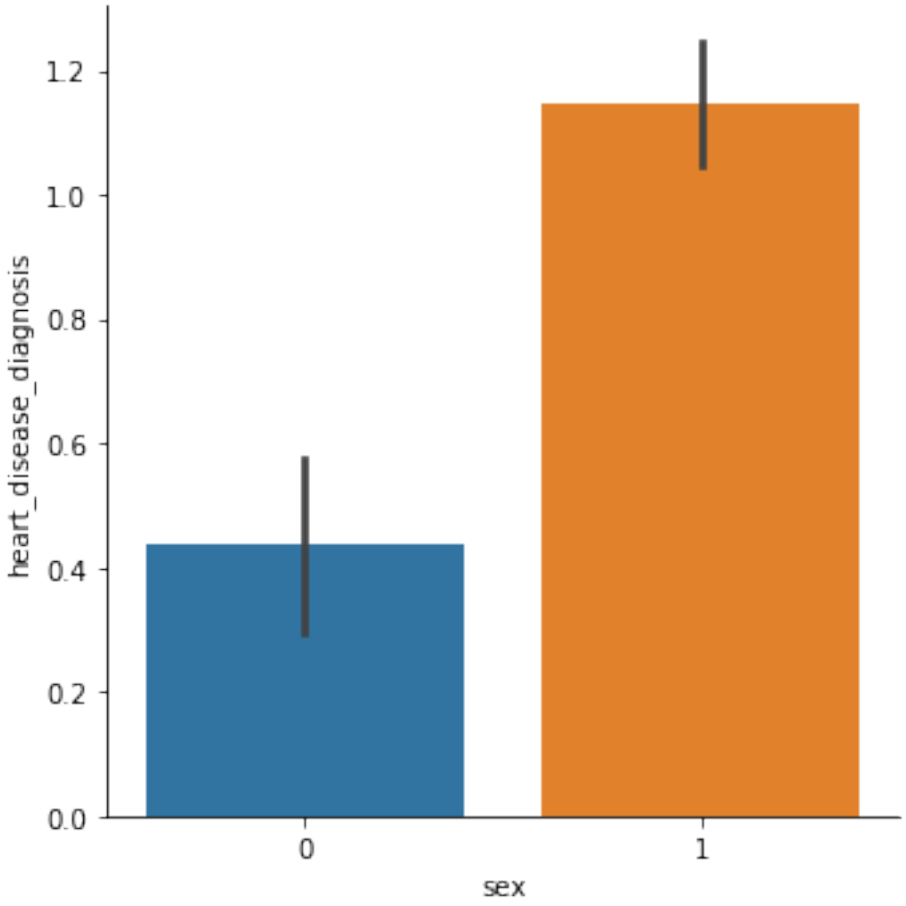


The figure above shows the boxplot of the age of the diagnosed people. We can see that the median value of each group of heart diagnosis increases as the diagnosis goes worse. This implies that the older the person is, there is much more chance to be diagnosed a worse heart disease diagnosis.

0.3.3 Sex

```
In [16]: sns.catplot(x="sex", y="heart_disease_diagnosis", data=train_df, kind='bar')
         plt.show()
```

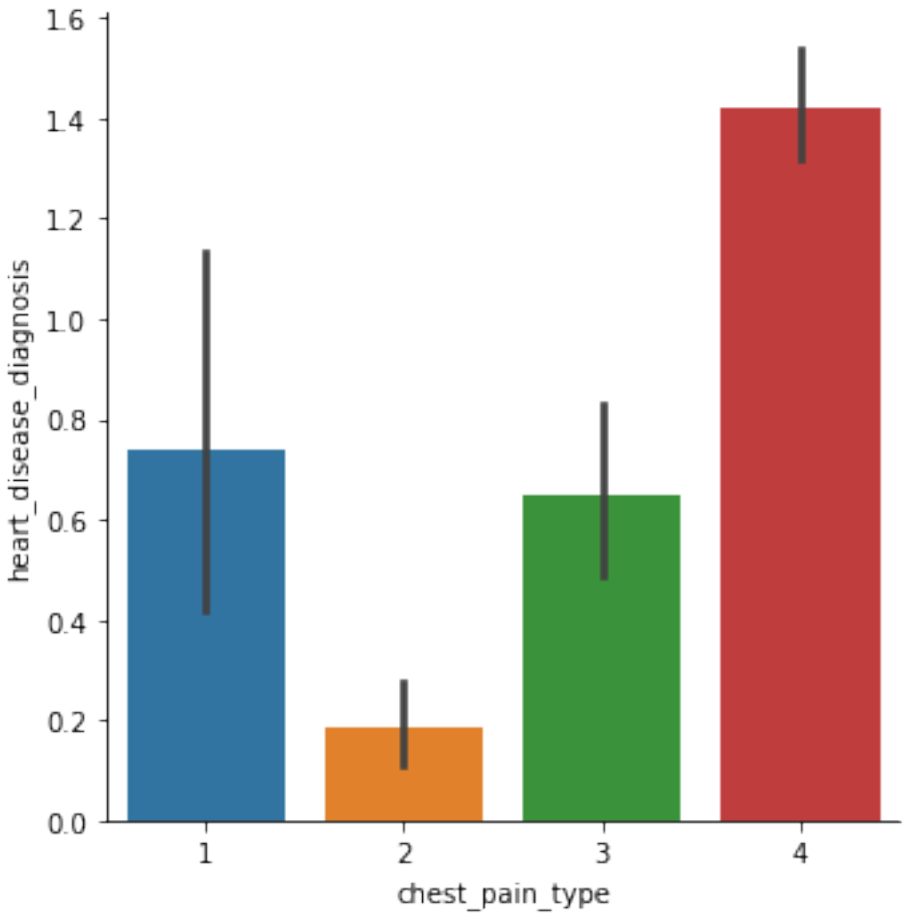
```
/usr/local/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



The figure above shows the catplot of the sex of the diagnosed people. Sex number 1 (male) is much more prone to the worse heart disease diagnosis.

Chest Pain Type

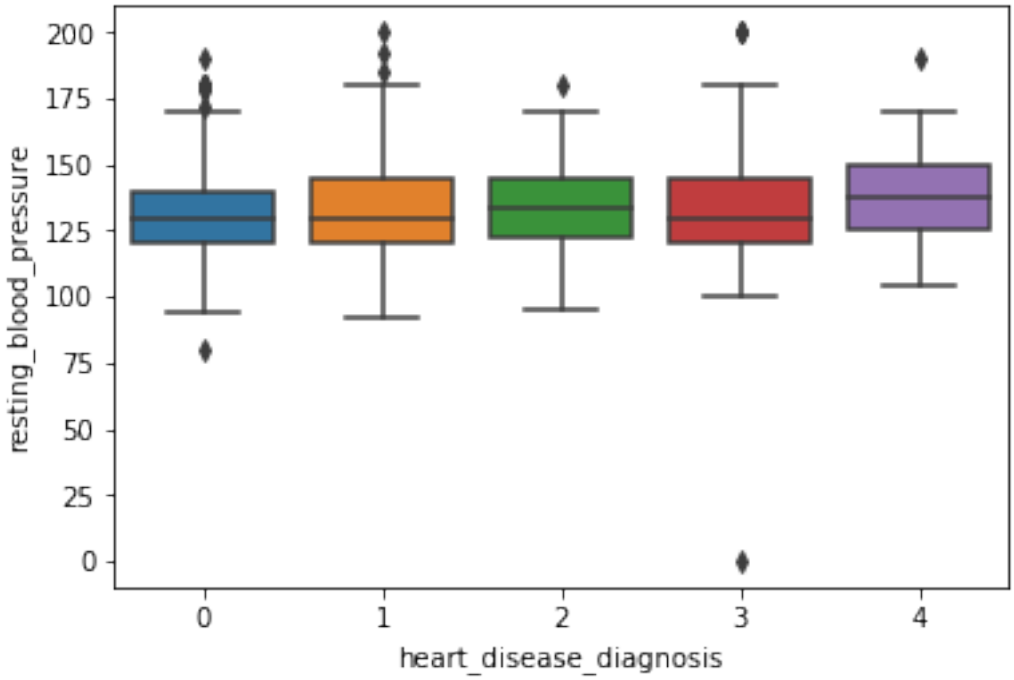
```
In [17]: sns.catplot(x="chest_pain_type", y="heart_disease_diagnosis",
                     data=train_df.loc[train_df['chest_pain_type'].astype(str) != '?'], kind='
         plt.show()
```



The figure above shows the cat plot of the chest pain type for each heart disease diagnosis. For the worst heart disease diagnosis, chest pain type 4 is dominant.

0.3.4 Resting Blood Pressure

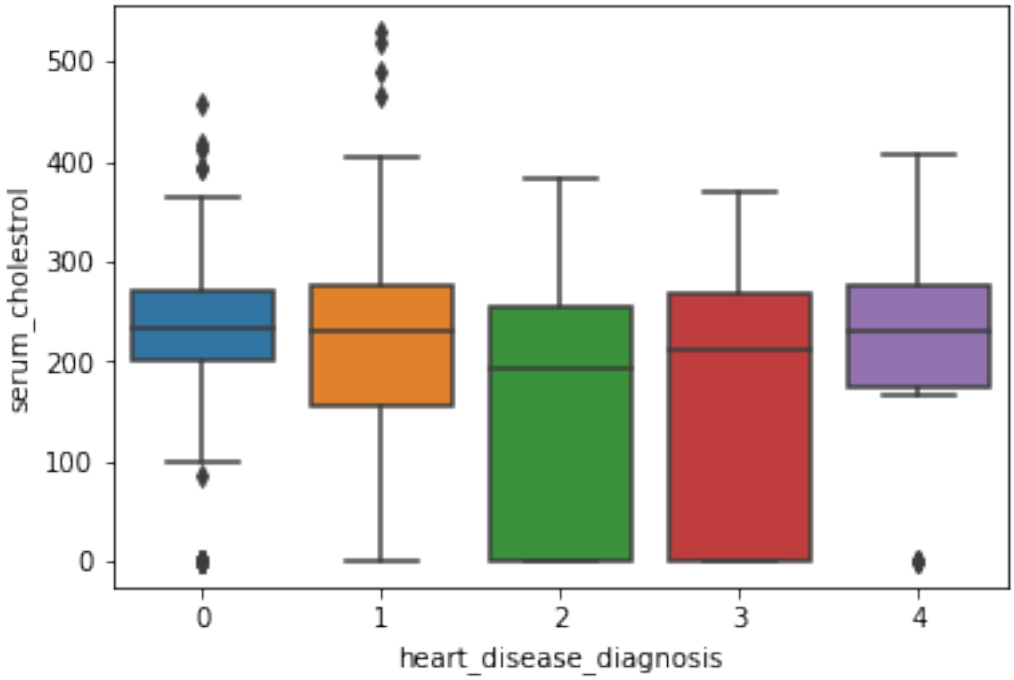
```
In [18]: clean_data = train_df.loc[train_df['resting_blood_pressure'].astype(str) != '?' ]
         clean_data = clean_data[['resting_blood_pressure', 'heart_disease_diagnosis']].astype
         sns.boxplot(x="heart_disease_diagnosis", y="resting_blood_pressure",
                     data=clean_data)
         plt.show()
```



The figure above shows the box plot of the resting blood pressure for each diagnosed person. There is no apparent difference between the five heart disease category. This shows a low variance of data.

0.3.5 Serum Cholestrol

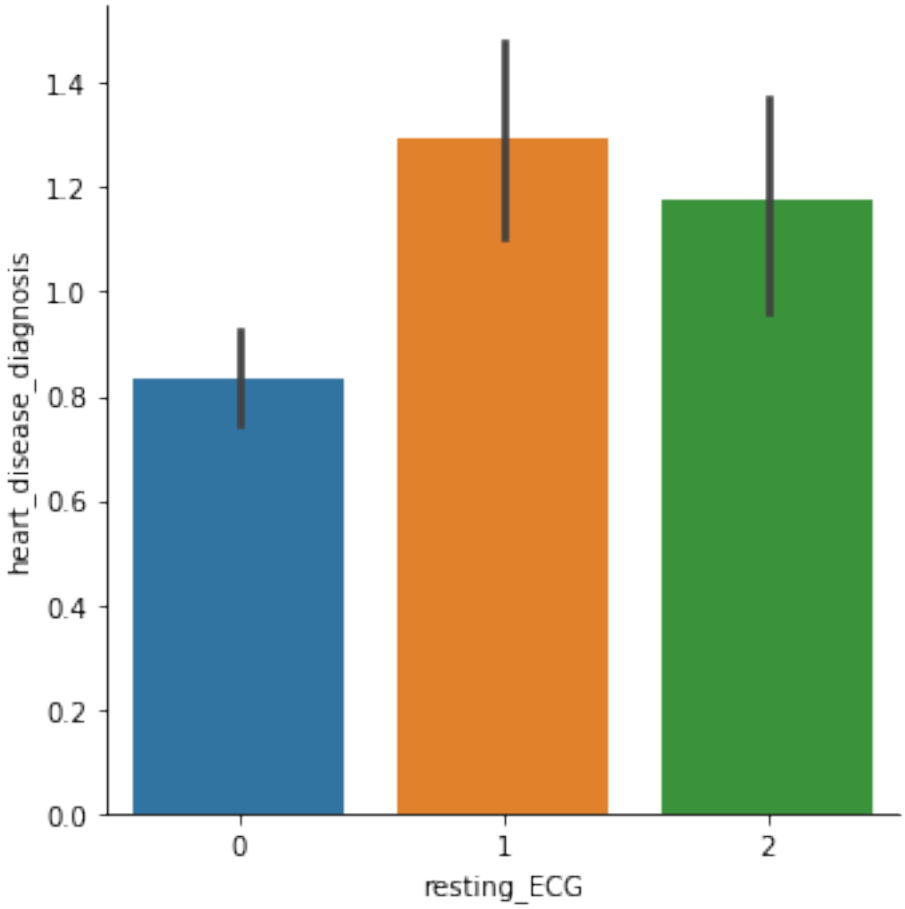
```
In [19]: clean_data = train_df.loc[train_df['serum_cholestrol'].astype(str) != '?' ]
         clean_data = clean_data[['serum_cholestrol', 'heart_disease_diagnosis']].astype(int)
         sns.boxplot(x="heart_disease_diagnosis", y="serum_cholestrol",
                     data=clean_data)
         plt.show()
```



The figure above shows the box plot of the serum cholestrol for each diagnosed person. There are a lot of zero values in the data, making the heart disease diagnosis 2 and 3 have a really wide boxplot.

0.3.6 Resting ECG

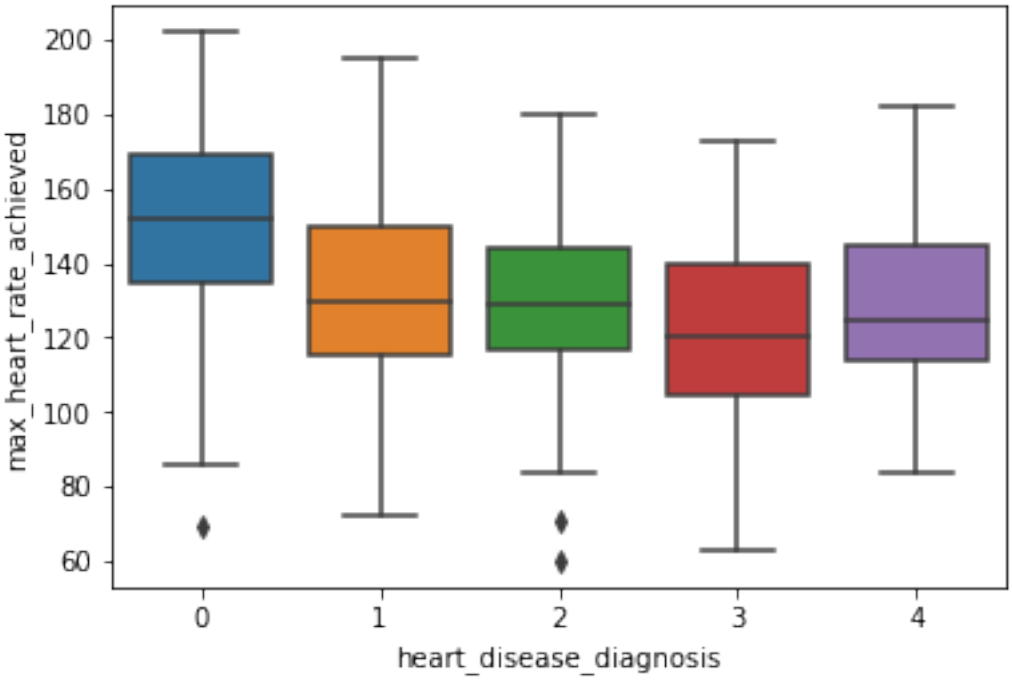
```
In [20]: clean_data = train_df.loc[train_df['resting_ECG'].astype(str) != '?']
         clean_data = clean_data.loc[~clean_data['resting_ECG'].isna()]
         clean_data = clean_data[['resting_ECG', 'heart_disease_diagnosis']].astype(int)
         sns.catplot(x="resting_ECG", y="heart_disease_diagnosis",
                     data=clean_data, kind='bar')
         plt.show()
```



The figure above shows the catplot of the resting ECG for each diagnosed person. There is no apparent difference between the 3 type of resting ECG, looking at the frequency of each heart disease category

0.3.7 Max Heart Rate Achieved

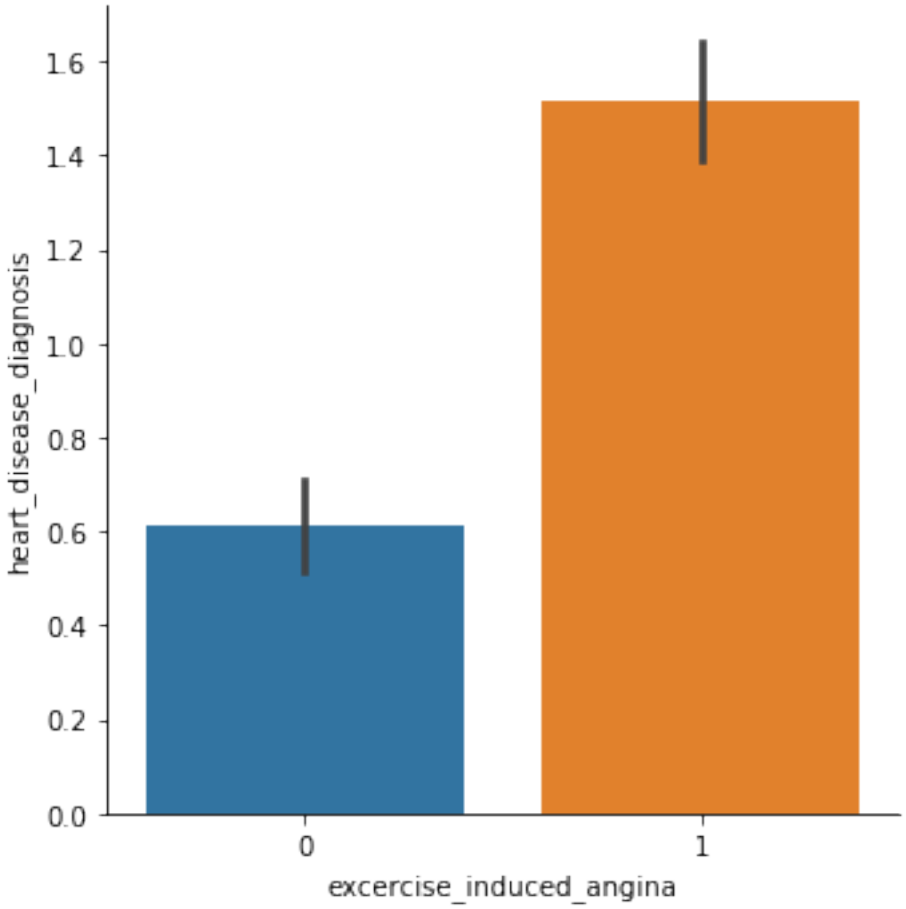
```
In [21]: clean_data = train_df.loc[train_df['max_heart_rate_achieved'].astype(str) != '?' ]
         clean_data = clean_data[['max_heart_rate_achieved', 'heart_disease_diagnosis']].astyp
         sns.boxplot(x="heart_disease_diagnosis", y="max_heart_rate_achieved",
                     data=clean_data)
         plt.show()
```



The figure above shows the box plot of the max heart rate achieved for each person. The max heart rate is higher when the person is not diagnosed with a heart disease.

0.3.8 Exercise Induced Angina

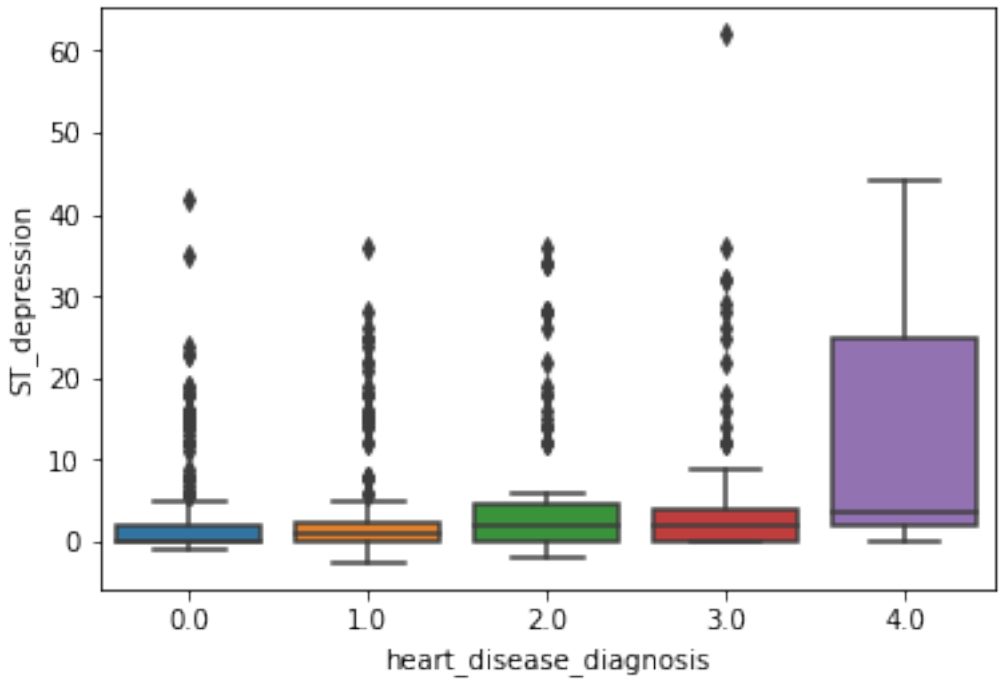
```
In [22]: clean_data = train_df.loc[train_df['excercise_induced_angina'].astype(str) != '?']
         clean_data = clean_data.loc[~clean_data['excercise_induced_angina'].isna()]
         clean_data = clean_data[['excercise_induced_angina', 'heart_disease_diagnosis']].asty
         sns.catplot(x="excercise_induced_angina", y="heart_disease_diagnosis",
                     data=clean_data, kind='bar')
         plt.show()
```



The figure above shows the catplot of the exercise induced angina for each diagnosed person. There is no apparent difference between the 2 type of exercise induced angina, looking at the frequency of each heart disease category

0.3.9 ST Depression

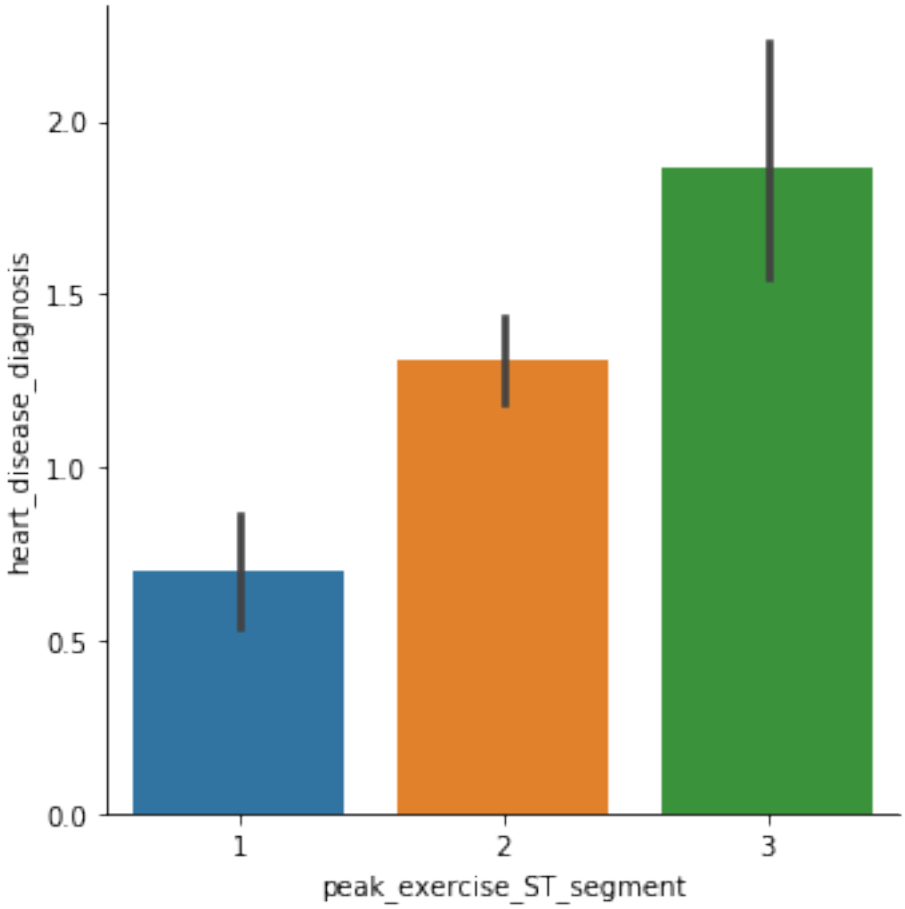
```
In [23]: clean_data = train_df.loc[train_df['ST_depression'].astype(str) != '?' ]
         clean_data = clean_data[['ST_depression', 'heart_disease_diagnosis']].astype(float)
         sns.boxplot(x="heart_disease_diagnosis", y="ST_depression",
                     data=clean_data)
         plt.show()
```



The figure above shows the box plot of the ST Depression for each person. There is a very low variance for the ST depression for each of the heart diagnosis.

0.3.10 Peak Exercise ST Segment

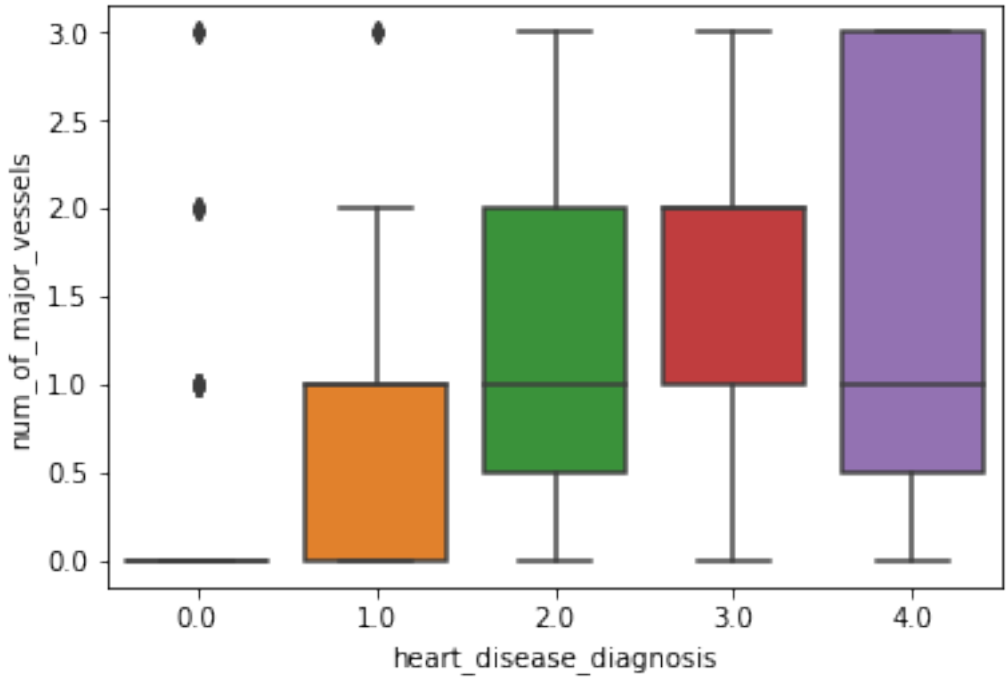
```
In [24]: clean_data = train_df.loc[train_df['peak_exercise_ST_segment'].astype(str) != '?']
         clean_data = clean_data.loc[~clean_data['peak_exercise_ST_segment'].isna()]
         clean_data = clean_data[['peak_exercise_ST_segment', 'heart_disease_diagnosis']].asty
         sns.catplot(x="peak_exercise_ST_segment", y="heart_disease_diagnosis",
                     data=clean_data, kind='bar')
         plt.show()
```



The figure above shows the catplot of the peak exercise ST segment for each diagnosed person. There is no apparent difference between the 2 type of exercise induced angina, looking at the frequency of each heart disease category

0.3.11 Number of Major Vessels

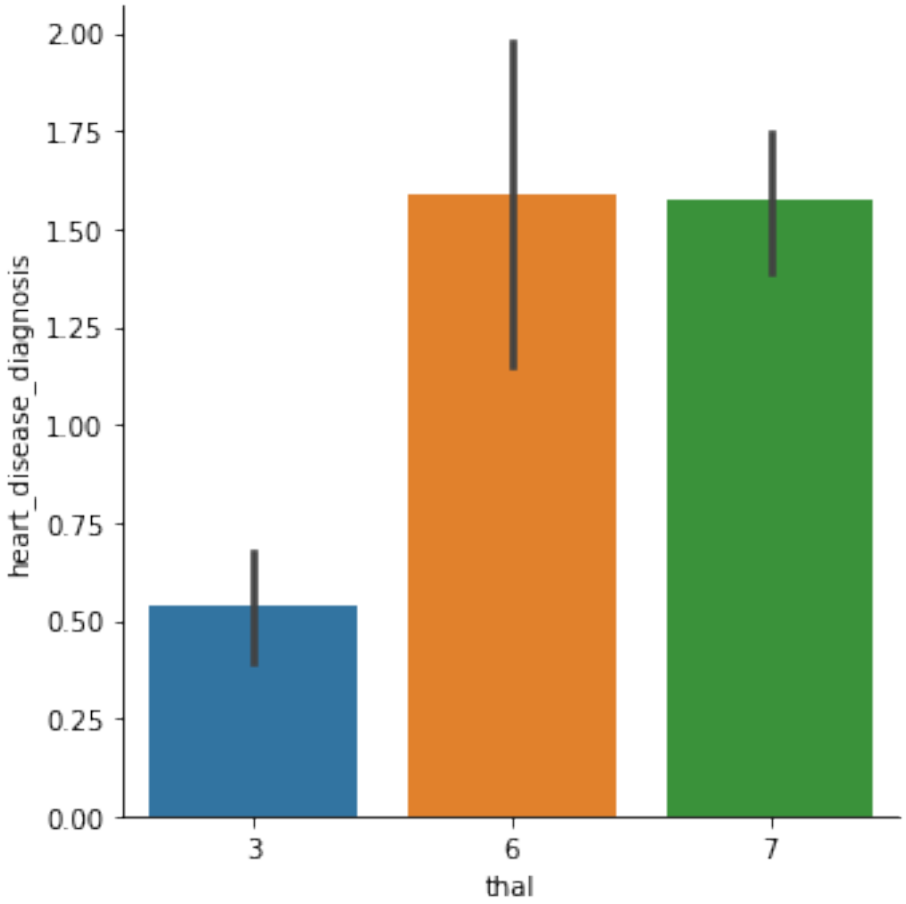
```
In [25]: clean_data = train_df.loc[train_df['num_of_major_vessels'].astype(str) != '?' ]
         clean_data = clean_data[['num_of_major_vessels', 'heart_disease_diagnosis']].astype(f
         sns.boxplot(x="heart_disease_diagnosis", y="num_of_major_vessels",
                     data=clean_data)
         plt.show()
```



The figure above shows the box plot of the number of major vessels for each person. No major vessel is found in the person with negative heart disease diagnosis.

0.3.12 Thal

```
In [26]: clean_data = train_df.loc[train_df['thal'].astype(str) != '?']
         clean_data = clean_data.loc[~clean_data['thal'].isna()]
         clean_data = clean_data[['thal', 'heart_disease_diagnosis']].astype(int)
         sns.catplot(x="thal", y="heart_disease_diagnosis",
                     data=clean_data, kind='bar')
         plt.show()
```



The figure above shows the catplot of the thal for each diagnosed person. There is no apparent difference between the 3 type of thal, looking at the frequency of each heart disease category

2. Data Preprocessing and Feature Engineering

November 21, 2018

In [18]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

pd.options.display.max_columns = 1000

%matplotlib inline
```

0.1 Load Data

0.1.1 Initialize Path Constants

In [19]:
```
RAW_DATA_PATH = '../data/raw'
PROCESSED_DATA_PATH = '../data/processed'
```

0.1.2 Load CSV Files

In [20]:
```
train_df = pd.read_csv('{}/tubes2_HeartDisease_train.csv'.format(RAW_DATA_PATH))
test_df = pd.read_csv('{}/tubes2_HeartDisease_test.csv'.format(RAW_DATA_PATH))
```

0.2 Rename Column Names & Convert '?' to NaN

The '?' symbol is converted into py.NaN in order to let the program classify the columns with missing values ('?') as a number, not an object.

In [21]:
```
from copy import deepcopy

test_columns_replacement = {
    'Column1': 'age',
    'Column2': 'sex',
    'Column3': 'chest_pain_type',
    'Column4': 'resting_blood_pressure',
    'Column5': 'serum_cholestrol',
    'Column6': 'fasting_blood_sugar',
    'Column7': 'resting_ECG',
    'Column8': 'max_heart_rate_achieved',
    'Column9': 'excercise_induced_angina',
```

```
    'Column10': 'ST_depression',
    'Column11': 'peak_exercise_ST_segment',
    'Column12': 'num_of_major_vessels',
    'Column13': 'thal',
}

train_columns_replacement = test_columns_replacement.copy()
train_columns_replacement['Column14'] = 'heart_disease_diagnosis'

train_df = train_df.rename(columns=train_columns_replacement).replace('?', np.NaN)
test_df = test_df.rename(columns=test_columns_replacement).replace('?', np.NaN)

combine = [train_df,test_df]
```

In [22]:
```
train_df.head()
```

Out[22]:

	age	sex	chest_pain_type	resting_blood_pressure	serum_cholestrol	fasting_blood_sugar	resting_ECG	max_heart_rate_achieved	excercise_induced_angina	ST_depression	peak_exercise_ST_segment	num_of_major_vessels	thal	heart_disease_diagnosis
0	54	1	4	125	216	0	0	140	0	0	NaN	NaN	NaN	1
1	55	1	4	158	217	0	0	110	1	2.5	2	NaN	NaN	1
2	54	0	3	135	304	1	0	170	0	0	1	0	3	0
3	48	0	3	120	195	0	0	125	0	0	NaN	NaN	NaN	0
4	50	1	4	120	0	0	1	156	1	0	1	NaN	6	3

In [23]:
```
test_df.head()
```

Out[23]:

	age	sex	chest_pain_type	resting_blood_pressure	serum_cholestrol
0	60	1	2	160	267
1	61	1	4	148	203

```
2   54    1                4                   130                242
3   48    1                4                   120                260
4   57    0                1                   130                308

  fasting_blood_sugar  resting_ECG max_heart_rate_achieved  \
0                   1            1                      157
1                   0            0                      161
2                   0            0                       91
3                   0            0                      115
4                   0            0                       98

  excercise_induced_angina ST_depression peak_exercise_ST_segment  \
0                        0           0.5                        2
1                        0             0                        1
2                        1             1                        2
3                        0             2                        2
4                        0             1                        2

  num_of_major_vessels thal
0                  NaN  NaN
1                    1    7
2                  NaN  NaN
3                  NaN  NaN
4                  NaN  NaN
```

0.3 Data Preprocessing

0.4 Impute Null Data

0.4.1 to make it simple, we encode null values with the median (numerical features)

```
In [24]: def stringify_categorical_data(df, colnames):
             for colname in colnames:
                 df[colname] = df[colname].astype(str)

         def impute_numerical_data(df,colnames):
             for colname in colnames:
                 df[colname].fillna(df.loc[df[colname]!=np.NaN][colname].median(),inplace=True)
                 df[colname] = df[colname].astype(np.float64)
```

0.4.2 Categorical data :

Categorical data is a categorical measurement expressed not in terms of numbers, but rather by means of a natural language description. In statistics, it is often used interchangeably with "categorical" data. Categorical data represent characteristics such as a person's gender, marital status, hometown, or the types of movies they like. Categorical data can take on numerical values (such as "1" indicating male and "2" indicating female), but those numbers don't have mathematical meaning. You couldn't add them together, for example.

0.4.3 Fields which is considered as Categorical in our problem :

- peak_exercise_ST_segment
- exercise_induced_angina
- thal

0.4.4 Numerical data :

Numerical data is a numerical measurement expressed not by means of a natural language description, but rather in terms of numbers.These data have meaning as a measurement, such as a person's height, weight, IQ, or blood pressure; or they're a count, such as the number of stock shares a person owns, how many teeth a dog has, or how many pages you can read of your favorite book before you fall asleep.

0.4.5 Fields which is considered as Numerical in our problem :

- max_heart_rate_achieved
- ST_depression
- num_of_major_vessels
- resting_blood_pressure
- fasting_blood_sugar
- serum_cholestrol
- resting_ECG

```
In [25]: categorical_data_colname = ['peak_exercise_ST_segment','excercise_induced_angina','tha
         numerical_data_colname = ['max_heart_rate_achieved',
                                   'ST_depression',
                                   'num_of_major_vessels',
                                   'resting_blood_pressure',
                                   'fasting_blood_sugar',
                                   'serum_cholestrol',
                                   'resting_ECG']

         for df in [train_df, test_df]:
             impute_numerical_data(df, numerical_data_colname)
             stringify_categorical_data(df, categorical_data_colname)

         train_df.head()
```

Out[25]:

	age	sex	chest_pain_type	resting_blood_pressure	serum_cholestrol	\
0	54	1	4	125.0	216.0	
1	55	1	4	158.0	217.0	
2	54	0	3	135.0	304.0	
3	48	0	3	120.0	195.0	
4	50	1	4	120.0	0.0	

	fasting_blood_sugar	resting_ECG	max_heart_rate_achieved	\
0	0.0	0.0	140.0	
1	0.0	0.0	110.0	

```
2                  1.0          0.0                   170.0
3                  0.0          0.0                   125.0
4                  0.0          1.0                   156.0

   excercise_induced_angina  ST_depression peak_exercise_ST_segment  \
0                         0            0.0                      nan
1                         1            2.5                        2
2                         0            0.0                        1
3                         0            0.0                      nan
4                         1            0.0                        1

    num_of_major_vessels thal  heart_disease_diagnosis
0                    0.0  nan                        1
1                    0.0  nan                        1
2                    0.0    3                        0
3                    0.0  nan                        0
4                    0.0    6                        3
```

In [26]:
```
test_df.head()
```

Out[26]:
```
   age  sex  chest_pain_type  resting_blood_pressure  serum_cholestrol  \
0   60    1                2                   160.0             267.0
1   61    1                4                   148.0             203.0
2   54    1                4                   130.0             242.0
3   48    1                4                   120.0             260.0
4   57    0                1                   130.0             308.0

    fasting_blood_sugar  resting_ECG  max_heart_rate_achieved  \
0                   1.0          1.0                   157.0
1                   0.0          0.0                   161.0
2                   0.0          0.0                    91.0
3                   0.0          0.0                   115.0
4                   0.0          0.0                    98.0

   excercise_induced_angina  ST_depression peak_exercise_ST_segment  \
0                         0            0.5                        2
1                         0            0.0                        1
2                         1            1.0                        2
3                         0            2.0                        2
4                         0            1.0                        2

    num_of_major_vessels thal
0                    0.0  nan
1                    1.0    7
2                    0.0  nan
3                    0.0  nan
4                    0.0  nan
```

In [27]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 779 entries, 0 to 778
Data columns (total 14 columns):
age                          779 non-null int64
sex                          779 non-null int64
chest_pain_type              779 non-null int64
resting_blood_pressure       779 non-null float64
serum_cholestrol             779 non-null float64
fasting_blood_sugar          779 non-null float64
resting_ECG                  779 non-null float64
max_heart_rate_achieved      779 non-null float64
excercise_induced_angina     779 non-null object
ST_depression                779 non-null float64
peak_exercise_ST_segment     779 non-null object
num_of_major_vessels         779 non-null float64
thal                         779 non-null object
heart_disease_diagnosis      779 non-null int64
dtypes: float64(7), int64(4), object(3)
memory usage: 85.3+ KB
```

0.5 One-Hot Encoding

```
In [28]: train_df = pd.get_dummies(train_df)
         test_df = pd.get_dummies(test_df)

In [29]: train_df.head()
```

Out[29]:

	age	sex	chest_pain_type	resting_blood_pressure	serum_cholestrol	\
0	54	1	4	125.0	216.0	
1	55	1	4	158.0	217.0	
2	54	0	3	135.0	304.0	
3	48	0	3	120.0	195.0	
4	50	1	4	120.0	0.0	

	fasting_blood_sugar	resting_ECG	max_heart_rate_achieved	ST_depression	\
0	0.0	0.0	140.0	0.0	
1	0.0	0.0	110.0	2.5	
2	1.0	0.0	170.0	0.0	
3	0.0	0.0	125.0	0.0	
4	0.0	1.0	156.0	0.0	

	num_of_major_vessels	heart_disease_diagnosis	excercise_induced_angina_0	\
0	0.0	1	1	
1	0.0	1	0	
2	0.0	0	1	
3	0.0	0	1	
4	0.0	3	0	

```
   excercise_induced_angina_1  excercise_induced_angina_nan  \
0                           0                             0
1                           1                             0
2                           0                             0
3                           0                             0
4                           1                             0

   peak_exercise_ST_segment_1  peak_exercise_ST_segment_2  \
0                           0                          0
1                           0                          1
2                           1                          0
3                           0                          0
4                           1                          0

   peak_exercise_ST_segment_3  peak_exercise_ST_segment_nan  thal_3  thal_6  \
0                           0                            1       0       0
1                           0                            0       0       0
2                           0                            0       1       0
3                           0                            1       0       0
4                           0                            0       0       1

   thal_7  thal_nan
0       0         1
1       0         1
2       0         0
3       0         1
4       0         0
```

In [30]: `train_df.values.shape`

Out[30]: (779, 22)

In [31]: `test_df.head()`

Out[31]:
```
   age  sex  chest_pain_type  resting_blood_pressure  serum_cholestrol  \
0   60    1                2                   160.0             267.0
1   61    1                4                   148.0             203.0
2   54    1                4                   130.0             242.0
3   48    1                4                   120.0             260.0
4   57    0                1                   130.0             308.0

   fasting_blood_sugar  resting_ECG  max_heart_rate_achieved  ST_depression  \
0                  1.0          1.0                    157.0            0.5
1                  0.0          0.0                    161.0            0.0
2                  0.0          0.0                     91.0            1.0
3                  0.0          0.0                    115.0            2.0
4                  0.0          0.0                     98.0            1.0
```

```
   num_of_major_vessels  excercise_induced_angina_0  \
0                   0.0                           1
1                   1.0                           1
2                   0.0                           0
3                   0.0                           1
4                   0.0                           1

   excercise_induced_angina_1  excercise_induced_angina_nan  \
0                           0                             0
1                           0                             0
2                           1                             0
3                           0                             0
4                           0                             0

   peak_exercise_ST_segment_1  peak_exercise_ST_segment_2  \
0                           0                           1
1                           1                           0
2                           0                           1
3                           0                           1
4                           0                           1

   peak_exercise_ST_segment_3  peak_exercise_ST_segment_nan  thal_3  thal_6  \
0                           0                             0       0       0
1                           0                             0       0       0
2                           0                             0       0       0
3                           0                             0       0       0
4                           0                             0       0       0

   thal_7  thal_nan
0       0         1
1       1         0
2       0         1
3       0         1
4       0         1
```

```
In [32]: test_df.shape
```

```
Out[32]: (141, 21)
```

```
In [39]: test_df = test_df[train_df.drop('heart_disease_diagnosis', axis=1).columns]
```

0.6 Save Processed Data

Save the processed data to the ../raw/processed folder

```
In [53]: train_df.to_csv('{}/processed_train_data.csv'.format(PROCESSED_DATA_PATH),index=False)
         test_df.to_csv('{}/processed_test_data.csv'.format(PROCESSED_DATA_PATH),index=False)
```

0.7 Reference

1. http://scaryscientist.blogspot.com/2015/02/classification-of-data-types.html

In [ ]:

3. Modelling and Tuning

November 21, 2018

In [18]:
```
import pandas as pd
import numpy as np

from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier

from sklearn.model_selection import train_test_split, KFold, cross_val_score, GridSea

from sklearn.metrics import accuracy_score, recall_score, precision_score

import pickle

import warnings

warnings.filterwarnings('ignore')
```

1 Load data

1.0.1 Initialize path constant

In [19]:
```
DATA_PATH = '../data'
RAW_DATA_PATH = '{}/raw'.format(DATA_PATH)
PROCESSED_DATA_PATH = '{}/processed'.format(DATA_PATH)

MODEL_PATH = '../models'
```

1.0.2 Read processed data from CSV

In [32]:
```
# data = pd.read_csv('{}/processed_data.csv'.format(PROCESSED_DATA_PATH))
data = pd.read_csv('{}/processed_train_data.csv'.format(PROCESSED_DATA_PATH))
X_df = data.drop(['heart_disease_diagnosis'], axis=1)
X = X_df
y = data['heart_disease_diagnosis']
```

2 Feature Elimination Function

In [4]:
```
from itertools import chain, combinations

        # RFE-ish with no rank. kwargs = arguments to be passed to the scorer
        # Maybe create a custom scoring function for combining multiple score?
        # X HAS TO BE DATAFRAME, because it supports .drop()
        def rfe_no_rank(model, X_train, X_vali, y_train, y_vali, scorer, **kwargs):
            attr_to_drop = []
            best_target_yet = 0
            attributes = X_train.columns.values.tolist()
            # For every element in the attribute powerset...
            for subset in chain.from_iterable(combinations(attributes, r) for r in range(len(at
                # ...fit, predict, and calculate score without them
                model.fit(X_train.drop(list(subset), axis=1), y_train)
                y_pred = model.predict(X_vali.drop(list(subset), axis=1))
                target_now = scorer(y_pred, y_vali, **kwargs)
                if target_now > best_target_yet:
                    best_target_yet = target_now
                    attr_to_drop = list(subset)
            return attr_to_drop, best_target_yet
```

3 Evaluation

We decided to use accuracy as out main evaluation metric to optimize so that we can see the performance of each model in predicting the correct label.

4 Cross Validation

4.0.1 CV Function

In [13]:
```
def get_kfold():
            return KFold(n_splits=5, shuffle=True, random_state=1)

        def print_cv_result(model, X, y):
            accuracy_scores = []
            precision_scores = []
            recall_scores = []

            kfold = get_kfold()

            for train_idx, validation_idx in kfold.split(X, y):
                X_train = X[train_idx]
                y_train = y[train_idx]
                X_validation = X[validation_idx]
                y_validation = y[validation_idx]
```

```
            model.fit(X_train, y_train)

            prediction = model.predict(X_validation)

            accuracy = accuracy_score(y_validation, prediction)
            precision = precision_score(y_validation, prediction, average='macro')
            recall = recall_score(y_validation, prediction, average='macro')

            accuracy_scores.append(accuracy)
            precision_scores.append(precision)
            recall_scores.append(recall)

        print('--- Validation Metrics ---')
        print('Accuracy  = {:.3f}'.format(np.mean(accuracy_scores)))
        print('Precision = {:.3f}'.format(np.mean(precision_scores)))
        print('Recall    = {:.3f}'.format(np.mean(recall_scores)))
```

4.1 Models

4.1.1 Naive Bayes

```
In [7]: model_name = 'Naive Bayes'
        nb_model = GaussianNB()

        print('=== {} ===\n'.format(model_name))
        print_cv_result(nb_model, X, y)
```

```
=== Naive Bayes ===

--- Validation Metrics ---
Accuracy  = 0.546
Precision = 0.353
Recall    = 0.370
```

4.1.2 KNN

In K-Nearest Neighbors algorithm, it's really important to scale the features first (feature scaling). Since the range of values of raw data varies widely, in K-Nearest Neighbors algoritm, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

```
In [8]: model_name = 'K-Nearest Neighbor'
        knn_model = KNeighborsClassifier()
        X_Scaled = preprocessing.scale(X)
```

```
        print('=== {} ===\n'.format(model_name))
        print_cv_result(knn_model, X_Scaled, y)
```

```
=== K-Nearest Neighbor ===

--- Validation Metrics ---
Accuracy  = 0.548
Precision = 0.298
Recall    = 0.314
```

4.1.3 Decision Tree

```
In [10]: model_name = 'Decision Tree'
         dtc_model = DecisionTreeClassifier(criterion='entropy', random_state=1)

         print('=== {} ===\n'.format(model_name))
         print_cv_result(dtc_model, X, y)

         # rfe_no_rank(dtc_model, X_df[:500], X_df[500:], y[:500], y[500:], accuracy_score)
```

```
=== Decision Tree ===

--- Validation Metrics ---
Accuracy  = 0.458
Precision = 0.313
Recall    = 0.313
```

4.1.4 ANN

```
In [11]: model_name = 'ANN'
         ann_model = MLPClassifier(random_state=1, activation='logistic')
         X_Scaled = preprocessing.scale(X)

         print('=== {} ===\n'.format(model_name))
         print_cv_result(ann_model, X_Scaled, y)
```

```
=== ANN ===

--- Validation Metrics ---
Accuracy  = 0.582
Precision = 0.347
Recall    = 0.351
```

5 Feature Elimination

```
In [38]: from sklearn.base import clone

         def rfe_no_rank(model, X, y, n_remove):
             if n_remove <= 0:
                 return

             best_score = -1
             best_removed = None
             for feat in X.columns:
                 score = cross_val_score(clone(model),
                                         preprocessing.scale(X.drop(feat, axis=1).values),
                                         y,
                                         cv=get_kfold(),
                                         scoring='accuracy').mean()
                 if score > best_score:
                     best_score = score
                     best_removed = feat

             print('remove', best_removed, '- Best accuracy:', best_score)

             rfe_no_rank(model, X.drop(best_removed, axis=1), y, n_remove - 1)

In [39]: rfe_no_rank(MLPClassifier(random_state=1, activation='logistic'), X, y, 5)
```

```
remove sex - Best accuracy: 0.5943507030603804
remove fasting_blood_sugar - Best accuracy: 0.582803970223325
remove max_heart_rate_achieved - Best accuracy: 0.5853598014888337
remove peak_exercise_ST_segment_1 - Best accuracy: 0.5892059553349875
remove resting_blood_pressure - Best accuracy: 0.5879239040529363
```

6 Tune Best Base Model

```
In [41]: param_grid = {
             'hidden_layer_sizes': [(50,), (100,), (50, 50), (100, 100)],
             'solver': ['lbfgs', 'sgd', 'adam'],
             'alpha': [1e-4, 1e-3, 1e-2, 1e-1],
             'learning_rate': ['constant', 'invscaling', 'adaptive']
         }

         best_model = GridSearchCV(MLPClassifier(random_state=1, activation='logistic'),
                                   param_grid,
                                   cv=get_kfold(),
                                   scoring='accuracy',
                                   verbose=20)
```

```
        best_model.fit(preprocessing.scale(X.drop('sex', axis=1).values), y)
```

```
Fitting 5 folds for each of 144 candidates, totalling 720 fits
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs


[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.467
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.525
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs


[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.3s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.5s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.467
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs


[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.7s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.474
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs


[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.9s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.477
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd


[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    1.1s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.55769
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd


[Parallel(n_jobs=1)]: Done   6 out of   6 | elapsed:    1.6s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.57051
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
```

```
[Parallel(n_jobs=1)]: Done   7 out of   7 | elapsed:    2.2s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.55128
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd


[Parallel(n_jobs=1)]: Done   8 out of   8 | elapsed:    2.7s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.63461
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd


[Parallel(n_jobs=1)]: Done   9 out of   9 | elapsed:    3.3s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.53548
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam


[Parallel(n_jobs=1)]: Done  10 out of  10 | elapsed:    3.7s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5897
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam


[Parallel(n_jobs=1)]: Done  11 out of  11 | elapsed:    4.3s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5897
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam


[Parallel(n_jobs=1)]: Done  12 out of  12 | elapsed:    4.9s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5641
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam


[Parallel(n_jobs=1)]: Done  13 out of  13 | elapsed:    5.4s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5705
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam


[Parallel(n_jobs=1)]: Done  14 out of  14 | elapsed:    6.0s remaining:    0.0s
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.548
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs


[Parallel(n_jobs=1)]: Done  15 out of  15 | elapsed:    6.4s remaining:    0.0s
[Parallel(n_jobs=1)]: Done  16 out of  16 | elapsed:    6.6s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs


[Parallel(n_jobs=1)]: Done  17 out of  17 | elapsed:    6.8s remaining:    0.0s
[Parallel(n_jobs=1)]: Done  18 out of  18 | elapsed:    7.0s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs


[Parallel(n_jobs=1)]: Done  19 out of  19 | elapsed:    7.2s remaining:    0.0s


[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.435
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.435
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.423
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.487
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.419
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.58
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.58
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.56
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.57
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.54
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.467
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.525
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.467
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.474
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.477
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.55769
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.57051
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.55128
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.63461
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.53548
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.589
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.589
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.564
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.570
[CV] alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.548
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.56
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.53
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.46
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.55
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.564
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.551
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.538
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.615
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.5225
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.602
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.583
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.593
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.42
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.6
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.50
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.53
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.46
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.55
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.564
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.551
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.538
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.615
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.522
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.59
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.60
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.59
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.58
[CV] alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.59
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.42
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.43
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.6
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.6
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, sco
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, sco
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, sco
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, sco
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, sco
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
```

```
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.0001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5128
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5128
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.4935
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5192
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.4838
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.557692
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.570512
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.551282
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.634615
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.535483
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5897
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5897
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5641
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5705
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.5483
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.48
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.4358
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.4358
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.4230
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.4871
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.4193
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.589
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.589
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.564
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.570
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.548
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5128
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5128
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.4935
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5192
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.4838
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.557692
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.570512
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.551282
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.634615
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.535483
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.5897
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.5897
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.5641
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.5705
[CV] alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.5483
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.480
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.525
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.557
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.493
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.516
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.5641
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.5512
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.5384
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.6153
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.5225
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.602
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.583
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.593
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.429
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.5,
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.438
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.59
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.60
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.59
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.58
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.59
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.480
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.525
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.557
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.493
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.516
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.5641
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.55128
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.53846
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.61538
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.52258
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.596
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.602
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.596
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.583
[CV] alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.593
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.429
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.5,
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.438
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.60
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.61
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.55
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.57
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.55
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.3
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.429
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.435
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.5,
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.438
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.60
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.61
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.55
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.57
[CV] alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.55
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
```

```
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.5
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.001, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5128
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5384
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.4551
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.5064
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.4516
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.557692
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.570512
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.551282
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.634615
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.535483
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.596153
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.596153
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.564102
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.570512
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.548387
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.512
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.538
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.455
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.506
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.451
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.42307
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.48717
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.41935
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.564
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.570
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.548
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5128
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5384
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.4551
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5064
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.4516
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.557692
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.570512
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.551282
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.634615
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.535483
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.596153
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.596153
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.564102
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.570512
[CV] alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.548387
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.493
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5448
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.519
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.467
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.535
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.564102
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.551282
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.538461
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.615384
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.52258
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.5961
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.6025
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.5961
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.5769
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.5935
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.49
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.54
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.51
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.46
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.53
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.4294
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.5, t
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.4387
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.602
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.576
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.593
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.4935
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.5448
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.5192
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.467
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.535
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.56410
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.55128
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.53846
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.61538
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.52258
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.5961
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.6025
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.5961
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.5769
[CV] alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.5935
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.53
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.53
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.48
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.51
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.50
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4294
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.5, t
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.438
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.615
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.544
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.57
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.56
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.3
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.2
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4294
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4358
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.5, t
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4387
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.596
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.615
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.544
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.576
[CV] alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.561
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.43
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.42
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.43
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.43
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.6
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.6
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.4
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
```

```
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.43
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.6
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.6
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.01, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.519230
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.570512
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.519230
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.538461
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=lbfgs, score=0.483870
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.5641025
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.5705128
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.5512820
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.6346153
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=sgd, score=0.5354838
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.596153
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.570512
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.557692
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=constant, solver=adam, score=0.567741
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.519
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.570
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.519
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.538
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=lbfgs, score=0.483
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.42307
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.48717
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=sgd, score=0.41935
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.5961
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.5897
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.5705
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.5576
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=invscaling, solver=adam, score=0.5677
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5192
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5705
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5192
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.5384
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=lbfgs, score=0.4838
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.5641025
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.5705128
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.5512820
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.6346153
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=sgd, score=0.5354838
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.596153
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.570512
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.557692
[CV] alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50,), learning_rate=adaptive, solver=adam, score=0.567741
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5128
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5833
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5576
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.5320
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=lbfgs, score=0.4709
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.570512
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.551282
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.538461
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.608974
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=sgd, score=0.522580
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.551282
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.576923
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=constant, solver=adam, score=0.554838
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.512
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.583
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.557
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.532
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=lbfgs, score=0.470
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.42948
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43589
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.5, to
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=sgd, score=0.43870
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5897
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5897
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5512
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5769
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=invscaling, solver=adam, score=0.5548
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.51282
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.58333
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.55769
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.53205
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=lbfgs, score=0.47096
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.570512
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.551282
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.538461
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.608974
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=sgd, score=0.522580
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.589743
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.551282
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.576923
[CV] alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100,), learning_rate=adaptive, solver=adam, score=0.554838
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.48
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.54
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.52
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.5,
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=lbfgs, score=0.50
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4358
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4294
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4358
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.5, t
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=sgd, score=0.4387
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.602
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.583
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.544
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.596
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=constant, solver=adam, score=0.554
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.32
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.28
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.26
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.27
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=sgd, score=0.29
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.6
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=invscaling, solver=adam, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.48
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.54
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.52
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.5,
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=lbfgs, score=0.50
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4358
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4294
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4358
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.5, t
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=sgd, score=0.4387
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.602
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.583
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.544
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.596
[CV] alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(50, 50), learning_rate=adaptive, solver=adam, score=0.554
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0.
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=lbfgs, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.435
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.429
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.435
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.5,
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=sgd, score=0.438
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.59
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.58
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.54
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.58
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=constant, solver=adam, score=0.57
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=lbfgs, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=sgd, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
```

```
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=invscaling, solver=adam, score=0
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0.5
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=lbfgs, score=0.4
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.435
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.429
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.435
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.5,
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=sgd, score=0.438
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.59
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.58
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.54
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.58
[CV] alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam
[CV]  alpha=0.1, hidden_layer_sizes=(100, 100), learning_rate=adaptive, solver=adam, score=0.57
```

```
[Parallel(n_jobs=1)]: Done 720 out of 720 | elapsed:  8.1min finished
```

```
Out[41]: GridSearchCV(cv=KFold(n_splits=5, random_state=1, shuffle=True),
                error_score='raise-deprecating',
                estimator=MLPClassifier(activation='logistic', alpha=0.0001, batch_size='auto'
                beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
                hidden_layer_sizes=(100,), learning_rate='constant',
                learning_rate_init=0.001, max_iter=200, momentum=0.9,
                n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
                random_state=1, shuffle=True, solver='adam', tol=0.0001,
```

```
                validation_fraction=0.1, verbose=False, warm_start=False),
                fit_params=None, iid='warn', n_jobs=None,
                param_grid={'hidden_layer_sizes': [(50,), (100,), (50, 50), (100, 100)], 'solve
                pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
                scoring='accuracy', verbose=20)
```

```
In [42]: print('Best accuracy :', best_model.best_score_)
         print('Best params   :', best_model.best_params_)
```

```
Best accuracy : 0.5943517329910141
Best params   : {'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 's
```

```
In [45]: best_params = {'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'consta
         model = MLPClassifier(random_state=1, activation='logistic', **best_params)
         print_cv_result(model, preprocessing.scale(X.drop('sex', axis=1).values), y)
```

```
--- Validation Metrics ---
Accuracy  = 0.594
Precision = 0.367
Recall    = 0.368
```

7 Final Train

```
In [46]: best_params = {'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'consta
         model = MLPClassifier(random_state=1, activation='logistic', **best_params)
         model.fit(X_Scaled, y)
```

```
Out[46]: MLPClassifier(activation='logistic', alpha=0.0001, batch_size='auto',
                beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
                hidden_layer_sizes=(100,), learning_rate='constant',
                learning_rate_init=0.001, max_iter=200, momentum=0.9,
                n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
                random_state=1, shuffle=True, solver='adam', tol=0.0001,
                validation_fraction=0.1, verbose=False, warm_start=False)
```

8 Predict Test Data

```
In [47]: test_data = pd.read_csv('{}/processed_test_data.csv'.format(PROCESSED_DATA_PATH))
         X_test = test_data.values
```

```
In [48]: X_test_scaled = preprocessing.scale(X_test)
```

```
In [49]: model.predict(X_test_scaled)
```

```
Out[49]: array([0, 1, 1, 1, 0, 0, 3, 1, 0, 0, 1, 1, 0, 2, 0, 3, 1, 0, 0, 1, 0, 1,
                0, 1, 1, 0, 1, 1, 3, 1, 0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
```

```
2, 1, 1, 3, 1, 1, 0, 0, 2, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 2, 0, 0,
1, 0, 0, 0, 1, 0, 0, 2, 0, 1, 1, 0, 0, 0, 0, 3, 1, 3, 0, 3, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 2, 0, 0, 0, 1, 0, 0, 1, 1, 3,
0, 0, 0, 0, 0, 2, 0, 3, 0, 1, 3, 1, 0, 1, 1, 0, 1, 2, 0, 0, 0, 1,
3, 2, 1, 2, 0, 0, 0, 0, 0])
```

8.1 Save Best Model

```
In [50]: pickle.dump(model, open('{}/best_model.pkl'.format(MODEL_PATH), 'wb'))
```

8.2 Reference

1. https://en.wikipedia.org/wiki/Feature_scaling


```
In [ ]:
```