Problem 1: Clustering

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

1.1 Read the data and do exploratory data analysis. Describe the data briefly.

In [1]:

```
# importing libraries.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib.inline
```

```
UsageError: Line magic function `%matplotlib.inline` not found.
```

In [9]:

```
# Lets read the Data and check the head of it to ensure the Data is being properly Load
ed.

df = pd.read_csv (r'E:\Great Learning\Projects\Data Mining- Clusters,CART,RF,ANN\Data s
ets\bank_marketing_part1_Data-1.csv')
df.head()
```

Out[9]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mi
0	19.94	16.92	0.8752	6.675	3.763	
1	15.99	14.89	0.9064	5.363	3.582	
2	18.95	16.42	0.8829	6.248	3.755	
3	10.83	12.96	0.8099	5.278	2.641	
4	17.99	15.86	0.8992	5.890	3.694	

Describing the Data :

In [10]:

```
# Lets check few basic information.

df.shape
```

Out[10]:

```
(210, 7)
```

In [11]:

```
df.columns
```

Out[11]:

```
Index(['spending', 'advance_payments', 'probability_of_full_payment',
       'current_balance', 'credit_limit', 'min_payment_amt',
       'max_spent_in_single_shopping'],
      dtype='object')
```

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
spending                        210 non-null float64
advance_payments                210 non-null float64
probability_of_full_payment     210 non-null float64
current_balance                 210 non-null float64
credit_limit                    210 non-null float64
min_payment_amt                 210 non-null float64
max_spent_in_single_shopping    210 non-null float64
dtypes: float64(7)
memory usage: 11.6 KB
```

In [13]:

```
df.duplicated().sum()
```

Out[13]:

```
0
```

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
spending                        0
advance_payments                0
probability_of_full_payment     0
current_balance                 0
credit_limit                    0
min_payment_amt                 0
max_spent_in_single_shopping    0
dtype: int64
```

There are 7 columns: spending, advance_payments, probability_of_full_payment, current_balance, credit_limit, min_payment_amt, max_spent_in_single_shopping.

As we can see from the above Table, there are no Missing Values and also the Data Type is Float 64 for all the 7 variables.

In [18]:

```
df.describe()
```

Out[18]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_lim
count	210.000000	210.000000	210.000000	210.000000	210.00000
mean	14.847524	14.559286	0.870999	5.628533	3.25860
std	2.909699	1.305959	0.023629	0.443063	0.37771
min	10.590000	12.410000	0.808100	4.899000	2.63000
25%	12.270000	13.450000	0.856900	5.262250	2.94400
50%	14.355000	14.320000	0.873450	5.523500	3.23700
75%	17.305000	15.715000	0.887775	5.979750	3.56175
max	21.180000	17.250000	0.918300	6.675000	4.03300

Above table shows us that Variance is somehow close to each other,but the Magnitude's are different.

In [ ]:

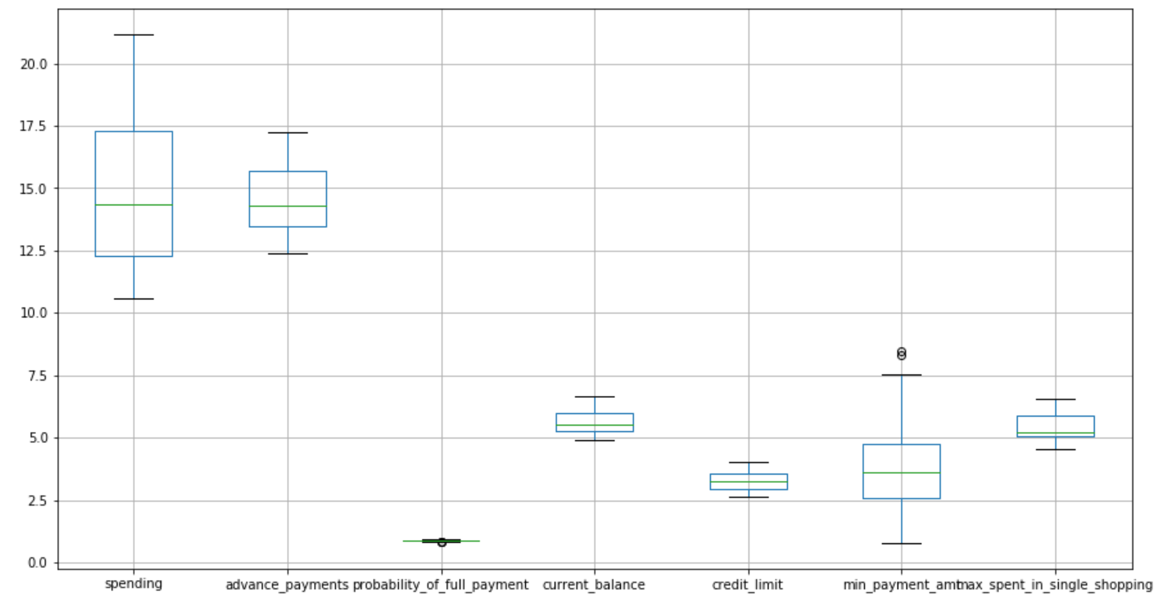
Exploratory Data Analysis (EDA).

In [ ]:

```
# Checing Outliers.
```

In [19]:

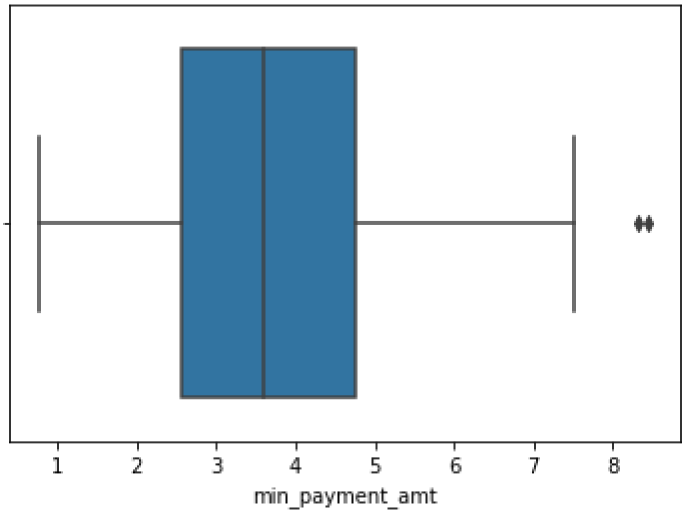
```
df.boxplot (figsize=(15,8));
```



In [ ]:

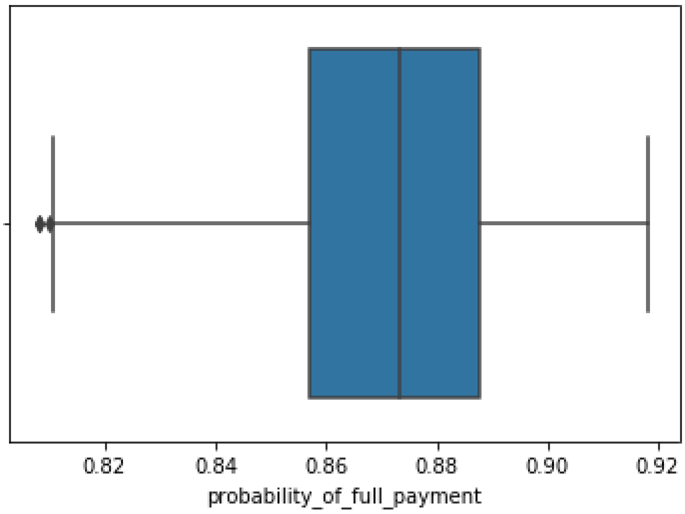
In [20]:

```
sns.boxplot (df ['min_payment_amt']);
```



In [21]:

```
sns.boxplot(df['probability_of_full_payment']);
```



Inferences : Above both the variables have outliers but this outliers cant change or harm the data as they are less and very close to the Minimim and Maximum value.So we will not treating the Outliers as if we will treat the outliers it can bring down all the values on single Digit as range for values are similar or close to each other for both the Variables.

In [39]:

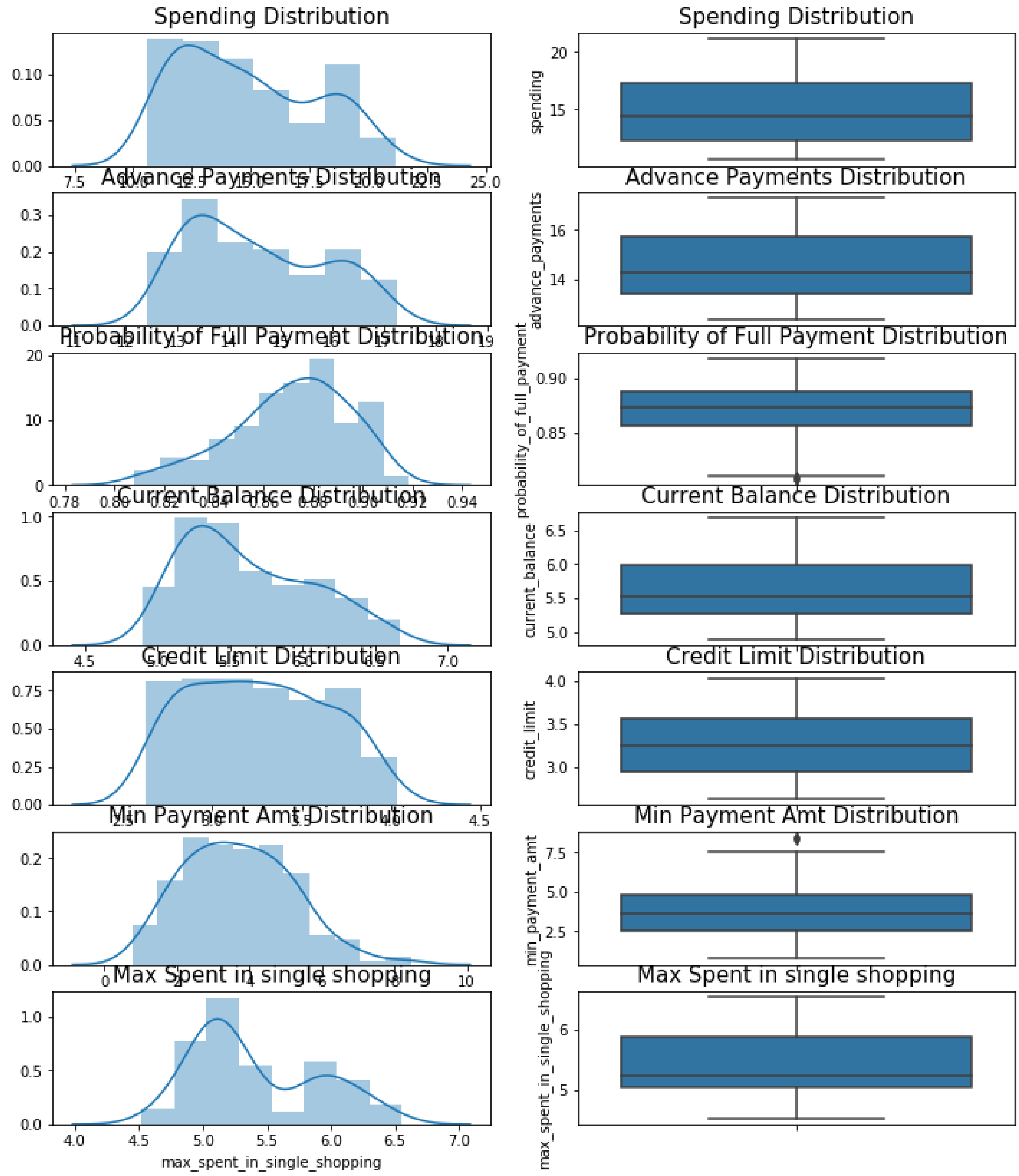
```
# Univariate and Multivariate:
```

In [28]:

```
fig , axes = plt.subplots (nrows=7,ncols=2)
fig.set_size_inches (12,14)

r = sns.distplot (df ['spending'],ax=axes [0][0]);
r.set_title ('Spending Distribution',fontsize=15)
r = sns.boxplot (df ['spending'],orient='v',ax=axes [0][1]);
r.set_title ('Spending Distribution',fontsize=15)

r = sns.distplot (df ['advance_payments'],ax=axes [1][0]);
r.set_title ('Advance Payments Distribution',fontsize=15)
r = sns.boxplot (df ['advance_payments'],orient='v',ax=axes [1][1]);
r.set_title ('Advance Payments Distribution',fontsize=15)

r = sns.distplot (df ['probability_of_full_payment'],ax=axes [2][0]);
r.set_title ('Probability of Full Payment Distribution',fontsize=15)
r = sns.boxplot (df ['probability_of_full_payment'],orient='v',ax=axes [2][1]);
r.set_title ('Probability of Full Payment Distribution',fontsize=15)

r = sns.distplot (df ['current_balance'],ax=axes [3][0]);
r.set_title ('Current Balance Distribution',fontsize=15)
r = sns.boxplot (df ['current_balance'],orient='v',ax=axes [3][1]);
r.set_title ('Current Balance Distribution',fontsize=15)

r = sns.distplot (df ['credit_limit'],ax=axes [4][0]);
r.set_title ('Credit Limit Distribution',fontsize=15)
r = sns.boxplot (df ['credit_limit'],orient='v',ax=axes [4][1]);
r.set_title ('Credit Limit Distribution',fontsize=15)

r= sns.distplot (df ['min_payment_amt'],ax=axes [5][0]);
r.set_title ('Min Payment Amt Distribution',fontsize=15)
r = sns.boxplot (df ['min_payment_amt'],orient='v',ax=axes [5][1]);
r.set_title ('Min Payment Amt Distribution',fontsize=15)

r = sns.distplot (df ['max_spent_in_single_shopping'],ax=axes[6][0]);
r.set_title ('Max Spent in single shopping',fontsize=15)
r= sns.boxplot (df ['max_spent_in_single_shopping'],orient='v', ax=axes [6][1]);
r.set_title ('Max Spent in single shopping',fontsize=15)
```

Out[28]:

```
Text(0.5, 1.0, 'Max Spent in single shopping')
```

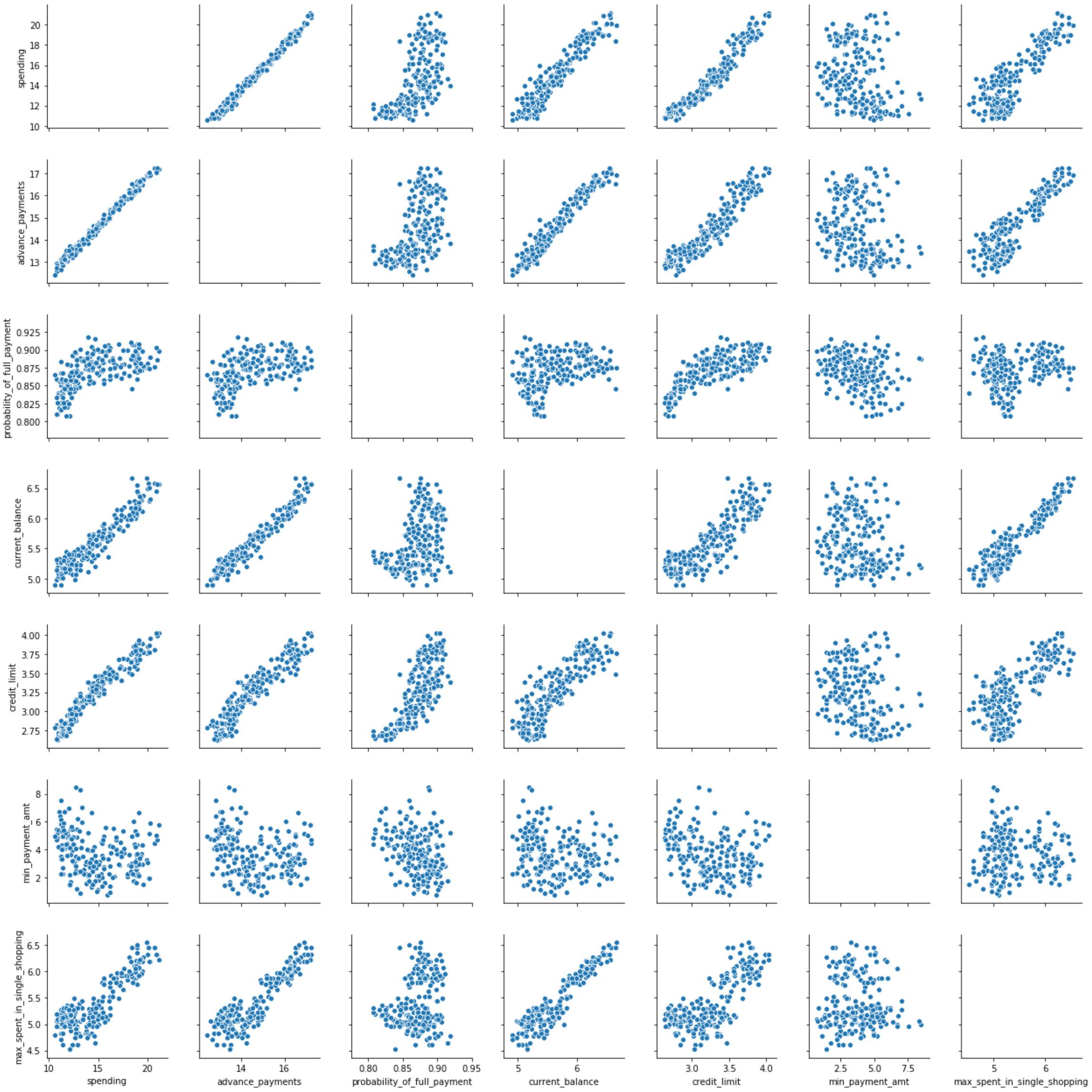

Spending Distribution
Spending Distribution
spending
Advance Payments Distribution
Advance Payments Distribution
advance_payments
Probability of Full Payment Distribution
Probability of Full Payment Distribution
probability_of_full_payment
Current Balance Distribution
Current Balance Distribution
current_balance
Credit Limit Distribution
Credit Limit Distribution
credit_limit
Min Payment Amt Distribution
Min Payment Amt Distribution
min_payment_amt
Max Spent in single shopping
Max Spent in single shopping
max_spent_in_single_shopping
max_spent_in_single_shopping

Inferences : We know that Data is not NORMALLY DISTRIBUTED as some variables are Right skewed,Left skewed etc and there are no Outliers except two Variables which we already checked,So we can proceed with this Data Set for further analysis.

In [ ]:

In [29]:

```
# PairPlot - to check the Variable importance.

sns.pairplot (df , diag_kind='hue');
plt.show()
```



Inferences : Maximum features are very much correlated with each other.

In [ ]:

In [30]:

```
# Heat Map - Get the correlation.
```

In [32]:

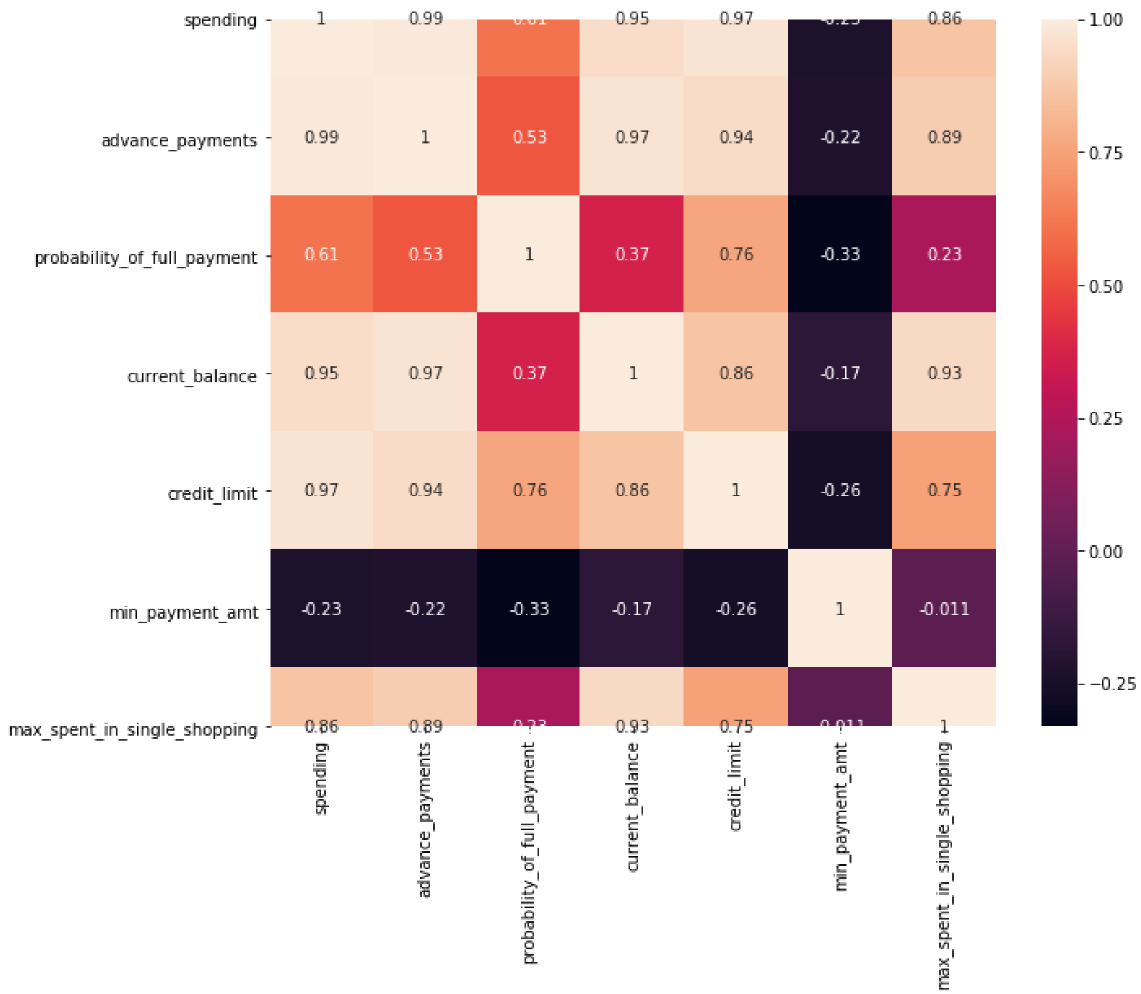
```
df.corr (method='pearson')
```

Out[32]:

	spending	advance_payments	probability_of_full_payment	curre
spending	1.000000	0.994341	0.608288	
advance_payments	0.994341	1.000000	0.529244	
probability_of_full_payment	0.608288	0.529244	1.000000	
current_balance	0.949985	0.972422	0.367915	
credit_limit	0.970771	0.944829	0.761635	
min_payment_amt	-0.229572	-0.217340	-0.331471	
max_spent_in_single_shopping	0.863693	0.890784	0.226825	

In [37]:

```
plt.subplots (figsize=(10,8))
sns.heatmap (df.corr(),annot=True);
```



Inferences :

As we can see Maximum features are highly correlated with each other like:

Spending with advance_payments probability_of_full_payment current_balance credit_limit and max_spent_in_single_shopping.

Advance Payments with spending,current_balance,credit_limit and max_spent_in_single_shopping.

Probability of full payment with spending and advance_payments.

Current balance with spending advance_payments,credit_limit and max_spent_in_single_shopping.

Credit limit with spending advance_payments probability_of_full_payment current_balance and max_spent_in_single_shopping.

Max spent in single shopping with spending advance_payments,current_balance and credit_limit.

In [ ]:

```

```

1.2 Do you think scaling is necessary for clustering in this case? Justify

First, we understand, what is Scaling, in simple words we want equality in our Data set which means all the magnitudes, values or numbers in the given data must be equal or somehow same in one line of number.

When Scaling is necessary: When we know the magnitudes are different for every given column and the Variance will also be different.

So, in this Case we will be doing Scaling.

Justification: We can see that Variance are quite similar in range for every column, but magnitudes are different hence we will be performing Scaling.

So, after scaling we will get one similar range magnitudes for every column so that we will performing Clustering which will lead to give the proper output with good accuracy as when we will provide same magnitudes for every column then we can trust the Output which we can get as Clusters.

In [43]:

```
# lets scale the Data.

from sklearn.preprocessing import StandardScaler

sc = StandardScaler ()
scaled_df = sc.fit_transform (df)
scaled_df
```

Out[43]:

```
array([[ 1.75435461,  1.81196782,  0.17822987, ...,  1.33857863,
        -0.29880602,  2.3289982 ],
       [ 0.39358228,  0.25383997,  1.501773  , ...,  0.85823561,
        -0.24280501, -0.53858174],
       [ 1.41330028,  1.42819249,  0.50487353, ...,  1.317348  ,
        -0.22147129,  1.50910692],
       ...,
       [-0.2816364 , -0.30647202,  0.36488339, ..., -0.15287318,
        -1.3221578 , -0.83023461],
       [ 0.43836719,  0.33827054,  1.23027698, ...,  0.60081421,
        -0.95348449,  0.07123789],
       [ 0.24889256,  0.45340314, -0.77624835, ..., -0.07325831,
        -0.70681338,  0.96047321]])
```

1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

Clustering means to group all the variables which having different information so that we can define the information. It is an unsupervised learning where we do not know the Dependent and Independent Variables. Hierarchical Clustering assumes every row is a single cluster and then by doing mathematical calculations it gives one good cluster.

In [44]:

```
# Calling the libraries for Hierarchical Clustering.

from scipy.cluster.hierarchy import dendrogram,linkage
```

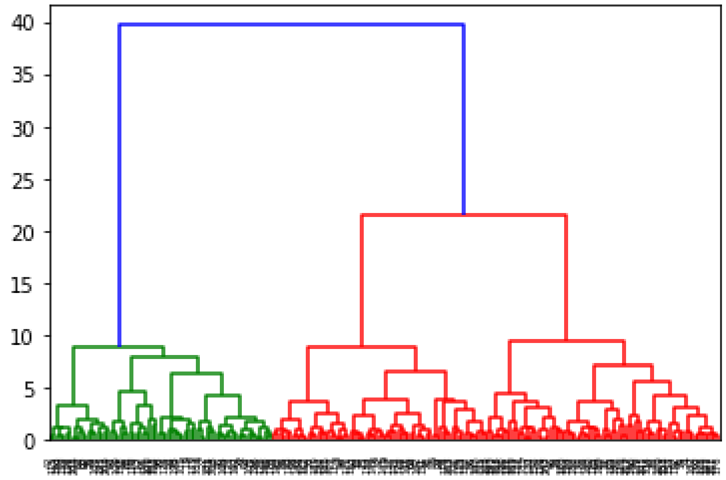
In [45]:

```
wardlink = linkage (scaled_df,method='ward') # Calculates the SSW within sum of square
 for every cluster and then compare with other clusters,whichever is having minimum dis
tance,it is considered.
```

Identifying number of clusters using Dendrogram.

In [46]:

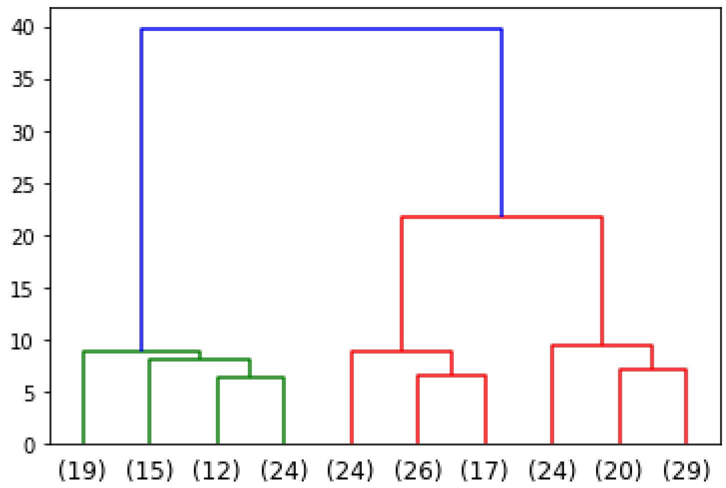
```
dend = dendrogram (wardlink)
```



We can see that there are 2 Clusters form, having colour in RED which is covering maximum information and Green is having minimum information.

In [47]:

```
# lets check the last 10 clusters for the above.

dend = dendrogram (wardlink,truncate_mode='lastp',p=10)
```



In [48]:

```
# We have form the 2 Clusters , lets check the different techniques for identifing the
 clusters numbers.

from scipy.cluster.hierarchy import fcluster

# Method 1.
clusters = fcluster (wardlink,2,criterion='maxclust')
clusters
```

Out[48]:

```
array([1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,
       1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1,
       2, 2, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1,
       1, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1,
       1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1,
       2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2,
       2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1,
       2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2,
       1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2], dtype=int32)
```

In [49]:

```
clusters = fcluster (wardlink,3,criterion='maxclust')
clusters
```

Out[49]:

```
array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,
       1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,
       1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,
       1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,
       3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
       3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,
       3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,
       3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
       1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)
```

In [50]:

```
# Method 2.

clusters = fcluster (wardlink,13,criterion='distance')
clusters
```

Out[50]:

```
array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,
       1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,
       1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,
       1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,
       3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
       3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,
       3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,
       3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
       1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)
```

In [51]:

```
clusters = fcluster (wardlink , 10 , criterion='distance')
clusters
```

Out[51]:

```
array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,
       1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,
       1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,
       1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,
       3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
       3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,
       3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,
       3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
       1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)
```

Now we will add all 3 Clusters in our original Data set so that we can define the existing data which relate all the clusters.

In [ ]:

```

```

In [52]:

```
# lets add the clusters to the original data.

df['clusters'] = clusters
df.head() # checking the Data with newly added clusters.
```

Out[52]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mi
0	19.94	16.92	0.8752	6.675	3.763	
1	15.99	14.89	0.9064	5.363	3.582	
2	18.95	16.42	0.8829	6.248	3.755	
3	10.83	12.96	0.8099	5.278	2.641	
4	17.99	15.86	0.8992	5.890	3.694	

In [53]:

```
# we will create a new CSV file for better understanding and save it in our local syste
m.

df.to_csv ('new_df.csv')
```

In [55]:

```
# Lets check the Data.

clust_newdf = pd.read_csv ('new_df.csv')
clust_newdf.head()
```

Out[55]:

	Unnamed: 0	spending	advance_payments	probability_of_full_payment	current_balance	cre
0	0	19.94	16.92	0.8752	6.675	
1	1	15.99	14.89	0.9064	5.363	
2	2	18.95	16.42	0.8829	6.248	
3	3	10.83	12.96	0.8099	5.278	
4	4	17.99	15.86	0.8992	5.890	

Describing the Clusters:

So we have identified clusters for the given data set by using dendrogram and also validate by using 2 techniques.

We can draw a horizontal line on distance 13 to make "3 CLUSTERS".

Cluster 1 have good amount of probability of full payment to the bank and the credit limit is very high for this cluster.

Cluster 2 have less amount of probability for full payment to the bank as compared to the cluster 1 and credit limit is less.

Cluster 3 also have a good amount of probability for full payment to the bank with equal amount of credit limit as compared to cluster 1

We can draw a horizontal line on distance 13 to make "3 CLUSTERS".

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

K-means clustering: At the beginning, we will specify how many clusters require and then we will proceed for the validation.

In [56]:

```
from sklearn.cluster import KMeans
```

In [62]:

```
kmeans = KMeans (n_clusters=1)
kmeans.fit(scaled_df)
kmeans.inertia_
```

Out[62]:

```
1470.0
```

In [57]:

```
kmeans = KMeans (n_clusters=2)
kmeans.fit(scaled_df) # fitting the scaled data.
kmeans.labels_
```

Out[57]:

```
array([1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1])
```

In [58]:

```
kmeans.inertia_ # WSS for 2 clusters is 659.17.
```

Out[58]:

```
659.1717544870407
```

In [59]:

```
# Lets try for number of clusters 3, 4 and 5.
# for 3 clusters.
kmeans = KMeans (n_clusters=3)
kmeans.fit(scaled_df)
kmeans.inertia_
```

Out[59]:

```
430.65897315130053
```

In [60]:

```
# for 4 clusters.

kmeans = KMeans (n_clusters=4)
kmeans.fit(scaled_df)
kmeans.inertia_
```

Out[60]:

```
371.24193066313256
```

In [61]:

```
# for 5 clusters.
kmeans = KMeans (n_clusters=5)
kmeans.fit(scaled_df)
kmeans.inertia_
```

Out[61]:

```
327.3281094192773
```

For all above 5 clusters Inertia gives us a clear idea that there is a significant drop from cluster 1 to cluster 2 of around 811 and from cluster 2 to cluster 3 of 229 and for cluster 3 to 4 of 59 and from cluster 4 to cluster 5 of 44.

SO we can only take those inertia which is having a huge drop in their inertia values which we can confirmed from cluster 1 and 2 and we can also take the cluster 3 as the drop is maximum but we can exclude the cluster 4 and 5.

Clusters	Inertia	Can Take
1	1470	Yes
2	659	Yes
3	430	Yes
4	371	NO
5	327	NO

We will apply elbow curve to visualize the maximum number of clusters we can take.

In [63]:

```
wssr = []
```

In [64]:

```
wssr
```

Out[64]:

```
[]
```

In [66]:

```
for i in range (1,11):
    KM = KMeans (n_clusters=i)
    KM.fit(scaled_df)
    wssr.append (KM.inertia_)
```

In [67]:

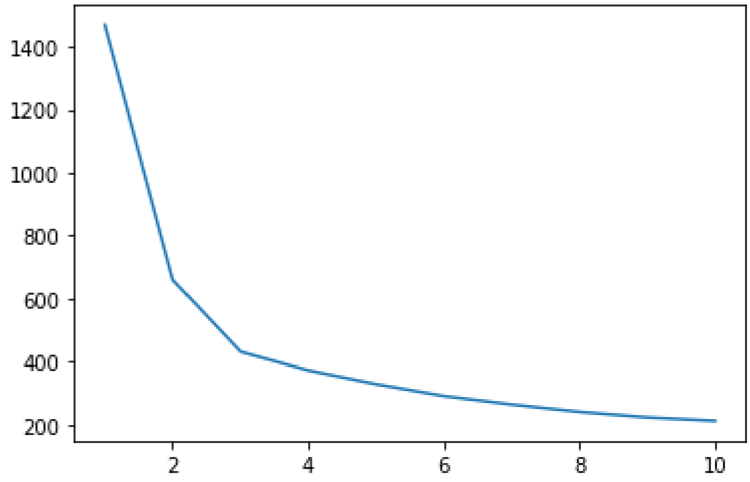
```
wssr
```

Out[67]:

```
[1470.0,
 659.1717544870407,
 430.65897315130053,
 371.301721277542,
 327.4732055881967,
 289.98904583584886,
 263.0910453972738,
 239.65012195641214,
 222.13439853914366,
 211.20447534124924]
```

In [79]:

```
plt.plot (range (1,11),wssr);
```



As we can see that there is a Significant drop from cluster 1 to 2 and a noticeable drop from cluster 2 to cluster 3.

Hence as we proved earlier that we can SELECT 3 CLUSTERS.

Silhouette score : by applying we can check how good our clusters are separated.

Benchmark - if score is +1 then good, if score is 0 then ok, if score is -1 then not good.

In [73]:

```
kmeans = KMeans (n_clusters=3)
kmeans.fit (scaled_df)
clust = kmeans.labels_
```

In [74]:

```
clust
```

Out[74]:

```
array([2, 1, 2, 0, 2, 0, 0, 1, 2, 0, 2, 1, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0,
       2, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 2, 2, 1, 2, 2,
       0, 0, 1, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 0, 0, 2, 1, 0, 0, 1, 1, 2,
       2, 1, 2, 0, 1, 0, 2, 2, 0, 2, 1, 0, 2, 1, 1, 1, 1, 2, 0, 1, 2, 1,
       2, 0, 1, 2, 1, 0, 0, 2, 2, 2, 0, 2, 1, 2, 1, 2, 1, 2, 2, 0, 0, 2,
       1, 1, 2, 0, 0, 2, 1, 1, 0, 2, 1, 0, 0, 0, 1, 1, 2, 0, 1, 1, 0, 1,
       1, 2, 0, 2, 2, 0, 2, 1, 1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 1,
       0, 1, 1, 0, 1, 2, 2, 0, 2, 2, 2, 0, 1, 1, 1, 0, 1, 0, 1, 2, 2, 2,
       1, 0, 1, 0, 1, 1, 1, 1, 2, 2, 0, 1, 1, 0, 0, 1, 0, 2, 1, 2, 2, 0,
       2, 0, 1, 2, 1, 0, 2, 1, 2, 1, 1, 1])
```

In [75]:

```
from sklearn.metrics import silhouette_samples , silhouette_score
```

In [77]:

```
silhouette_score (scaled_df,clust)
```

Out[77]:

```
0.4007270552751299
```

Score is 40 and it is positive, so our clusters are separated properly.

In [78]:

```
sil_width = silhouette_samples (scaled_df,clust)
df ['sil_width'] = sil_width
df ['clust'] = clust
df.head()
```

Out[78]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mi
0	19.94	16.92	0.8752	6.675	3.763	
1	15.99	14.89	0.9064	5.363	3.582	
2	18.95	16.42	0.8829	6.248	3.755	
3	10.83	12.96	0.8099	5.278	2.641	
4	17.99	15.86	0.8992	5.890	3.694	

Interpret Inferences are:

Our clusters are very well separated from each other as there is no jumping of magnitudes. We have checked the inertia which had given us the clear idea that there is a good significant drop from cluster 1 to 2 and to 3. Also, we have visualised by using the Elbow curve and our Silhouette score is 40 which is very good for cluster separation.

1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.in context to the business problem in-hand

Cluster 1:

Per month spending for this cluster is maximum which is on average of 18,000. They paid almost 1600 advance in cash. 88 % customer pay their full payment for the month to the bank (so bank have good recovery rate from this cluster). At the end of the month this customer has on average of 6000 remaining in their account. Credit Limit for every customer is around 30000 for this cluster. They paid around 300 for every minimum purchases. They spent around 6000 for single purchase at one time.

Promotional strategies from BANK:

After considering all the above points, I would suggest bank to provide their promotional/product offers which will in range of 5000 to 12000.

Cluster 2:

Per month spending for this cluster is maximum which is on average of 12,000. They paid almost 1300 advance in cash. 84 % customer pay their full payment for the month to the bank (Bank can increase the rate by providing some easy solutions). At the end of the month this customer has on average of 5000 remaining in their account. Credit Limit for every customer is around 20000 for this cluster. They paid around 500 for every minimum purchases. They spent around 5000 for single purchase at one time.

Promotional strategies from BANK:

After considering all the above points, I would suggest bank to provide their promotional/product offers which will in range of 2600 to 6000. (we can increase the initial rate once recovery of payment will go above 85 %).

Cluster 3:

Per month spending for this cluster is maximum which is on average of 14,000. They paid almost 1400 advance in cash. 87 % customer pay their full payment for the month to the bank (so bank have good recovery rate from this cluster). At the end of the month this customer has on average of 5000 remaining in their account. Credit Limit for every customer is around 30000 for this cluster. They paid around 200 for every minimum purchases. They spent around 5000 for single purchase at one time.

Promotional strategies from BANK:

After considering all the above points, I would suggest bank to provide their promotional/product offers which will in range of 4000 to 9000.

Conclusion for all 3 Clusters to the Bank.

Clusters	Promotional Offers in Rupees
1	5000 to 12000
2	2600 to 6000
3	4000 to 9000

```
==========================================================================================
```

Problem 2: CART-RF-ANN

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. You are assigned the task to make a model which predicts the claim status and provide recommendations to management. Use CART, RF & ANN and compare the models' performances in train and test sets.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check.Interpret the inferences from the descriptive statistics in a detailed manner.

In [81]:

```
# Lets pull the Data and Read it.

cra = pd.read_csv (r'E:\Great Learning\Projects\Data Mining- Clusters,CART,RF,ANN\Data
 sets\insurance_part2_data.csv')
cra.head()
```

Out[81]:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan

1. Target: Claim Status (Claimed)
2. Code of tour firm (Agency_Code)
3. Type of tour insurance firms (Type)
4. Distribution channel of tour insurance agencies (Channel)
5. Name of the tour insurance products (Product)
6. Duration of the tour (Duration)
7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
10. Age of insured (Age)

Descriptive statistics:

In [82]:

```
# We will check the basic information for the above data.

cra.shape # 3000 rows and 10 columns.
```

Out[82]:

```
(3000, 10)
```

In [84]:

```
cra.columns
```

Out[84]:

```
Index(['Age', 'Agency_Code', 'Type', 'Claimed', 'Commision', 'Channel',
       'Duration', 'Sales', 'Product Name', 'Destination'],
      dtype='object')
```

In [88]:

```
cra.describe().T
```

Out[88]:

	count	mean	std	min	25%	50%	75%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Duration	3000.0	70.001333	134.053313	-1.0	11.0	26.50	63.000	4580.00
Sales	3000.0	60.249913	70.733954	0.0	20.0	33.00	69.000	539.00

In [91]:

```
cra.dtypes.value_counts()
```

Out[91]:

```
object     6
float64    2
int64      2
dtype: int64
```

In [93]:

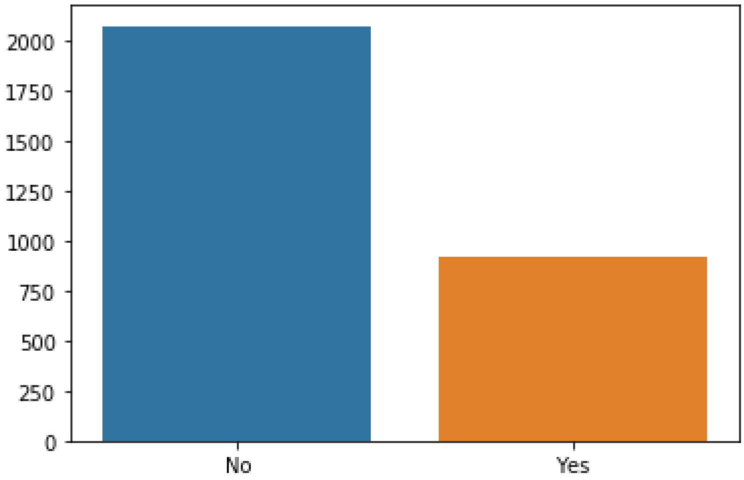
```
cra.Claimed.value_counts()
```

Out[93]:

```
No     2076
Yes     924
Name: Claimed, dtype: int64
```

In [94]:

```
sns.barplot (cra.Claimed.value_counts().index , cra.Claimed.value_counts().values);
plt.show()
print(cra.Claimed.value_counts(normalize=True))
```




```
No     0.692
Yes    0.308
Name: Claimed, dtype: float64
```

In [95]:

```
du = cra.duplicated()
du.sum()
print ('Number of Duplicates in this Data set are %d' % du.sum())
```

```
Number of Duplicates in this Data set are 139
```

Our objective is to build models which are CART,RF and ANN which require INTEGERS as data type and we have different data types,so we will be converting all of them to the INTEGERS.

In [107]:

```
for feature in cra.columns:
    if cra [feature].dtype=='object':
        cra [feature]=pd.Categorical (cra[feature]).codes # coverting all dtypes in Int
egers.
```

In [108]:

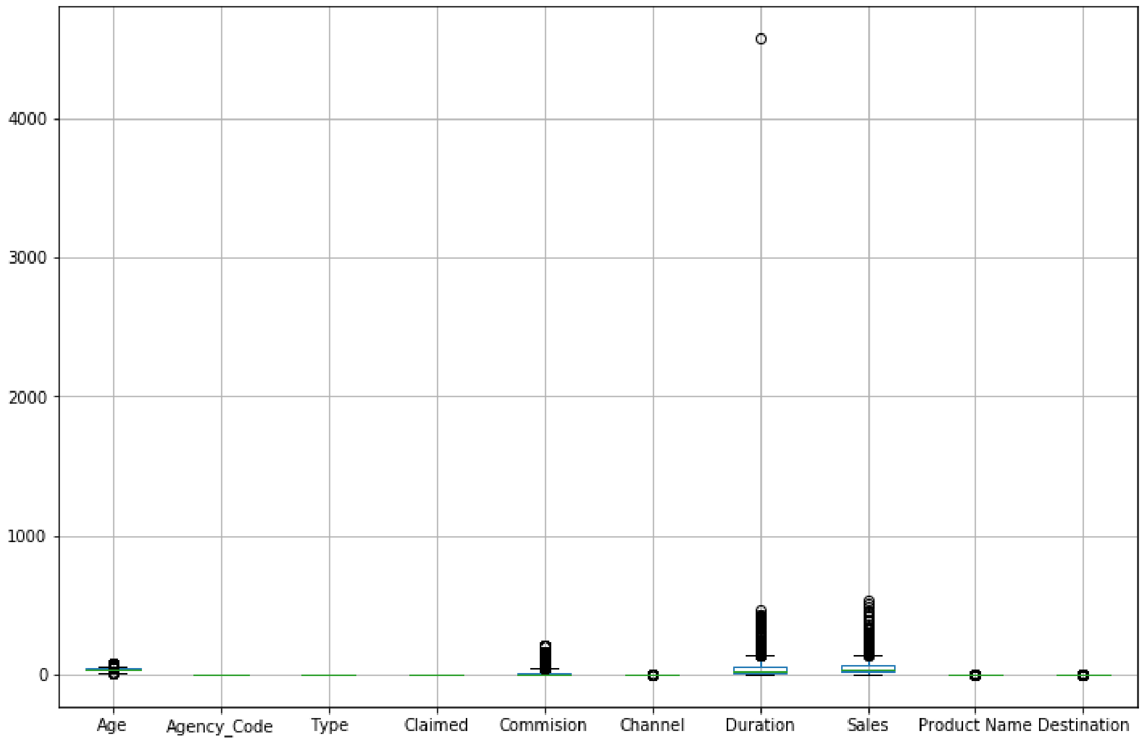
```
cra.dtypes.value_counts() # Now we have all Data types in Numerical.
```

Out[108]:

```
int8       6
int64      2
float64    2
dtype: int64
```

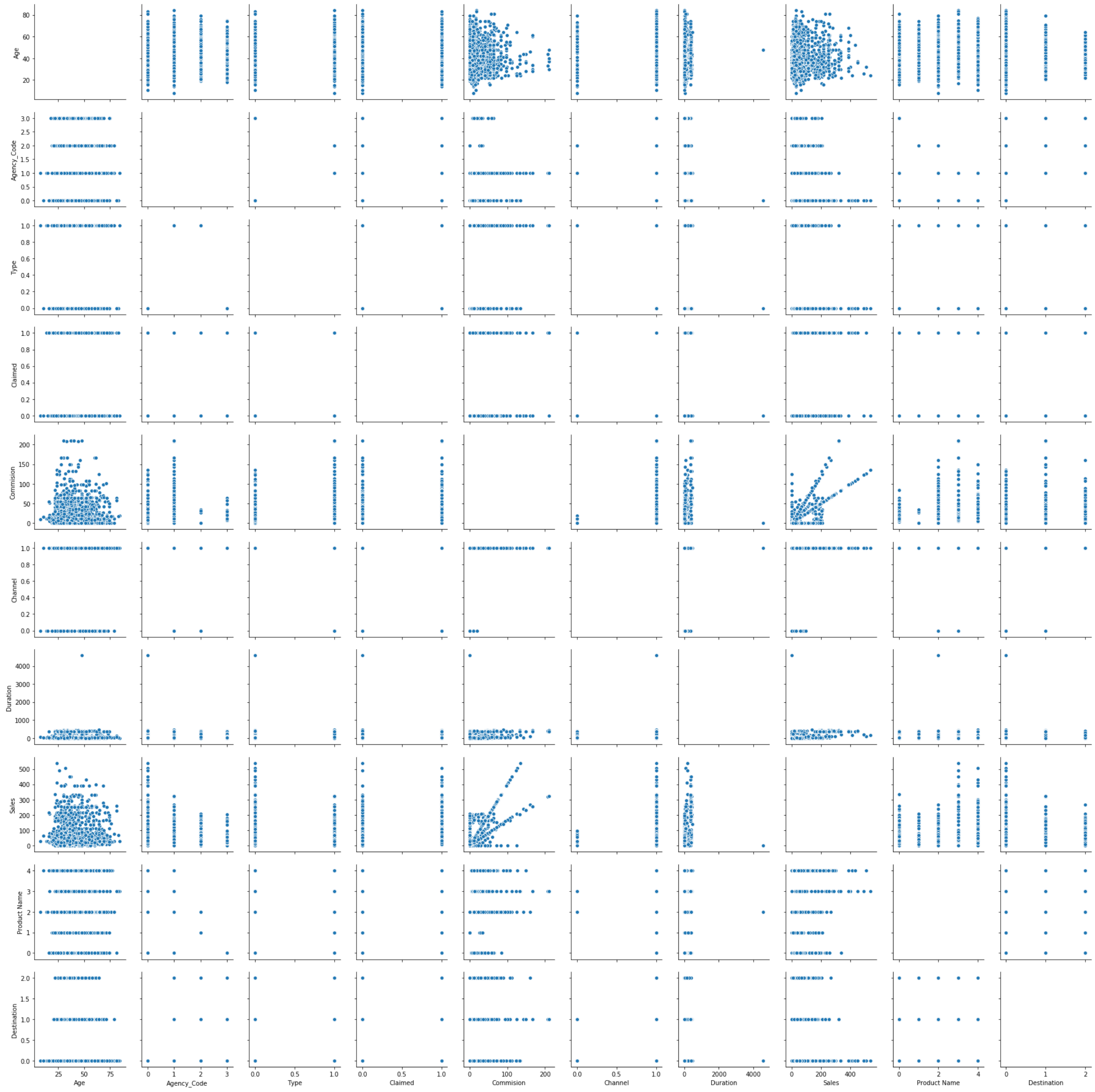
In [109]:

```
# Outliers:

cra.boxplot (figsize=(12,8));
```



In [110]:

```
# Feature relations.

sns.pairplot (cra,diag_kind='hue');
```


Age
Agency_Code
Type
Claimed
Commision
Channel
Duration
Sales
Product Name
Destination

In [111]:

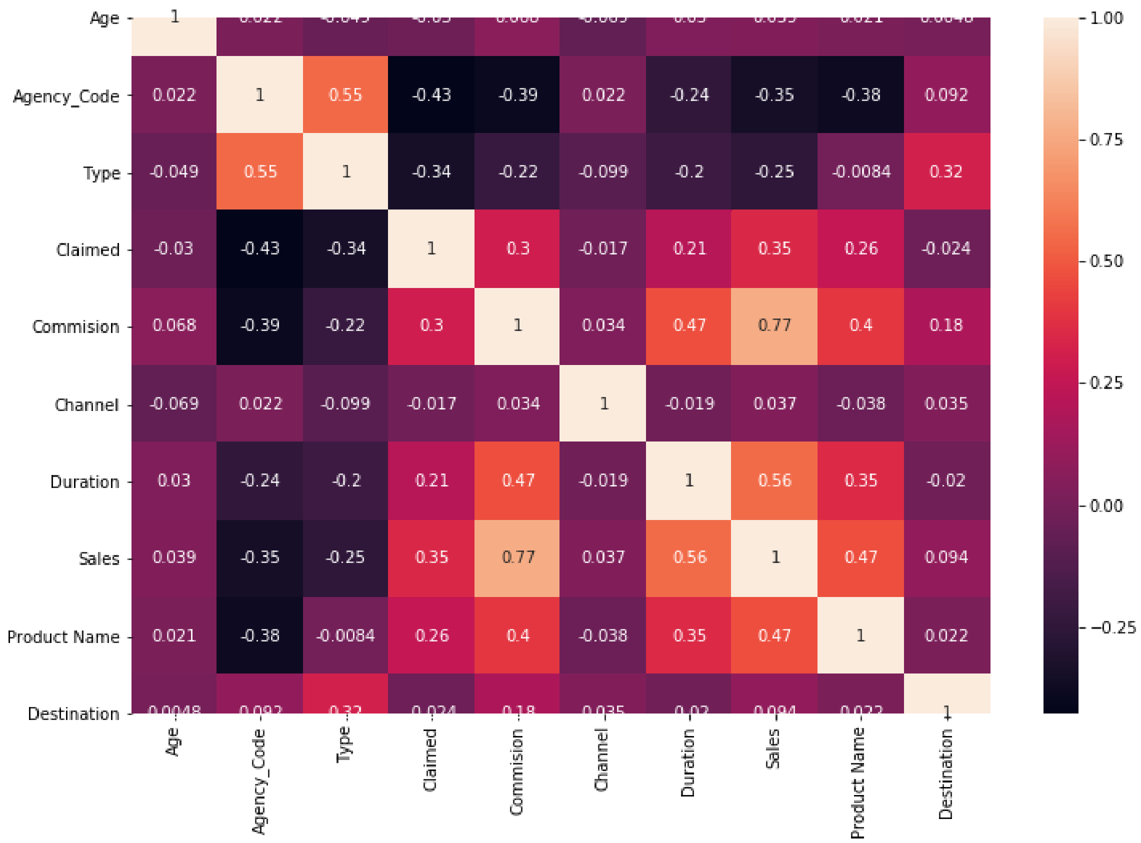
```
# Correlation
cra.corr(method='pearson')
```

Out[111]:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duratio
Age	1.000000	0.021939	-0.048992	-0.030027	0.067717	-0.068927	0.03042
Agency_Code	0.021939	1.000000	0.552247	-0.428647	-0.392585	0.022085	-0.23928
Type	-0.048992	0.552247	1.000000	-0.343505	-0.216599	-0.099291	-0.19821
Claimed	-0.030027	-0.428647	-0.343505	1.000000	0.297498	-0.016641	0.21492
Commision	0.067717	-0.392585	-0.216599	0.297498	1.000000	0.033563	0.47138
Channel	-0.068927	0.022085	-0.099291	-0.016641	0.033563	1.000000	-0.01938
Duration	0.030425	-0.239285	-0.198219	0.214923	0.471389	-0.019389	1.00000
Sales	0.039455	-0.346404	-0.254868	0.348877	0.766505	0.037418	0.55893
Product Name	0.020643	-0.379233	-0.008416	0.257951	0.399306	-0.037859	0.35476
Destination	0.004750	0.092274	0.317289	-0.023851	0.184017	0.035373	-0.01966

In [112]:

```
# Heatmap.
plt.figure(figsize=(12,8))
sns.heatmap (cra.corr(), annot=True),
plt.show()
```



In [103]:

```
# Null Value

cra.isnull().sum() # No missing values.
```

Out[103]:

```
Age             0
Agency_Code     0
Type            0
Claimed         0
Commision       0
Channel         0
Duration        0
Sales           0
Product Name    0
Destination     0
dtype: int64
```

Null Value Condition:

We have checked from the above that there are no NULL (Missing value) in Data set.

Interpret the inferences from the descriptive statistics in a detailed manner.

As of now we have learned the information from the Data set and where we had done a several checks. From null value condition, we had confirmed that there are no missing values in the given data set. There are duplicates in the Data which are 139 (5% of complete Data), which include the repeatative Claimed status,Agency code,Type,Channel,Product name and destination, so each feature represents the given information for one particular row hence we have to keep the exact numbers for every column and row.

Data is Balanced as 70-30, 70 percent consisting of CLAIMED (NO) and 30 percent consisting of CLAIMED (YES). Magnitudes and Variance for all the columns are different from each other, which can either seen in their MEANS also, hence scaling will need for specific modelling. There are Outliers in 7 columns (Age,Commision,Channel,Duration,Sales,Product Name,Destination) we will be treating them for specific model too.

Below Features have Good Correlation which we have seen in Heat-Map. Commission and Sales good correlation (77 %) Duration and Sales good correlation (56 %) Product name and Sales good correlation (47 %) Sales is correlated with commission, duration and product name Duration and commission (47 %) Type and Agency code (55 %).

In [ ]:

```

```

2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network.

In [113]:

```
# Before splitting the data, lets define a new variable x and y where we will keep inde
pendent (x) and dependent (y) variables.

x = cra.drop (['Claimed'],axis=1)
y = cra.pop ('Claimed')
```

In [114]:

```
x.shape
```

Out[114]:

```
(3000, 9)
```

In [115]:

```
y.shape
```

Out[115]:

```
(3000,)
```

In [116]:

```
# Lets split the Data into training and test set, where x will define independent varia
bles and y will define dependent variables.

from sklearn.model_selection import train_test_split
```

In [120]:

```
xtrain, xtest, ytrain, ytest = train_test_split (x, y, test_size = .30, random_state =
8)
```

In [123]:

```
xtrain.shape
```

Out[123]:

```
(2100, 9)
```

In [125]:

```
ytest.shape
```

Out[125]:

```
(900,)
```

In [126]:

```
xtest.shape
```

Out[126]:

```
(900, 9)
```

In [127]:

```
ytrain.shape
```

Out[127]:

```
(2100,)
```

In [ ]:

```
# Building Models.
```

Building Classification Model or Decision Tree (CART) model:

No need to treat the outliers and no need to perform scaling.

In [128]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [130]:

```
dt = DecisionTreeClassifier(criterion='gini',random_state=8)
```

In [131]:

```
dt.fit (xtrain , ytrain)
```

Out[131]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=Non
e,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=8, splitter='best')
```

In [ ]:

```

```

Building Random Forest (RF):

For this model, we don’t need to treat outliers and no requirement for scaling the data.

In [132]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [133]:

```
rf = RandomForestClassifier (n_estimators=50, oob_score=True)
```

In [134]:

```
rf.fit (xtrain, ytrain)
```

Out[134]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gin
i',
                       max_depth=None, max_features='auto', max_leaf_nodes
=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=True, random_state=None,
                       verbose=0, warm_start=False)
```

In [135]:

```
rf.oob_score_ # checking how the model performs.
```

Out[135]:

```
0.7509523809523809
```

Out of Bag score given us value of 75 %, so we can say that Error rate is of 25 % and our accuracy is 75 %, we can also increase it by fine tunning the model.

In [ ]:

```

```

Building Artificial Neural Netwrok (ANN):

For this model, we will require to treat the outliers and scaling needs to be done as this is mandatory for this specific model.

Lets create a Copy of our original data set and then we will work for this model.

In [136]:

```
new_cra = cra.copy()
```

In [137]:

```
new_cra.shape
```

Out[137]:

```
(3000, 9)
```

In [144]:

```
# Lets treat the outliers by defining one function.

def remove_outlier (col):
    sorted (col)
    Q1,Q3 = np.percentile (col,[25,75])
    IQR = Q3 - Q1
    lower_range = Q1 - (1.5 * IQR)
    upper_range = Q3 + (1.5 * IQR)
    return lower_range,upper_range
```

In [142]:

```
new_cra.columns
```

Out[142]:

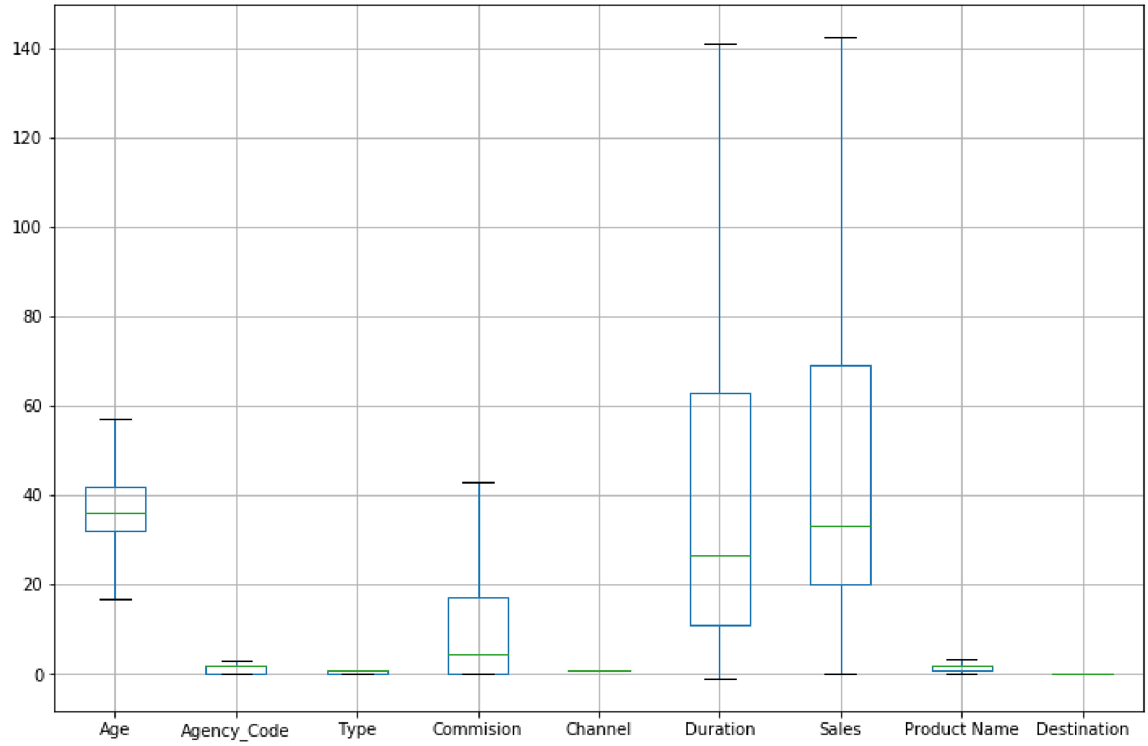
```
Index(['Age', 'Agency_Code', 'Type', 'Commision', 'Channel', 'Duration',
       'Sales', 'Product Name', 'Destination'],
      dtype='object')
```

In [145]:

```
lage , uage = remove_outlier (new_cra ['Age'])
new_cra ['Age'] = np.where (new_cra ['Age'] > uage,uage, new_cra ['Age'])
new_cra ['Age'] = np.where (new_cra ['Age'] < lage,lage, new_cra ['Age'])

lcom , ucom = remove_outlier (new_cra ['Commision'])
new_cra ['Commision'] = np.where (new_cra ['Commision'] > ucom, ucom, new_cra ['Commisi
on'])
new_cra ['Commision'] = np.where (new_cra ['Commision'] < lcom, lcom, new_cra ['Commisi
on'])

lch , uch = remove_outlier (new_cra ['Channel'])
new_cra ['Channel'] = np.where (new_cra ['Channel'] > uch, uch, new_cra ['Channel'])
new_cra ['Channel'] = np.where (new_cra ['Channel'] < lch, lch, new_cra ['Channel'])

ldu, udu = remove_outlier (new_cra ['Duration'])
new_cra ['Duration'] = np.where (new_cra ['Duration'] > udu , udu , new_cra ['Duration'
])
new_cra ['Duration'] = np.where (new_cra ['Duration'] < ldu, ldu, new_cra ['Duration'])

lsal, usal = remove_outlier (new_cra ['Sales'])
new_cra ['Sales'] = np.where (new_cra ['Sales'] > usal, usal, new_cra ['Sales'])
new_cra ['Sales']= np.where (new_cra ['Sales'] < lsal, lsal, new_cra ['Sales'])

lpr, upr = remove_outlier (new_cra ['Product Name'])
new_cra ['Product Name'] = np.where (new_cra ['Product Name'] > upr, upr , new_cra ['Pr
oduct Name'])
new_cra ['Product Name'] = np.where (new_cra ['Product Name'] < lpr, lpr, new_cra ['Pro
duct Name'])

udes, ldes = remove_outlier (new_cra ['Destination'])
new_cra ['Destination'] = np.where (new_cra ['Destination'] > udes , udes, new_cra ['De
stination'])
new_cra ['Destination'] = np.where (new_cra ['Destination'] < ldes, ldes, new_cra ['Des
tination'])
```

In [146]:

```
new_cra.boxplot (figsize=(12,8));
```



So we have treated the Outliers.

In [ ]:

```

```

Now we will be scaling the Data set.

In [147]:

```
from sklearn.preprocessing import StandardScaler
```

In [148]:

```
sc = StandardScaler()
```

In [150]:

```
xtrain = sc.fit_transform (xtrain)
```

In [151]:

```
xtest = sc.transform (xtest)
```

In [152]:

```
xtrain
```

Out[152]:

```
array([[-0.65698385, -0.31079598,  0.79320231, ..., -0.56423715,
         0.2763925 , -0.4417093 ],
       [-0.17740026,  0.69641321,  0.79320231, ..., -0.56137973,
         0.2763925 , -0.4417093 ],
       [-1.23248416, -1.31800517, -1.26071242, ..., -0.51137483,
        -1.3181796 , -0.4417093 ],
       ...,
       [-0.56106713, -0.31079598,  0.79320231, ...,  0.28441737,
         0.2763925 ,  1.2888682 ],
       [-1.80798447,  0.69641321,  0.79320231, ...,  0.29584706,
        -0.52089355, -0.4417093 ],
       [-0.17740026,  0.69641321,  0.79320231, ..., -0.21848901,
         0.2763925 , -0.4417093 ]])
```

In [153]:

```
xtest
```

Out[153]:

```
array([[-0.17740026,  0.69641321,  0.79320231, ..., -0.33278592,
         0.2763925 , -0.4417093 ],
       [-0.17740026, -1.31800517, -1.26071242, ..., -0.5828104 ,
         1.8709646 , -0.4417093 ],
       [-1.23248416, -0.31079598,  0.79320231, ..., -0.43279571,
         0.2763925 , -0.4417093 ],
       ...,
       [ 2.60418459, -0.31079598,  0.79320231, ..., -0.4185086 ,
         0.2763925 , -0.4417093 ],
       [-0.17740026,  0.69641321,  0.79320231, ..., -0.50423127,
         0.2763925 , -0.4417093 ],
       [-0.17740026,  0.69641321,  0.79320231, ...,  0.13868882,
         0.2763925 ,  1.2888682 ]])
```

So we have Scaled the given Data and now we can build our ANN model.

In [154]:

```
from sklearn.neural_network import MLPClassifier
```

In [155]:

```
mlp = MLPClassifier (hidden_layer_sizes=50 , max_iter=500, solver='adam', verbose=True,
tol = 0.001)
```

In [157]:

```
mlp.fit (xtrain , ytrain)
```

```
Iteration 1, loss = 0.66734678
Iteration 2, loss = 0.61234238
Iteration 3, loss = 0.57359428
Iteration 4, loss = 0.54652214
Iteration 5, loss = 0.52685568
Iteration 6, loss = 0.51378469
Iteration 7, loss = 0.50466535
Iteration 8, loss = 0.49887967
Iteration 9, loss = 0.49445303
Iteration 10, loss = 0.49165051
Iteration 11, loss = 0.48983594
Iteration 12, loss = 0.48844674
Iteration 13, loss = 0.48740281
Iteration 14, loss = 0.48627778
Iteration 15, loss = 0.48561194
Iteration 16, loss = 0.48515982
Iteration 17, loss = 0.48461145
Iteration 18, loss = 0.48416163
Iteration 19, loss = 0.48376906
Iteration 20, loss = 0.48345773
Iteration 21, loss = 0.48283792
Iteration 22, loss = 0.48262768
Iteration 23, loss = 0.48210111
Iteration 24, loss = 0.48181265
Iteration 25, loss = 0.48151750
Training loss did not improve more than tol=0.001000 for 10 consecutive ep
ochs. Stopping.
```

Out[157]:

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=
0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=50, learning_rate='constant',
              learning_rate_init=0.001, max_iter=500, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=None, shuffle=True, solver='adam', tol=0.001,
              validation_fraction=0.1, verbose=True, warm_start=False)
```

In [ ]:

```

```

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model.

We will get all the Perofrmance measures on every model.

1: Classification Model (CART): Decision Tree.

In [159]:

```
# Lets Fine tune the model by upgrading the features.

dt_fine = DecisionTreeClassifier (criterion='gini',max_depth=5,max_features=5,random_st
ate=8)
```

In [160]:

```
dt_fine.fit (xtrain,ytrain)
```

Out[160]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
                       max_features=5, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=8, splitter='best')
```

In [161]:

```
y_train_pre_dt = dt_fine.predict (xtrain)

y_test_pre_dt = dt_fine.predict (xtest)
```

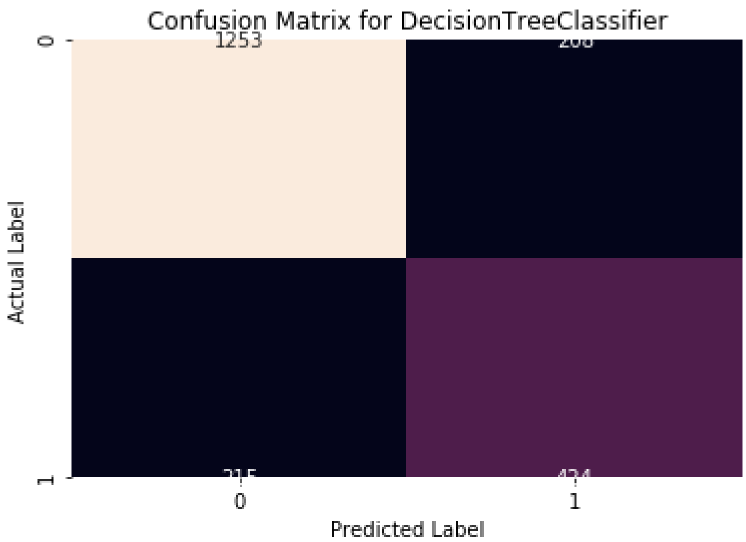
In [163]:

```
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc
_curve,accuracy_score
```

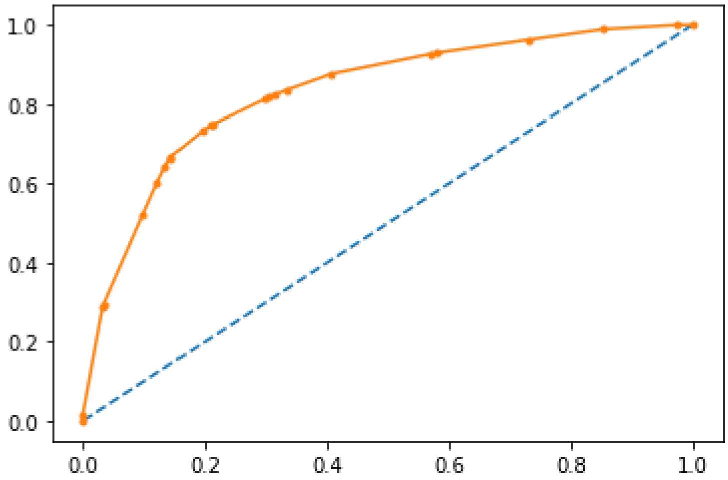
In [200]:

```
# Performance measures on Train set.

models_names={dt_fine:'DecisionTreeClassifier'}

print('Accuracy for {} model is'.format(models_names[dt_fine]),'\n',accuracy_score(ytra
in,y_train_pre_dt))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[dt_fine]),'\n',roc_auc_score
(ytrain,y_train_pre_dt))

print('\n')

print('Classification report for {} model is'.format(models_names[dt_fine]),'\n',classi
fication_report(ytrain,y_train_pre_dt))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[dt_fine]),'\n',confusion_m
atrix(ytrain,y_train_pre_dt))

print ('\n')


print('Confusion Matrix for {} model is'.format(models_names[dt_fine]))


sns.heatmap(confusion_matrix(ytrain,y_train_pre_dt),annot=True,fmt='d',cbar=False)
plt.title('Confusion Matrix for {}'.format(models_names[dt_fine]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = dt_fine.predict_proba (xtrain)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytrain, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytrain, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for DecisionTreeClassifier model is
 0.7985714285714286


ROC AUC score for DecisionTreeClassifier model is
 0.7605842676409815


Classification report for DecisionTreeClassifier model is
               precision    recall  f1-score   support

           0       0.85      0.86      0.86      1461
           1       0.67      0.66      0.67       639

    accuracy                           0.80      2100
   macro avg       0.76      0.76      0.76      2100
weighted avg       0.80      0.80      0.80      2100



Confusion Matrix for DecisionTreeClassifier model is
 [[1253  208]
 [ 215  424]]


Confusion Matrix for DecisionTreeClassifier model is
```




```
AUC: 0.831
```

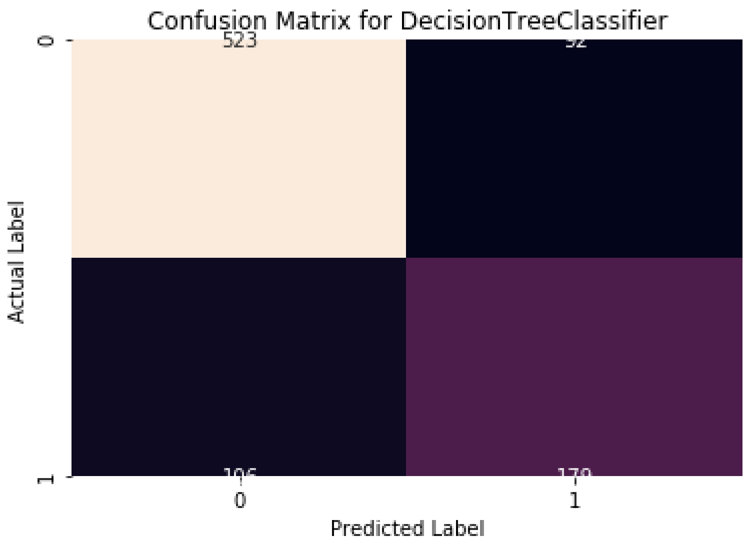


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

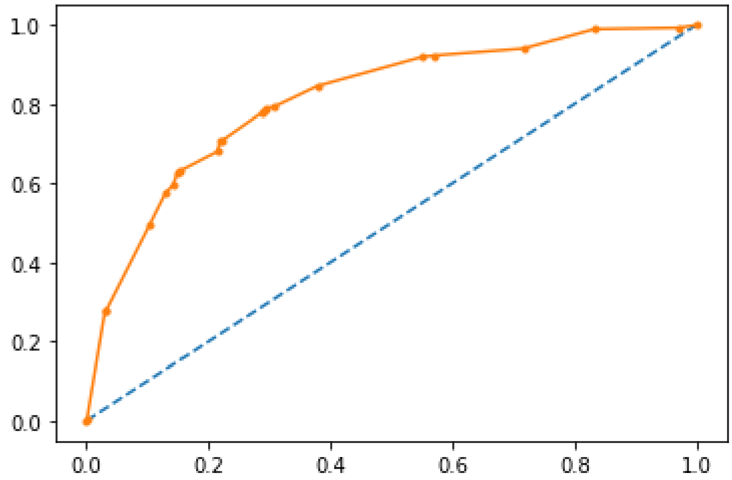
In [202]:

```
# Performance measures on Test set.

models_names={dt_fine:'DecisionTreeClassifier'}

print('Accuracy for {} model is'.format(models_names[dt_fine]),'\n',accuracy_score(ytes
t,y_test_pre_dt))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[dt_fine]),'\n',roc_auc_score
(ytest,y_test_pre_dt))

print('\n')

print('Classification report for {} model is'.format(models_names[dt_fine]),'\n',classi
fication_report(ytest,y_test_pre_dt))

print('\n')


print('Confusion Matrix for {} model is'.format(models_names[dt_fine]),'\n',confusion_m
atrix(ytest,y_test_pre_dt))

print ('\n')

print('Confusion Matrix for {} model is'.format(models_names[dt_fine]))


sns.heatmap(confusion_matrix(ytest,y_test_pre_dt),annot=True,fmt='d',cbar=False)
plt.title('Confusion Matrix for {}'.format(models_names[dt_fine]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = dt_fine.predict_proba (xtest)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytest, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytest, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for DecisionTreeClassifier model is
 0.78


ROC AUC score for DecisionTreeClassifier model is
 0.7392383397518185


Classification report for DecisionTreeClassifier model is
               precision    recall  f1-score   support

           0       0.83      0.85      0.84       615
           1       0.66      0.63      0.64       285

    accuracy                           0.78       900
   macro avg       0.75      0.74      0.74       900
weighted avg       0.78      0.78      0.78       900



Confusion Matrix for DecisionTreeClassifier model is
 [[523  92]
 [106 179]]


Confusion Matrix for DecisionTreeClassifier model is
```




```
AUC: 0.812
```



Inferences for Classification Model (CART) Model:Decision Tree.

Only checking for 1's. Performance Metrics on Training set has given us Accuracy of 79.85 %. ROC AUC score of 76 %. Recall 66 %. Precision 67 %. f1 score is also 67 %. AUC is 83 %.

False positives (FP) are 215 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1.

Only checking for 1's. Performance Metrics on Test set has given us Accuracy of 78 %. ROC AUC score of 73 %. Recall 63 %. Precision 66 %. f1 score is also 64 %. AUC is 81 %.

False positives (FP) are 106 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1.

In [ ]:

```

```

In [ ]:

```

```

2: Random Forest (RF):

In [ ]:

```

```

In [170]:

```
# Lets fine tune the model by upgrading the Features.

rf_ft = RandomForestClassifier (n_estimators=1000,
                                max_depth=5,
                                max_features=8,
                               min_samples_leaf=300,
                               min_samples_split=450,
                               random_state=8,
                               oob_score=True)
```

In [171]:

```
rf_ft = rf_ft.fit (xtrain, ytrain)
```

In [172]:

```
rf_ft.oob_score_
```

Out[172]:

```
0.7566666666666667
```

In [173]:

```
# Lets check how accurate our model with all the above values are by performing GRID SE
ARCH.

from sklearn.model_selection import GridSearchCV
```

In [174]:

```
param_grid = {'n_estimators': [501,1008],
             'max_depth': [4,6],
             'max_features': [5,8],
             'min_samples_leaf': [300,600],
             'min_samples_split': [900,1800],
             }
```

In [175]:

```
rf_ft = RandomForestClassifier()
```

In [176]:

```
grid_search_rf = GridSearchCV (estimator=rf_ft, param_grid= param_grid, cv=3)
```

In [177]:

```
grid_search_rf.fit (xtrain, ytrain)
```

Out[177]:

```
GridSearchCV(cv=3, error_score='raise-deprecating',
             estimator=RandomForestClassifier(bootstrap=True, class_weight
=None,
                                              criterion='gini', max_depth=
None,
                                              max_features='auto',
                                              max_leaf_nodes=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.
0,
                                              n_estimators='warn', n_jobs=
None,
                                              oob_score=False,
                                              random_state=None, verbose=
0,
                                              warm_start=False),
             iid='warn', n_jobs=None,
             param_grid={'max_depth': [4, 6], 'max_features': [5, 8],
                         'min_samples_leaf': [300, 600],
                         'min_samples_split': [900, 1800],
                         'n_estimators': [501, 1008]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=Fals
e,
             scoring=None, verbose=0)
```

In [178]:

```
grid_search_rf.best_params_
```

Out[178]:

```
{'max_depth': 4,
 'max_features': 5,
 'min_samples_leaf': 300,
 'min_samples_split': 900,
 'n_estimators': 501}
```

In [179]:

```
grid_search_rf.best_estimator_
```

Out[179]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gin
i',
                       max_depth=4, max_features=5, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=300, min_samples_split=900,
                       min_weight_fraction_leaf=0.0, n_estimators=501,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [180]:

```
best_model = grid_search_rf.best_estimator_
```

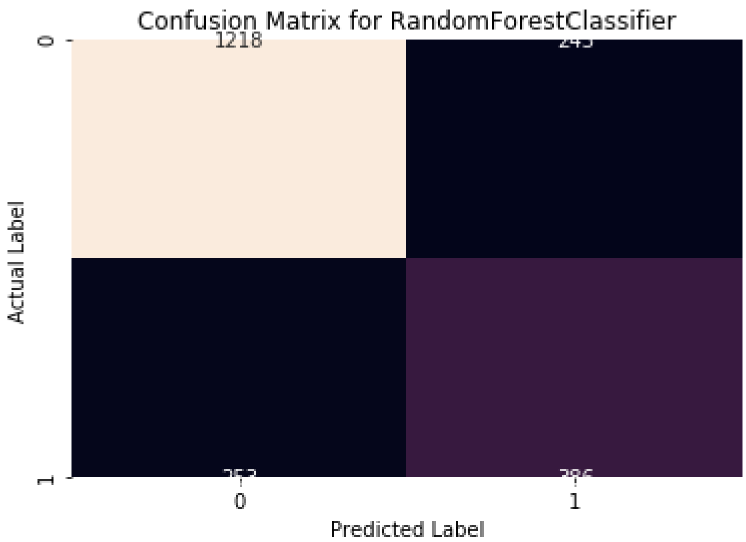
In [182]:

```
ytrain_predict_rf = best_model.predict (xtrain)
ytest_predict_rf = best_model.predict (xtest)
```

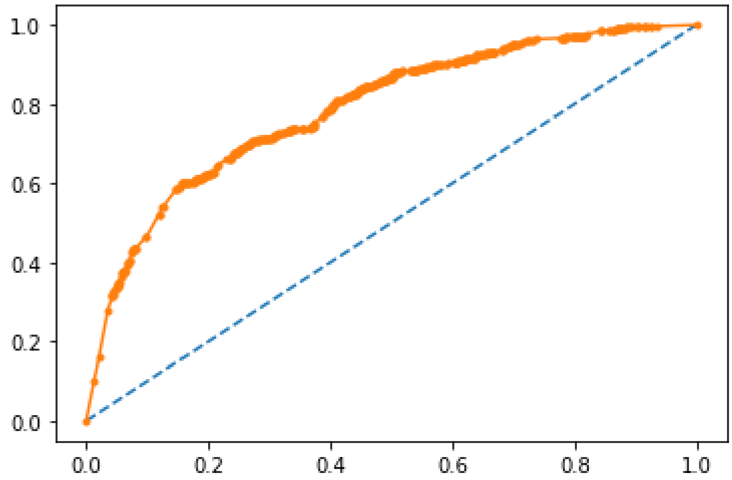
In [205]:

```
# Performance measures on Train set.

models_names={best_model:'RandomForestClassifier'}

print('Accuracy for {} model is'.format(models_names[best_model]),'\n',accuracy_score(y
train,ytrain_predict_rf))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[best_model]),'\n',roc_auc_sco
re (ytrain,ytrain_predict_rf))

print('\n')

print('Classification report for {} model is'.format(models_names[best_model]),'\n',cla
ssification_report(ytrain,ytrain_predict_rf))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[best_model]),'\n',confusio
n_matrix(ytrain,ytrain_predict_rf))

print ('\n')


print('Confusion Matrix for {} model is'.format(models_names[best_model]))

sns.heatmap(confusion_matrix(ytrain,ytrain_predict_rf),annot=True,fmt='d',cbar=False)

plt.title('Confusion Matrix for {}'.format(models_names[best_model]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = best_model.predict_proba (xtrain)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytrain, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytrain, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for RandomForestClassifier model is
 0.7638095238095238


ROC AUC score for RandomForestClassifier model is
 0.7188722111358546


Classification report for RandomForestClassifier model is
               precision    recall  f1-score   support

           0       0.83      0.83      0.83      1461
           1       0.61      0.60      0.61       639

    accuracy                           0.76      2100
   macro avg       0.72      0.72      0.72      2100
weighted avg       0.76      0.76      0.76      2100



Confusion Matrix for RandomForestClassifier model is
 [[1218  243]
 [ 253  386]]


Confusion Matrix for RandomForestClassifier model is
```



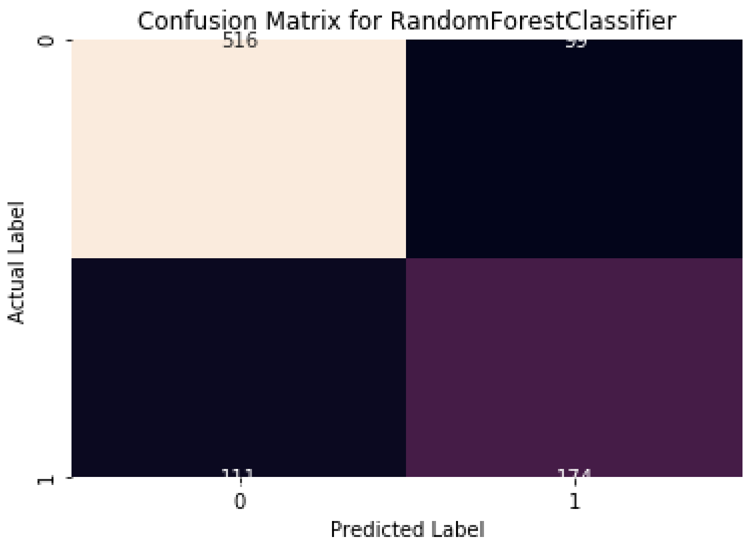

```
AUC: 0.788
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

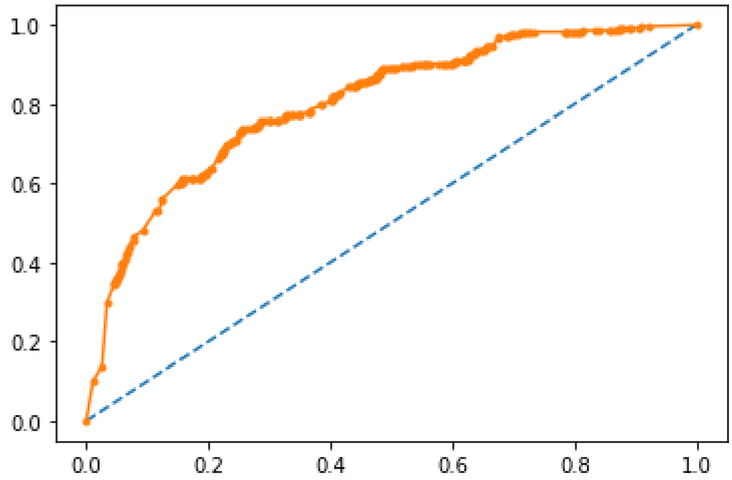
In [206]:

```
# Performance measures on Test set.

models_names={best_model:'RandomForestClassifier'}

print('Accuracy for {} model is'.format(models_names[best_model]),'\n',accuracy_score(y
test,ytest_predict_rf))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[best_model]),'\n',roc_auc_sco
re (ytest,ytest_predict_rf))

print('\n')

print('Classification report for {} model is'.format(models_names[best_model]),'\n',cla
ssification_report(ytest,ytest_predict_rf))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[best_model]),'\n',confusio
n_matrix(ytest,ytest_predict_rf))

print ('\n')


print('Confusion Matrix for {} model is'.format(models_names[best_model]))

sns.heatmap(confusion_matrix(ytest,ytest_predict_rf),annot=True,fmt='d',cbar=False)

plt.title('Confusion Matrix for {}'.format(models_names[best_model]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = best_model.predict_proba (xtest)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytest, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytest, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for RandomForestClassifier model is
 0.7666666666666667


ROC AUC score for RandomForestClassifier model is
 0.724775353016688


Classification report for RandomForestClassifier model is
               precision    recall  f1-score   support

           0       0.82      0.84      0.83       615
           1       0.64      0.61      0.62       285

    accuracy                           0.77       900
   macro avg       0.73      0.72      0.73       900
weighted avg       0.76      0.77      0.77       900



Confusion Matrix for RandomForestClassifier model is
 [[516  99]
 [111 174]]


Confusion Matrix for RandomForestClassifier model is
```




```
AUC: 0.804
```



Inferences for Random Forest (RF):

On Training Data Set Only checking for 1's. Performance Metrics on Training set has given us Accuracy of 76 %. ROC AUC score of 71 %. Recall 60 %. Precision 61 %. f1 score is also 61 %. AUC is 78 %.

False positives (FP) are 253 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1.

Only checking for 1's. Performance Metrics on Test set has given us Accuracy of 76 %. ROC AUC score of 72 %. Recall 61 %. Precision 64 %. f1 score is also 62 %. AUC is 80 %.

False positives (FP) are 111 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1.

So, both the Training and Test gives us the nearby similar score for all the Performance measures.

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

3: Artificial Neural Netwrok (ANN):

In [216]:

```
# lets fine tune the model and we will be using the features.

param_grid_ann = {'hidden_layer_sizes': [100,300],
                  'max_iter': [300,500],
                  'tol': [0.1,0.000001],
                  'solver':['adam'],
                  'verbose': ['true'],
                  }
```

In [218]:

```
mlp = MLPClassifier()
```

In [219]:

```
grid_search_ann = GridSearchCV (estimator=mlp, param_grid=param_grid_ann,cv=3)
```

In [220]:

```
grid_search_ann.fit (xtrain,ytrain)
```

```
Iteration 1, loss = 0.63686509
Iteration 2, loss = 0.59726647
Iteration 3, loss = 0.56917134
Iteration 4, loss = 0.54694730
Iteration 5, loss = 0.53020592
Iteration 6, loss = 0.51713298
Iteration 7, loss = 0.50654580
Iteration 8, loss = 0.49918397
Iteration 9, loss = 0.49308125
Iteration 10, loss = 0.48895490
Iteration 11, loss = 0.48589515
Iteration 12, loss = 0.48334300
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.59203268
Iteration 2, loss = 0.56205919
Iteration 3, loss = 0.54005918
Iteration 4, loss = 0.52318335
Iteration 5, loss = 0.51179474
Iteration 6, loss = 0.50331295
Iteration 7, loss = 0.49699078
Iteration 8, loss = 0.49427795
Iteration 9, loss = 0.49064980
Iteration 10, loss = 0.48893585
Iteration 11, loss = 0.48772182
Iteration 12, loss = 0.48646589
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.70156943
Iteration 2, loss = 0.63926737
Iteration 3, loss = 0.59468705
Iteration 4, loss = 0.56267514
Iteration 5, loss = 0.54041521
Iteration 6, loss = 0.52521953
Iteration 7, loss = 0.51549284
Iteration 8, loss = 0.50820088
Iteration 9, loss = 0.50315085
Iteration 10, loss = 0.49996212
Iteration 11, loss = 0.49725667
Iteration 12, loss = 0.49521720
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.67854489
Iteration 2, loss = 0.62701478
Iteration 3, loss = 0.58650712
Iteration 4, loss = 0.55724204
Iteration 5, loss = 0.53570137
Iteration 6, loss = 0.52021629
Iteration 7, loss = 0.50904534
Iteration 8, loss = 0.50161986
Iteration 9, loss = 0.49533131
Iteration 10, loss = 0.49112050
Iteration 11, loss = 0.48787814
Iteration 12, loss = 0.48538535
Iteration 13, loss = 0.48366964
Iteration 14, loss = 0.48199109
Iteration 15, loss = 0.48063039
Iteration 16, loss = 0.47989130
Iteration 17, loss = 0.47898473
Iteration 18, loss = 0.47802523
Iteration 19, loss = 0.47707722
```

```
Iteration 20, loss = 0.47636328
Iteration 21, loss = 0.47575798
Iteration 22, loss = 0.47511769
Iteration 23, loss = 0.47440276
Iteration 24, loss = 0.47380977
Iteration 25, loss = 0.47320956
Iteration 26, loss = 0.47266804
Iteration 27, loss = 0.47222703
Iteration 28, loss = 0.47210784
Iteration 29, loss = 0.47140196
Iteration 30, loss = 0.47069027
Iteration 31, loss = 0.47029411
Iteration 32, loss = 0.46981737
Iteration 33, loss = 0.46941133
Iteration 34, loss = 0.46880085
Iteration 35, loss = 0.46852722
Iteration 36, loss = 0.46794601
Iteration 37, loss = 0.46768628
Iteration 38, loss = 0.46719754
Iteration 39, loss = 0.46682494
Iteration 40, loss = 0.46640698
Iteration 41, loss = 0.46602547
Iteration 42, loss = 0.46562009
Iteration 43, loss = 0.46524889
Iteration 44, loss = 0.46498989
Iteration 45, loss = 0.46450098
Iteration 46, loss = 0.46410577
Iteration 47, loss = 0.46400029
Iteration 48, loss = 0.46332734
Iteration 49, loss = 0.46297624
Iteration 50, loss = 0.46262527
Iteration 51, loss = 0.46235214
Iteration 52, loss = 0.46202890
Iteration 53, loss = 0.46216945
Iteration 54, loss = 0.46139920
Iteration 55, loss = 0.46097159
Iteration 56, loss = 0.46075273
Iteration 57, loss = 0.46055736
Iteration 58, loss = 0.46018409
Iteration 59, loss = 0.45980081
Iteration 60, loss = 0.45950977
Iteration 61, loss = 0.45915627
Iteration 62, loss = 0.45892207
Iteration 63, loss = 0.45867327
Iteration 64, loss = 0.45835973
Iteration 65, loss = 0.45810283
Iteration 66, loss = 0.45807691
Iteration 67, loss = 0.45767644
Iteration 68, loss = 0.45731613
Iteration 69, loss = 0.45714833
Iteration 70, loss = 0.45705722
Iteration 71, loss = 0.45657923
Iteration 72, loss = 0.45636413
Iteration 73, loss = 0.45610540
Iteration 74, loss = 0.45592686
Iteration 75, loss = 0.45577023
Iteration 76, loss = 0.45557387
Iteration 77, loss = 0.45511549
Iteration 78, loss = 0.45496825
Iteration 79, loss = 0.45471229
Iteration 80, loss = 0.45449195
```

```
Iteration 81, loss = 0.45419821
Iteration 82, loss = 0.45400926
Iteration 83, loss = 0.45382154
Iteration 84, loss = 0.45363880
Iteration 85, loss = 0.45337200
Iteration 86, loss = 0.45333229
Iteration 87, loss = 0.45281687
Iteration 88, loss = 0.45257498
Iteration 89, loss = 0.45245518
Iteration 90, loss = 0.45252906
Iteration 91, loss = 0.45216347
Iteration 92, loss = 0.45170239
Iteration 93, loss = 0.45193196
Iteration 94, loss = 0.45151456
Iteration 95, loss = 0.45139306
Iteration 96, loss = 0.45106298
Iteration 97, loss = 0.45091992
Iteration 98, loss = 0.45059357
Iteration 99, loss = 0.45052968
Iteration 100, loss = 0.45029285
Iteration 101, loss = 0.45008778
Iteration 102, loss = 0.44980588
Iteration 103, loss = 0.44982298
Iteration 104, loss = 0.44969536
Iteration 105, loss = 0.44918278
Iteration 106, loss = 0.44939597
Iteration 107, loss = 0.44922101
Iteration 108, loss = 0.44883634
Iteration 109, loss = 0.44906384
Iteration 110, loss = 0.44858612
Iteration 111, loss = 0.44831510
Iteration 112, loss = 0.44798004
Iteration 113, loss = 0.44778972
Iteration 114, loss = 0.44765207
Iteration 115, loss = 0.44748874
Iteration 116, loss = 0.44739030
Iteration 117, loss = 0.44729021
Iteration 118, loss = 0.44698140
Iteration 119, loss = 0.44701599
Iteration 120, loss = 0.44652527
Iteration 121, loss = 0.44671392
Iteration 122, loss = 0.44636331
Iteration 123, loss = 0.44599348
Iteration 124, loss = 0.44605089
Iteration 125, loss = 0.44592342
Iteration 126, loss = 0.44584924
Iteration 127, loss = 0.44562694
Iteration 128, loss = 0.44511200
Iteration 129, loss = 0.44515231
Iteration 130, loss = 0.44484733
Iteration 131, loss = 0.44464741
Iteration 132, loss = 0.44462423
Iteration 133, loss = 0.44449820
Iteration 134, loss = 0.44442346
Iteration 135, loss = 0.44427190
Iteration 136, loss = 0.44410934
Iteration 137, loss = 0.44374957
Iteration 138, loss = 0.44435489
Iteration 139, loss = 0.44390008
Iteration 140, loss = 0.44341899
Iteration 141, loss = 0.44323345
```

```
Iteration 142, loss = 0.44320294
Iteration 143, loss = 0.44289376
Iteration 144, loss = 0.44270575
Iteration 145, loss = 0.44283465
Iteration 146, loss = 0.44242554
Iteration 147, loss = 0.44260893
Iteration 148, loss = 0.44216498
Iteration 149, loss = 0.44218591
Iteration 150, loss = 0.44194323
Iteration 151, loss = 0.44180124
Iteration 152, loss = 0.44188544
Iteration 153, loss = 0.44174103
Iteration 154, loss = 0.44151229
Iteration 155, loss = 0.44140340
Iteration 156, loss = 0.44121495
Iteration 157, loss = 0.44135408
Iteration 158, loss = 0.44083771
Iteration 159, loss = 0.44064083
Iteration 160, loss = 0.44064191
Iteration 161, loss = 0.44056548
Iteration 162, loss = 0.44041060
Iteration 163, loss = 0.44042286
Iteration 164, loss = 0.44000689
Iteration 165, loss = 0.44012998
Iteration 166, loss = 0.43992402
Iteration 167, loss = 0.43965025
Iteration 168, loss = 0.43974335
Iteration 169, loss = 0.43924455
Iteration 170, loss = 0.43918028
Iteration 171, loss = 0.43920077
Iteration 172, loss = 0.43893255
Iteration 173, loss = 0.43929237
Iteration 174, loss = 0.43880692
Iteration 175, loss = 0.43922180
Iteration 176, loss = 0.43873781
Iteration 177, loss = 0.43846588
Iteration 178, loss = 0.43865610
Iteration 179, loss = 0.43824939
Iteration 180, loss = 0.43874923
Iteration 181, loss = 0.43787399
Iteration 182, loss = 0.43777549
Iteration 183, loss = 0.43773143
Iteration 184, loss = 0.43760142
Iteration 185, loss = 0.43756634
Iteration 186, loss = 0.43701453
Iteration 187, loss = 0.43732219
Iteration 188, loss = 0.43718408
Iteration 189, loss = 0.43697909
Iteration 190, loss = 0.43720928
Iteration 191, loss = 0.43682970
Iteration 192, loss = 0.43672864
Iteration 193, loss = 0.43651536
Iteration 194, loss = 0.43629551
Iteration 195, loss = 0.43649173
Iteration 196, loss = 0.43627944
Iteration 197, loss = 0.43594578
Iteration 198, loss = 0.43623972
Iteration 199, loss = 0.43567398
Iteration 200, loss = 0.43587708
Iteration 201, loss = 0.43631900
Iteration 202, loss = 0.43531593
```

```
Iteration 203, loss = 0.43609359
Iteration 204, loss = 0.43528116
Iteration 205, loss = 0.43493643
Iteration 206, loss = 0.43481160
Iteration 207, loss = 0.43497402
Iteration 208, loss = 0.43482647
Iteration 209, loss = 0.43474894
Iteration 210, loss = 0.43455336
Iteration 211, loss = 0.43447859
Iteration 212, loss = 0.43454834
Iteration 213, loss = 0.43418905
Iteration 214, loss = 0.43397030
Iteration 215, loss = 0.43390157
Iteration 216, loss = 0.43435006
Iteration 217, loss = 0.43371780
Iteration 218, loss = 0.43398959
Iteration 219, loss = 0.43344134
Iteration 220, loss = 0.43347725
Iteration 221, loss = 0.43344848
Iteration 222, loss = 0.43328105
Iteration 223, loss = 0.43358445
Iteration 224, loss = 0.43298356
Iteration 225, loss = 0.43276298
Iteration 226, loss = 0.43307095
Iteration 227, loss = 0.43261422
Iteration 228, loss = 0.43271350
Iteration 229, loss = 0.43253106
Iteration 230, loss = 0.43232533
Iteration 231, loss = 0.43223724
Iteration 232, loss = 0.43215828
Iteration 233, loss = 0.43199041
Iteration 234, loss = 0.43191735
Iteration 235, loss = 0.43240252
Iteration 236, loss = 0.43208964
Iteration 237, loss = 0.43172107
Iteration 238, loss = 0.43164627
Iteration 239, loss = 0.43209542
Iteration 240, loss = 0.43139971
Iteration 241, loss = 0.43140480
Iteration 242, loss = 0.43130476
Iteration 243, loss = 0.43107798
Iteration 244, loss = 0.43083892
Iteration 245, loss = 0.43087615
Iteration 246, loss = 0.43100840
Iteration 247, loss = 0.43129701
Iteration 248, loss = 0.43049313
Iteration 249, loss = 0.43044566
Iteration 250, loss = 0.43031749
Iteration 251, loss = 0.43088406
Iteration 252, loss = 0.42986987
Iteration 253, loss = 0.43045109
Iteration 254, loss = 0.43012349
Iteration 255, loss = 0.43000703
Iteration 256, loss = 0.42988786
Iteration 257, loss = 0.42957218
Iteration 258, loss = 0.43010387
Iteration 259, loss = 0.42949043
Iteration 260, loss = 0.42955488
Iteration 261, loss = 0.42938047
Iteration 262, loss = 0.42935741
Iteration 263, loss = 0.42958660
```

```
Iteration 264, loss = 0.42941466
Iteration 265, loss = 0.42886153
Iteration 266, loss = 0.42872626
Iteration 267, loss = 0.42908873
Iteration 268, loss = 0.42871010
Iteration 269, loss = 0.42879865
Iteration 270, loss = 0.42848519
Iteration 271, loss = 0.42841871
Iteration 272, loss = 0.42832140
Iteration 273, loss = 0.42827439
Iteration 274, loss = 0.42829113
Iteration 275, loss = 0.42795791
Iteration 276, loss = 0.42820596
Iteration 277, loss = 0.42768385
Iteration 278, loss = 0.42761485
Iteration 279, loss = 0.42803133
Iteration 280, loss = 0.42841751
Iteration 281, loss = 0.42780276
Iteration 282, loss = 0.42755037
Iteration 283, loss = 0.42762090
Iteration 284, loss = 0.42740873
Iteration 285, loss = 0.42716759
Iteration 286, loss = 0.42733527
Iteration 287, loss = 0.42708719
Iteration 288, loss = 0.42707266
Iteration 289, loss = 0.42706501
Iteration 290, loss = 0.42671980
Iteration 291, loss = 0.42690703
Iteration 292, loss = 0.42655664
Iteration 293, loss = 0.42646960
Iteration 294, loss = 0.42656753
Iteration 295, loss = 0.42661623
Iteration 296, loss = 0.42613590
Iteration 297, loss = 0.42636332
Iteration 298, loss = 0.42593731
Iteration 299, loss = 0.42583370
Iteration 300, loss = 0.42602710
Iteration 1, loss = 0.75052963
Iteration 2, loss = 0.67763800
Iteration 3, loss = 0.61992390
Iteration 4, loss = 0.57450800
Iteration 5, loss = 0.54553775
Iteration 6, loss = 0.52660072

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 7, loss = 0.51603812
Iteration 8, loss = 0.50910174
Iteration 9, loss = 0.50450672
Iteration 10, loss = 0.50092921
Iteration 11, loss = 0.49860148
Iteration 12, loss = 0.49655460
Iteration 13, loss = 0.49502805
Iteration 14, loss = 0.49324995
Iteration 15, loss = 0.49208482
Iteration 16, loss = 0.49070838
Iteration 17, loss = 0.48954738
Iteration 18, loss = 0.48862372
Iteration 19, loss = 0.48760396
Iteration 20, loss = 0.48666774
Iteration 21, loss = 0.48564139
Iteration 22, loss = 0.48483559
Iteration 23, loss = 0.48396246
Iteration 24, loss = 0.48318352
Iteration 25, loss = 0.48262798
Iteration 26, loss = 0.48178112
Iteration 27, loss = 0.48126355
Iteration 28, loss = 0.48051173
Iteration 29, loss = 0.48008689
Iteration 30, loss = 0.47939172
Iteration 31, loss = 0.47871020
Iteration 32, loss = 0.47823724
Iteration 33, loss = 0.47755876
Iteration 34, loss = 0.47713180
Iteration 35, loss = 0.47680783
Iteration 36, loss = 0.47617094
Iteration 37, loss = 0.47562997
Iteration 38, loss = 0.47508905
Iteration 39, loss = 0.47469024
Iteration 40, loss = 0.47423185
Iteration 41, loss = 0.47378240
Iteration 42, loss = 0.47344999
Iteration 43, loss = 0.47297316
Iteration 44, loss = 0.47266104
Iteration 45, loss = 0.47204554
Iteration 46, loss = 0.47181506
Iteration 47, loss = 0.47164969
Iteration 48, loss = 0.47110015
Iteration 49, loss = 0.47037950
Iteration 50, loss = 0.47006199
Iteration 51, loss = 0.46976008
Iteration 52, loss = 0.46933457
Iteration 53, loss = 0.46895835
Iteration 54, loss = 0.46874948
Iteration 55, loss = 0.46820910
Iteration 56, loss = 0.46778215
Iteration 57, loss = 0.46741947
Iteration 58, loss = 0.46721878
Iteration 59, loss = 0.46658010
Iteration 60, loss = 0.46636896
Iteration 61, loss = 0.46634957
Iteration 62, loss = 0.46555884
Iteration 63, loss = 0.46584629
Iteration 64, loss = 0.46527377
Iteration 65, loss = 0.46493059
Iteration 66, loss = 0.46426397
Iteration 67, loss = 0.46414739
```

```
Iteration 68, loss = 0.46384025
Iteration 69, loss = 0.46375683
Iteration 70, loss = 0.46323627
Iteration 71, loss = 0.46289947
Iteration 72, loss = 0.46279674
Iteration 73, loss = 0.46230407
Iteration 74, loss = 0.46215653
Iteration 75, loss = 0.46165599
Iteration 76, loss = 0.46126111
Iteration 77, loss = 0.46139407
Iteration 78, loss = 0.46065728
Iteration 79, loss = 0.46047704
Iteration 80, loss = 0.46030924
Iteration 81, loss = 0.46016018
Iteration 82, loss = 0.45961748
Iteration 83, loss = 0.45943408
Iteration 84, loss = 0.45909761
Iteration 85, loss = 0.45863627
Iteration 86, loss = 0.45840677
Iteration 87, loss = 0.45802776
Iteration 88, loss = 0.45781196
Iteration 89, loss = 0.45745843
Iteration 90, loss = 0.45727499
Iteration 91, loss = 0.45695867
Iteration 92, loss = 0.45684692
Iteration 93, loss = 0.45645684
Iteration 94, loss = 0.45625530
Iteration 95, loss = 0.45636695
Iteration 96, loss = 0.45563777
Iteration 97, loss = 0.45549290
Iteration 98, loss = 0.45525828
Iteration 99, loss = 0.45505496
Iteration 100, loss = 0.45484673
Iteration 101, loss = 0.45457591
Iteration 102, loss = 0.45430973
Iteration 103, loss = 0.45381821
Iteration 104, loss = 0.45403850
Iteration 105, loss = 0.45379180
Iteration 106, loss = 0.45335186
Iteration 107, loss = 0.45342942
Iteration 108, loss = 0.45285182
Iteration 109, loss = 0.45264586
Iteration 110, loss = 0.45236510
Iteration 111, loss = 0.45225368
Iteration 112, loss = 0.45217298
Iteration 113, loss = 0.45163570
Iteration 114, loss = 0.45153091
Iteration 115, loss = 0.45106019
Iteration 116, loss = 0.45085404
Iteration 117, loss = 0.45076890
Iteration 118, loss = 0.45058592
Iteration 119, loss = 0.45030759
Iteration 120, loss = 0.45042577
Iteration 121, loss = 0.44983365
Iteration 122, loss = 0.44970109
Iteration 123, loss = 0.44943103
Iteration 124, loss = 0.44940253
Iteration 125, loss = 0.44884338
Iteration 126, loss = 0.44892394
Iteration 127, loss = 0.44850491
Iteration 128, loss = 0.44831412
```

```
Iteration 129, loss = 0.44833514
Iteration 130, loss = 0.44769058
Iteration 131, loss = 0.44796773
Iteration 132, loss = 0.44763060
Iteration 133, loss = 0.44730978
Iteration 134, loss = 0.44705325
Iteration 135, loss = 0.44704812
Iteration 136, loss = 0.44667941
Iteration 137, loss = 0.44660591
Iteration 138, loss = 0.44634462
Iteration 139, loss = 0.44632832
Iteration 140, loss = 0.44591687
Iteration 141, loss = 0.44574148
Iteration 142, loss = 0.44569713
Iteration 143, loss = 0.44550009
Iteration 144, loss = 0.44530354
Iteration 145, loss = 0.44539667
Iteration 146, loss = 0.44535864
Iteration 147, loss = 0.44471162
Iteration 148, loss = 0.44463656
Iteration 149, loss = 0.44427589
Iteration 150, loss = 0.44411165
Iteration 151, loss = 0.44408914
Iteration 152, loss = 0.44380527
Iteration 153, loss = 0.44362129
Iteration 154, loss = 0.44341689
Iteration 155, loss = 0.44334914
Iteration 156, loss = 0.44306380
Iteration 157, loss = 0.44305300
Iteration 158, loss = 0.44300993
Iteration 159, loss = 0.44257232
Iteration 160, loss = 0.44261325
Iteration 161, loss = 0.44237633
Iteration 162, loss = 0.44215868
Iteration 163, loss = 0.44201495
Iteration 164, loss = 0.44171925
Iteration 165, loss = 0.44151078
Iteration 166, loss = 0.44161040
Iteration 167, loss = 0.44150046
Iteration 168, loss = 0.44118958
Iteration 169, loss = 0.44093910
Iteration 170, loss = 0.44111566
Iteration 171, loss = 0.44074225
Iteration 172, loss = 0.44049823
Iteration 173, loss = 0.44049856
Iteration 174, loss = 0.44014116
Iteration 175, loss = 0.44048504
Iteration 176, loss = 0.44014405
Iteration 177, loss = 0.43973066
Iteration 178, loss = 0.43990501
Iteration 179, loss = 0.43936366
Iteration 180, loss = 0.43948181
Iteration 181, loss = 0.43908215
Iteration 182, loss = 0.43903293
Iteration 183, loss = 0.43891901
Iteration 184, loss = 0.43922032
Iteration 185, loss = 0.43844938
Iteration 186, loss = 0.43857236
Iteration 187, loss = 0.43822607
Iteration 188, loss = 0.43803058
Iteration 189, loss = 0.43774921
```

```
Iteration 190, loss = 0.43815280
Iteration 191, loss = 0.43759848
Iteration 192, loss = 0.43780568
Iteration 193, loss = 0.43778333
Iteration 194, loss = 0.43725554
Iteration 195, loss = 0.43716289
Iteration 196, loss = 0.43715935
Iteration 197, loss = 0.43663883
Iteration 198, loss = 0.43737365
Iteration 199, loss = 0.43689216
Iteration 200, loss = 0.43619383
Iteration 201, loss = 0.43651312
Iteration 202, loss = 0.43615099
Iteration 203, loss = 0.43652986
Iteration 204, loss = 0.43577136
Iteration 205, loss = 0.43615943
Iteration 206, loss = 0.43533269
Iteration 207, loss = 0.43544209
Iteration 208, loss = 0.43544134
Iteration 209, loss = 0.43536656
Iteration 210, loss = 0.43509118
Iteration 211, loss = 0.43524728
Iteration 212, loss = 0.43474650
Iteration 213, loss = 0.43490496
Iteration 214, loss = 0.43485316
Iteration 215, loss = 0.43417171
Iteration 216, loss = 0.43436307
Iteration 217, loss = 0.43409255
Iteration 218, loss = 0.43386712
Iteration 219, loss = 0.43362491
Iteration 220, loss = 0.43351711
Iteration 221, loss = 0.43403566
Iteration 222, loss = 0.43337610
Iteration 223, loss = 0.43364089
Iteration 224, loss = 0.43399718
Iteration 225, loss = 0.43358323
Iteration 226, loss = 0.43303699
Iteration 227, loss = 0.43273267
Iteration 228, loss = 0.43305307
Iteration 229, loss = 0.43277518
Iteration 230, loss = 0.43320906
Iteration 231, loss = 0.43257725
Iteration 232, loss = 0.43227598
Iteration 233, loss = 0.43294038
Iteration 234, loss = 0.43214893
Iteration 235, loss = 0.43234490
Iteration 236, loss = 0.43193759
Iteration 237, loss = 0.43165013
Iteration 238, loss = 0.43168878
Iteration 239, loss = 0.43131156
Iteration 240, loss = 0.43146709
Iteration 241, loss = 0.43236382
Iteration 242, loss = 0.43144020
Iteration 243, loss = 0.43159449
Iteration 244, loss = 0.43124991
Iteration 245, loss = 0.43065951
Iteration 246, loss = 0.43065774
Iteration 247, loss = 0.43079621
Iteration 248, loss = 0.43046837
Iteration 249, loss = 0.43029399
Iteration 250, loss = 0.43011481
```

```
Iteration 251, loss = 0.42995149
Iteration 252, loss = 0.42998795
Iteration 253, loss = 0.43001869
Iteration 254, loss = 0.42990800
Iteration 255, loss = 0.42953727
Iteration 256, loss = 0.43010980
Iteration 257, loss = 0.42992599
Iteration 258, loss = 0.42956468
Iteration 259, loss = 0.42960354
Iteration 260, loss = 0.42934003
Iteration 261, loss = 0.42887875
Iteration 262, loss = 0.42926062
Iteration 263, loss = 0.42865279
Iteration 264, loss = 0.42944879
Iteration 265, loss = 0.42889817
Iteration 266, loss = 0.42864497
Iteration 267, loss = 0.42868963
Iteration 268, loss = 0.42860850
Iteration 269, loss = 0.42813330
Iteration 270, loss = 0.42792817
Iteration 271, loss = 0.42774368
Iteration 272, loss = 0.42787816
Iteration 273, loss = 0.42768026
Iteration 274, loss = 0.42753048
Iteration 275, loss = 0.42780068
Iteration 276, loss = 0.42739172
Iteration 277, loss = 0.42753743
Iteration 278, loss = 0.42725283
Iteration 279, loss = 0.42718154
Iteration 280, loss = 0.42682928
Iteration 281, loss = 0.42690421
Iteration 282, loss = 0.42665271
Iteration 283, loss = 0.42669260
Iteration 284, loss = 0.42702765
Iteration 285, loss = 0.42665586
Iteration 286, loss = 0.42694447
Iteration 287, loss = 0.42634216
Iteration 288, loss = 0.42610790
Iteration 289, loss = 0.42610277
Iteration 290, loss = 0.42578014
Iteration 291, loss = 0.42657057
Iteration 292, loss = 0.42621735
Iteration 293, loss = 0.42560184
Iteration 294, loss = 0.42552363
Iteration 295, loss = 0.42528215
Iteration 296, loss = 0.42514596
Iteration 297, loss = 0.42541784
Iteration 298, loss = 0.42546575
Iteration 299, loss = 0.42499267
Iteration 300, loss = 0.42502431
Iteration 1, loss = 0.71280332
Iteration 2, loss = 0.65487656
Iteration 3, loss = 0.61106746
Iteration 4, loss = 0.57726980
Iteration 5, loss = 0.55256159
Iteration 6, loss = 0.53406725
Iteration 7, loss = 0.52175711
Iteration 8, loss = 0.51266949
Iteration 9, loss = 0.50574097
Iteration 10, loss = 0.50158208
Iteration 11, loss = 0.49838364
```

```
Iteration 12, loss = 0.49569733
Iteration 13, loss = 0.49405283
Iteration 14, loss = 0.49276079
Iteration 15, loss = 0.49142382
Iteration 16, loss = 0.49041736
Iteration 17, loss = 0.48938170
Iteration 18, loss = 0.48870416
Iteration 19, loss = 0.48790293

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 20, loss = 0.48708469
Iteration 21, loss = 0.48614826
Iteration 22, loss = 0.48569421
Iteration 23, loss = 0.48492017
Iteration 24, loss = 0.48441124
Iteration 25, loss = 0.48378680
Iteration 26, loss = 0.48322575
Iteration 27, loss = 0.48259093
Iteration 28, loss = 0.48209264
Iteration 29, loss = 0.48148069
Iteration 30, loss = 0.48087570
Iteration 31, loss = 0.48033593
Iteration 32, loss = 0.47992852
Iteration 33, loss = 0.47943743
Iteration 34, loss = 0.47902764
Iteration 35, loss = 0.47853220
Iteration 36, loss = 0.47797694
Iteration 37, loss = 0.47754277
Iteration 38, loss = 0.47702377
Iteration 39, loss = 0.47666315
Iteration 40, loss = 0.47653105
Iteration 41, loss = 0.47586137
Iteration 42, loss = 0.47550866
Iteration 43, loss = 0.47492271
Iteration 44, loss = 0.47460361
Iteration 45, loss = 0.47410635
Iteration 46, loss = 0.47380635
Iteration 47, loss = 0.47348007
Iteration 48, loss = 0.47330545
Iteration 49, loss = 0.47272905
Iteration 50, loss = 0.47229252
Iteration 51, loss = 0.47180768
Iteration 52, loss = 0.47147716
Iteration 53, loss = 0.47119741
Iteration 54, loss = 0.47089448
Iteration 55, loss = 0.47067846
Iteration 56, loss = 0.47026234
Iteration 57, loss = 0.47018697
Iteration 58, loss = 0.46957207
Iteration 59, loss = 0.46918884
Iteration 60, loss = 0.46890464
Iteration 61, loss = 0.46851505
Iteration 62, loss = 0.46812895
Iteration 63, loss = 0.46786543
Iteration 64, loss = 0.46773795
Iteration 65, loss = 0.46723277
Iteration 66, loss = 0.46695250
Iteration 67, loss = 0.46665194
Iteration 68, loss = 0.46642041
Iteration 69, loss = 0.46588138
Iteration 70, loss = 0.46569692
Iteration 71, loss = 0.46529868
Iteration 72, loss = 0.46533235
Iteration 73, loss = 0.46492983
Iteration 74, loss = 0.46458767
Iteration 75, loss = 0.46438827
Iteration 76, loss = 0.46393369
Iteration 77, loss = 0.46354942
Iteration 78, loss = 0.46352355
Iteration 79, loss = 0.46307419
Iteration 80, loss = 0.46307535
```

```
Iteration 81, loss = 0.46257671
Iteration 82, loss = 0.46250248
Iteration 83, loss = 0.46214211
Iteration 84, loss = 0.46181943
Iteration 85, loss = 0.46135146
Iteration 86, loss = 0.46134246
Iteration 87, loss = 0.46107599
Iteration 88, loss = 0.46057532
Iteration 89, loss = 0.46065943
Iteration 90, loss = 0.46044155
Iteration 91, loss = 0.46017326
Iteration 92, loss = 0.46003274
Iteration 93, loss = 0.45956299
Iteration 94, loss = 0.45953240
Iteration 95, loss = 0.45905892
Iteration 96, loss = 0.45883161
Iteration 97, loss = 0.45862844
Iteration 98, loss = 0.45867544
Iteration 99, loss = 0.45825353
Iteration 100, loss = 0.45819967
Iteration 101, loss = 0.45776939
Iteration 102, loss = 0.45742407
Iteration 103, loss = 0.45744632
Iteration 104, loss = 0.45708461
Iteration 105, loss = 0.45690380
Iteration 106, loss = 0.45660923
Iteration 107, loss = 0.45653451
Iteration 108, loss = 0.45624285
Iteration 109, loss = 0.45596058
Iteration 110, loss = 0.45594244
Iteration 111, loss = 0.45542374
Iteration 112, loss = 0.45549656
Iteration 113, loss = 0.45553078
Iteration 114, loss = 0.45522455
Iteration 115, loss = 0.45464997
Iteration 116, loss = 0.45451132
Iteration 117, loss = 0.45439180
Iteration 118, loss = 0.45402751
Iteration 119, loss = 0.45392407
Iteration 120, loss = 0.45383717
Iteration 121, loss = 0.45353706
Iteration 122, loss = 0.45357941
Iteration 123, loss = 0.45320096
Iteration 124, loss = 0.45295035
Iteration 125, loss = 0.45280757
Iteration 126, loss = 0.45250100
Iteration 127, loss = 0.45253739
Iteration 128, loss = 0.45240534
Iteration 129, loss = 0.45220412
Iteration 130, loss = 0.45211805
Iteration 131, loss = 0.45178470
Iteration 132, loss = 0.45181528
Iteration 133, loss = 0.45123728
Iteration 134, loss = 0.45124081
Iteration 135, loss = 0.45102125
Iteration 136, loss = 0.45119983
Iteration 137, loss = 0.45069493
Iteration 138, loss = 0.45142586
Iteration 139, loss = 0.45052398
Iteration 140, loss = 0.45010815
Iteration 141, loss = 0.44993322
```

```
Iteration 142, loss = 0.44991576
Iteration 143, loss = 0.44971006
Iteration 144, loss = 0.44967365
Iteration 145, loss = 0.44934685
Iteration 146, loss = 0.44958539
Iteration 147, loss = 0.44912694
Iteration 148, loss = 0.44865516
Iteration 149, loss = 0.44889177
Iteration 150, loss = 0.44859757
Iteration 151, loss = 0.44835128
Iteration 152, loss = 0.44832096
Iteration 153, loss = 0.44788960
Iteration 154, loss = 0.44763713
Iteration 155, loss = 0.44762350
Iteration 156, loss = 0.44757990
Iteration 157, loss = 0.44717519
Iteration 158, loss = 0.44685002
Iteration 159, loss = 0.44708894
Iteration 160, loss = 0.44658853
Iteration 161, loss = 0.44654018
Iteration 162, loss = 0.44665477
Iteration 163, loss = 0.44627364
Iteration 164, loss = 0.44626284
Iteration 165, loss = 0.44612524
Iteration 166, loss = 0.44584676
Iteration 167, loss = 0.44584338
Iteration 168, loss = 0.44551981
Iteration 169, loss = 0.44539778
Iteration 170, loss = 0.44488394
Iteration 171, loss = 0.44549024
Iteration 172, loss = 0.44560879
Iteration 173, loss = 0.44457545
Iteration 174, loss = 0.44451378
Iteration 175, loss = 0.44425938
Iteration 176, loss = 0.44411399
Iteration 177, loss = 0.44411881
Iteration 178, loss = 0.44425934
Iteration 179, loss = 0.44371911
Iteration 180, loss = 0.44362309
Iteration 181, loss = 0.44344817
Iteration 182, loss = 0.44353260
Iteration 183, loss = 0.44335224
Iteration 184, loss = 0.44302148
Iteration 185, loss = 0.44296923
Iteration 186, loss = 0.44278946
Iteration 187, loss = 0.44249241
Iteration 188, loss = 0.44236003
Iteration 189, loss = 0.44237120
Iteration 190, loss = 0.44246057
Iteration 191, loss = 0.44231222
Iteration 192, loss = 0.44188028
Iteration 193, loss = 0.44273870
Iteration 194, loss = 0.44139408
Iteration 195, loss = 0.44151732
Iteration 196, loss = 0.44168699
Iteration 197, loss = 0.44108605
Iteration 198, loss = 0.44087809
Iteration 199, loss = 0.44122445
Iteration 200, loss = 0.44071666
Iteration 201, loss = 0.44062059
Iteration 202, loss = 0.44036485
```

```
Iteration 203, loss = 0.44029989
Iteration 204, loss = 0.44011550
Iteration 205, loss = 0.44015819
Iteration 206, loss = 0.44014644
Iteration 207, loss = 0.43993945
Iteration 208, loss = 0.43958104
Iteration 209, loss = 0.43977078
Iteration 210, loss = 0.43942153
Iteration 211, loss = 0.43895548
Iteration 212, loss = 0.43888008
Iteration 213, loss = 0.43891162
Iteration 214, loss = 0.43879227
Iteration 215, loss = 0.43853808
Iteration 216, loss = 0.43844632
Iteration 217, loss = 0.43824957
Iteration 218, loss = 0.43817354
Iteration 219, loss = 0.43791089
Iteration 220, loss = 0.43796245
Iteration 221, loss = 0.43790625
Iteration 222, loss = 0.43803805
Iteration 223, loss = 0.43750471
Iteration 224, loss = 0.43732299
Iteration 225, loss = 0.43739246
Iteration 226, loss = 0.43723406
Iteration 227, loss = 0.43705511
Iteration 228, loss = 0.43690906
Iteration 229, loss = 0.43713381
Iteration 230, loss = 0.43655875
Iteration 231, loss = 0.43670971
Iteration 232, loss = 0.43644058
Iteration 233, loss = 0.43633497
Iteration 234, loss = 0.43602398
Iteration 235, loss = 0.43615711
Iteration 236, loss = 0.43588764
Iteration 237, loss = 0.43601179
Iteration 238, loss = 0.43547158
Iteration 239, loss = 0.43640901
Iteration 240, loss = 0.43590719
Iteration 241, loss = 0.43592131
Iteration 242, loss = 0.43542275
Iteration 243, loss = 0.43549670
Iteration 244, loss = 0.43495076
Iteration 245, loss = 0.43487649
Iteration 246, loss = 0.43505650
Iteration 247, loss = 0.43453389
Iteration 248, loss = 0.43463166
Iteration 249, loss = 0.43456822
Iteration 250, loss = 0.43425320
Iteration 251, loss = 0.43415288
Iteration 252, loss = 0.43409408
Iteration 253, loss = 0.43437025
Iteration 254, loss = 0.43415329
Iteration 255, loss = 0.43363621
Iteration 256, loss = 0.43380772
Iteration 257, loss = 0.43382504
Iteration 258, loss = 0.43354262
Iteration 259, loss = 0.43363426
Iteration 260, loss = 0.43317585
Iteration 261, loss = 0.43327747
Iteration 262, loss = 0.43309868
Iteration 263, loss = 0.43280290
```

```
Iteration 264, loss = 0.43265060
Iteration 265, loss = 0.43265780
Iteration 266, loss = 0.43255243
Iteration 267, loss = 0.43215839
Iteration 268, loss = 0.43245790
Iteration 269, loss = 0.43193174
Iteration 270, loss = 0.43178598
Iteration 271, loss = 0.43246498
Iteration 272, loss = 0.43172387
Iteration 273, loss = 0.43174062
Iteration 274, loss = 0.43123201
Iteration 275, loss = 0.43157604
Iteration 276, loss = 0.43143691
Iteration 277, loss = 0.43124381
Iteration 278, loss = 0.43100688
Iteration 279, loss = 0.43103795
Iteration 280, loss = 0.43082843
Iteration 281, loss = 0.43052749
Iteration 282, loss = 0.43047358
Iteration 283, loss = 0.43061694
Iteration 284, loss = 0.43054572
Iteration 285, loss = 0.42971066
Iteration 286, loss = 0.43070829
Iteration 287, loss = 0.43012744
Iteration 288, loss = 0.42966070
Iteration 289, loss = 0.43028912
Iteration 290, loss = 0.42933191
Iteration 291, loss = 0.42974691
Iteration 292, loss = 0.42980095
Iteration 293, loss = 0.42899070
Iteration 294, loss = 0.42919229
Iteration 295, loss = 0.42933338
Iteration 296, loss = 0.42875458
Iteration 297, loss = 0.42863464
Iteration 298, loss = 0.42883907
Iteration 299, loss = 0.42839322
Iteration 300, loss = 0.42861709
Iteration 1, loss = 0.70934017
Iteration 2, loss = 0.65563811
Iteration 3, loss = 0.61431013
Iteration 4, loss = 0.58146967

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 5, loss = 0.55653102
Iteration 6, loss = 0.53668810
Iteration 7, loss = 0.52133940
Iteration 8, loss = 0.51010408
Iteration 9, loss = 0.50097433
Iteration 10, loss = 0.49478146
Iteration 11, loss = 0.49037974
Iteration 12, loss = 0.48653796
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.65153072
Iteration 2, loss = 0.60331339
Iteration 3, loss = 0.56956820
Iteration 4, loss = 0.54364693
Iteration 5, loss = 0.52636732
Iteration 6, loss = 0.51478294
Iteration 7, loss = 0.50690247
Iteration 8, loss = 0.50161272
Iteration 9, loss = 0.49807954
Iteration 10, loss = 0.49620165
Iteration 11, loss = 0.49428552
Iteration 12, loss = 0.49286163
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.66078570
Iteration 2, loss = 0.60784720
Iteration 3, loss = 0.57028367
Iteration 4, loss = 0.54584073
Iteration 5, loss = 0.52838990
Iteration 6, loss = 0.51612417
Iteration 7, loss = 0.50829183
Iteration 8, loss = 0.50246854
Iteration 9, loss = 0.49842353
Iteration 10, loss = 0.49522495
Iteration 11, loss = 0.49328603
Iteration 12, loss = 0.49148059
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.68368246
Iteration 2, loss = 0.62559529
Iteration 3, loss = 0.57938351
Iteration 4, loss = 0.54668647
Iteration 5, loss = 0.52441220
Iteration 6, loss = 0.50984131
Iteration 7, loss = 0.49971664
Iteration 8, loss = 0.49308582
Iteration 9, loss = 0.48925834
Iteration 10, loss = 0.48633258
Iteration 11, loss = 0.48429615
Iteration 12, loss = 0.48280440
Iteration 13, loss = 0.48136777
Iteration 14, loss = 0.48036651
Iteration 15, loss = 0.47945134
Iteration 16, loss = 0.47840417
Iteration 17, loss = 0.47753286
Iteration 18, loss = 0.47687040
Iteration 19, loss = 0.47606223
Iteration 20, loss = 0.47546756
Iteration 21, loss = 0.47472801
Iteration 22, loss = 0.47418027
Iteration 23, loss = 0.47362286
```

```
Iteration 24, loss = 0.47278600
Iteration 25, loss = 0.47236664
Iteration 26, loss = 0.47162040
Iteration 27, loss = 0.47122806
Iteration 28, loss = 0.47053878
Iteration 29, loss = 0.47018677
Iteration 30, loss = 0.46948645
Iteration 31, loss = 0.46904310
Iteration 32, loss = 0.46860661
Iteration 33, loss = 0.46815587
Iteration 34, loss = 0.46796911
Iteration 35, loss = 0.46731829
Iteration 36, loss = 0.46692663
Iteration 37, loss = 0.46651799
Iteration 38, loss = 0.46585863
Iteration 39, loss = 0.46566221
Iteration 40, loss = 0.46503570
Iteration 41, loss = 0.46482239
Iteration 42, loss = 0.46452619
Iteration 43, loss = 0.46406401
Iteration 44, loss = 0.46367231
Iteration 45, loss = 0.46332739
Iteration 46, loss = 0.46286644
Iteration 47, loss = 0.46265537
Iteration 48, loss = 0.46236026
Iteration 49, loss = 0.46192545
Iteration 50, loss = 0.46155354
Iteration 51, loss = 0.46125668
Iteration 52, loss = 0.46081399
Iteration 53, loss = 0.46049819
Iteration 54, loss = 0.46020925
Iteration 55, loss = 0.46006094
Iteration 56, loss = 0.45960209
Iteration 57, loss = 0.45961996
Iteration 58, loss = 0.45908115
Iteration 59, loss = 0.45878296
Iteration 60, loss = 0.45846988
Iteration 61, loss = 0.45807778
Iteration 62, loss = 0.45802070
Iteration 63, loss = 0.45763655
Iteration 64, loss = 0.45729173
Iteration 65, loss = 0.45707577
Iteration 66, loss = 0.45687923
Iteration 67, loss = 0.45664604
Iteration 68, loss = 0.45642418
Iteration 69, loss = 0.45599838
Iteration 70, loss = 0.45593416
Iteration 71, loss = 0.45573305
Iteration 72, loss = 0.45539118
Iteration 73, loss = 0.45506621
Iteration 74, loss = 0.45479891
Iteration 75, loss = 0.45449433
Iteration 76, loss = 0.45428445
Iteration 77, loss = 0.45404893
Iteration 78, loss = 0.45402323
Iteration 79, loss = 0.45366191
Iteration 80, loss = 0.45345905
Iteration 81, loss = 0.45318113
Iteration 82, loss = 0.45293292
Iteration 83, loss = 0.45310473
Iteration 84, loss = 0.45242245
```

```
Iteration 85, loss = 0.45221040
Iteration 86, loss = 0.45215367
Iteration 87, loss = 0.45234427
Iteration 88, loss = 0.45174623
Iteration 89, loss = 0.45152963
Iteration 90, loss = 0.45150189
Iteration 91, loss = 0.45098801
Iteration 92, loss = 0.45100312
Iteration 93, loss = 0.45105210
Iteration 94, loss = 0.45079598
Iteration 95, loss = 0.45104424
Iteration 96, loss = 0.45075709
Iteration 97, loss = 0.44999639
Iteration 98, loss = 0.44975292
Iteration 99, loss = 0.44948488
Iteration 100, loss = 0.44967891
Iteration 101, loss = 0.44917896
Iteration 102, loss = 0.44913451
Iteration 103, loss = 0.44884475
Iteration 104, loss = 0.44866948
Iteration 105, loss = 0.44850089
Iteration 106, loss = 0.44854495
Iteration 107, loss = 0.44823891
Iteration 108, loss = 0.44842024
Iteration 109, loss = 0.44776032
Iteration 110, loss = 0.44779256
Iteration 111, loss = 0.44741958
Iteration 112, loss = 0.44781424
Iteration 113, loss = 0.44744415
Iteration 114, loss = 0.44704018
Iteration 115, loss = 0.44674479
Iteration 116, loss = 0.44674743
Iteration 117, loss = 0.44635027
Iteration 118, loss = 0.44635970
Iteration 119, loss = 0.44620674
Iteration 120, loss = 0.44594639
Iteration 121, loss = 0.44556415
Iteration 122, loss = 0.44574812
Iteration 123, loss = 0.44559634
Iteration 124, loss = 0.44544866
Iteration 125, loss = 0.44530684
Iteration 126, loss = 0.44500202
Iteration 127, loss = 0.44504806
Iteration 128, loss = 0.44466337
Iteration 129, loss = 0.44446683
Iteration 130, loss = 0.44452248
Iteration 131, loss = 0.44437112
Iteration 132, loss = 0.44429556
Iteration 133, loss = 0.44406837
Iteration 134, loss = 0.44388301
Iteration 135, loss = 0.44379199
Iteration 136, loss = 0.44385841
Iteration 137, loss = 0.44335201
Iteration 138, loss = 0.44320748
Iteration 139, loss = 0.44309066
Iteration 140, loss = 0.44305288
Iteration 141, loss = 0.44287114
Iteration 142, loss = 0.44294238
Iteration 143, loss = 0.44260060
Iteration 144, loss = 0.44241481
Iteration 145, loss = 0.44292671
```

```
Iteration 146, loss = 0.44238408
Iteration 147, loss = 0.44216969
Iteration 148, loss = 0.44171995
Iteration 149, loss = 0.44177555
Iteration 150, loss = 0.44150587
Iteration 151, loss = 0.44157687
Iteration 152, loss = 0.44142962
Iteration 153, loss = 0.44112795
Iteration 154, loss = 0.44095052
Iteration 155, loss = 0.44102409
Iteration 156, loss = 0.44075330
Iteration 157, loss = 0.44056553
Iteration 158, loss = 0.44058984
Iteration 159, loss = 0.44031154
Iteration 160, loss = 0.44050194
Iteration 161, loss = 0.44000327
Iteration 162, loss = 0.44021755
Iteration 163, loss = 0.43995130
Iteration 164, loss = 0.43969403
Iteration 165, loss = 0.43947205
Iteration 166, loss = 0.43927925
Iteration 167, loss = 0.43934419
Iteration 168, loss = 0.43928803
Iteration 169, loss = 0.43919460
Iteration 170, loss = 0.43886813
Iteration 171, loss = 0.43906625
Iteration 172, loss = 0.43860254
Iteration 173, loss = 0.43858721
Iteration 174, loss = 0.43861162
Iteration 175, loss = 0.43832876
Iteration 176, loss = 0.43816588
Iteration 177, loss = 0.43855758
Iteration 178, loss = 0.43809661
Iteration 179, loss = 0.43783836
Iteration 180, loss = 0.43773453
Iteration 181, loss = 0.43759287
Iteration 182, loss = 0.43749476
Iteration 183, loss = 0.43777317
Iteration 184, loss = 0.43728205
Iteration 185, loss = 0.43712925
Iteration 186, loss = 0.43705092
Iteration 187, loss = 0.43700818
Iteration 188, loss = 0.43706347
Iteration 189, loss = 0.43671851
Iteration 190, loss = 0.43652863
Iteration 191, loss = 0.43621724
Iteration 192, loss = 0.43610185
Iteration 193, loss = 0.43625546
Iteration 194, loss = 0.43626064
Iteration 195, loss = 0.43631986
Iteration 196, loss = 0.43605386
Iteration 197, loss = 0.43633402
Iteration 198, loss = 0.43562050
Iteration 199, loss = 0.43582762
Iteration 200, loss = 0.43551202
Iteration 201, loss = 0.43544551
Iteration 202, loss = 0.43551689
Iteration 203, loss = 0.43521941
Iteration 204, loss = 0.43514236
Iteration 205, loss = 0.43499841
Iteration 206, loss = 0.43497823
```

```
Iteration 207, loss = 0.43520639
Iteration 208, loss = 0.43469310
Iteration 209, loss = 0.43452862
Iteration 210, loss = 0.43447517
Iteration 211, loss = 0.43441333
Iteration 212, loss = 0.43456234
Iteration 213, loss = 0.43400326
Iteration 214, loss = 0.43378267
Iteration 215, loss = 0.43379605
Iteration 216, loss = 0.43400805
Iteration 217, loss = 0.43352830
Iteration 218, loss = 0.43385903
Iteration 219, loss = 0.43351430
Iteration 220, loss = 0.43339359
Iteration 221, loss = 0.43328974
Iteration 222, loss = 0.43367155
Iteration 223, loss = 0.43288457
Iteration 224, loss = 0.43287813
Iteration 225, loss = 0.43313685
Iteration 226, loss = 0.43272651
Iteration 227, loss = 0.43265664
Iteration 228, loss = 0.43243414
Iteration 229, loss = 0.43253914
Iteration 230, loss = 0.43220230
Iteration 231, loss = 0.43248851
Iteration 232, loss = 0.43271879
Iteration 233, loss = 0.43219103
Iteration 234, loss = 0.43184168
Iteration 235, loss = 0.43185794
Iteration 236, loss = 0.43202377
Iteration 237, loss = 0.43170182
Iteration 238, loss = 0.43160153
Iteration 239, loss = 0.43131497
Iteration 240, loss = 0.43147480
Iteration 241, loss = 0.43111630
Iteration 242, loss = 0.43127168
Iteration 243, loss = 0.43134502
Iteration 244, loss = 0.43087402
Iteration 245, loss = 0.43094949
Iteration 246, loss = 0.43160528
Iteration 247, loss = 0.43092614
Iteration 248, loss = 0.43080419
Iteration 249, loss = 0.43056050
Iteration 250, loss = 0.43049615
Iteration 251, loss = 0.43055068
Iteration 252, loss = 0.43018708
Iteration 253, loss = 0.43004537
Iteration 254, loss = 0.43029013
Iteration 255, loss = 0.42996786
Iteration 256, loss = 0.43013541
Iteration 257, loss = 0.42960505
Iteration 258, loss = 0.42956292
Iteration 259, loss = 0.42971097
Iteration 260, loss = 0.42937336
Iteration 261, loss = 0.42919551
Iteration 262, loss = 0.42909581
Iteration 263, loss = 0.43025695
Iteration 264, loss = 0.42908755
Iteration 265, loss = 0.42922048
Iteration 266, loss = 0.42894164
Iteration 267, loss = 0.42898043
```

```
Iteration 268, loss = 0.42870496
Iteration 269, loss = 0.42856827
Iteration 270, loss = 0.42904235
Iteration 271, loss = 0.42874262
Iteration 272, loss = 0.42878272
Iteration 273, loss = 0.42847988
Iteration 274, loss = 0.42801970
Iteration 275, loss = 0.42813851
Iteration 276, loss = 0.42841355
Iteration 277, loss = 0.42831800
Iteration 278, loss = 0.42829885
Iteration 279, loss = 0.42757949
Iteration 280, loss = 0.42764286
Iteration 281, loss = 0.42782110
Iteration 282, loss = 0.42758533
Iteration 283, loss = 0.42783550
Iteration 284, loss = 0.42727413
Iteration 285, loss = 0.42731294
Iteration 286, loss = 0.42702903
Iteration 287, loss = 0.42726875
Iteration 288, loss = 0.42688660
Iteration 289, loss = 0.42667721
Iteration 290, loss = 0.42692107
Iteration 291, loss = 0.42658637
Iteration 292, loss = 0.42655697
Iteration 293, loss = 0.42663531
Iteration 294, loss = 0.42654779
Iteration 295, loss = 0.42610048
Iteration 296, loss = 0.42611906
Iteration 297, loss = 0.42646596
Iteration 298, loss = 0.42569016
Iteration 299, loss = 0.42591169
Iteration 300, loss = 0.42604728
Iteration 301, loss = 0.42582663
Iteration 302, loss = 0.42566064
Iteration 303, loss = 0.42548663
Iteration 304, loss = 0.42548625
Iteration 305, loss = 0.42548043
Iteration 306, loss = 0.42547279
Iteration 307, loss = 0.42532717
Iteration 308, loss = 0.42501838
Iteration 309, loss = 0.42509117
Iteration 310, loss = 0.42476029
Iteration 311, loss = 0.42464673
Iteration 312, loss = 0.42460575
Iteration 313, loss = 0.42481432
Iteration 314, loss = 0.42484683
Iteration 315, loss = 0.42453067
Iteration 316, loss = 0.42421232
Iteration 317, loss = 0.42432788
Iteration 318, loss = 0.42381043
Iteration 319, loss = 0.42398277
Iteration 320, loss = 0.42422052
Iteration 321, loss = 0.42419768
Iteration 322, loss = 0.42450098
Iteration 323, loss = 0.42379044
Iteration 324, loss = 0.42334094
Iteration 325, loss = 0.42348604
Iteration 326, loss = 0.42404969
Iteration 327, loss = 0.42376387
Iteration 328, loss = 0.42322596
```

```
Iteration 329, loss = 0.42356666
Iteration 330, loss = 0.42308511
Iteration 331, loss = 0.42352260
Iteration 332, loss = 0.42368559
Iteration 333, loss = 0.42281107
Iteration 334, loss = 0.42263346
Iteration 335, loss = 0.42261533
Iteration 336, loss = 0.42268789
Iteration 337, loss = 0.42251415
Iteration 338, loss = 0.42242426
Iteration 339, loss = 0.42218754
Iteration 340, loss = 0.42213811
Iteration 341, loss = 0.42224828
Iteration 342, loss = 0.42215406
Iteration 343, loss = 0.42188142
Iteration 344, loss = 0.42215313
Iteration 345, loss = 0.42197048
Iteration 346, loss = 0.42191583
Iteration 347, loss = 0.42190666
Iteration 348, loss = 0.42152262
Iteration 349, loss = 0.42182529
Iteration 350, loss = 0.42142171
Iteration 351, loss = 0.42146593
Iteration 352, loss = 0.42134403
Iteration 353, loss = 0.42135268
Iteration 354, loss = 0.42132772
Iteration 355, loss = 0.42175639
Iteration 356, loss = 0.42128695
Iteration 357, loss = 0.42102091
Iteration 358, loss = 0.42084959
Iteration 359, loss = 0.42068975
Iteration 360, loss = 0.42093215
Iteration 361, loss = 0.42038992
Iteration 362, loss = 0.42030669
Iteration 363, loss = 0.42088146
Iteration 364, loss = 0.42060866
Iteration 365, loss = 0.42018567
Iteration 366, loss = 0.42027920
Iteration 367, loss = 0.41988085
Iteration 368, loss = 0.42047720
Iteration 369, loss = 0.42000490
Iteration 370, loss = 0.42004733
Iteration 371, loss = 0.41967011
Iteration 372, loss = 0.41969044
Iteration 373, loss = 0.41968654
Iteration 374, loss = 0.41974724
Iteration 375, loss = 0.42004279
Iteration 376, loss = 0.41969750
Iteration 377, loss = 0.41936616
Iteration 378, loss = 0.41923262
Iteration 379, loss = 0.41987764
Iteration 380, loss = 0.41921595
Iteration 381, loss = 0.41935240
Iteration 382, loss = 0.41953550
Iteration 383, loss = 0.41879505
Iteration 384, loss = 0.41876394
Iteration 385, loss = 0.41908059
Iteration 386, loss = 0.41893028
Iteration 387, loss = 0.41901088
Iteration 388, loss = 0.41864204
Iteration 389, loss = 0.41874949
```

```
Iteration 390, loss = 0.41884703
Iteration 391, loss = 0.41873782
Iteration 392, loss = 0.41842569
Iteration 393, loss = 0.41854921
Iteration 394, loss = 0.41842775
Iteration 395, loss = 0.41812789
Iteration 396, loss = 0.41807714
Iteration 397, loss = 0.41796047
Iteration 398, loss = 0.41766271
Iteration 399, loss = 0.41776606
Iteration 400, loss = 0.41786512
Iteration 401, loss = 0.41808778
Iteration 402, loss = 0.41772561
Iteration 403, loss = 0.41771334
Iteration 404, loss = 0.41754214
Iteration 405, loss = 0.41726498
Iteration 406, loss = 0.41722660
Iteration 407, loss = 0.41739007
Iteration 408, loss = 0.41730127
Iteration 409, loss = 0.41731643
Iteration 410, loss = 0.41708841
Iteration 411, loss = 0.41706962
Iteration 412, loss = 0.41683194
Iteration 413, loss = 0.41677918
Iteration 414, loss = 0.41676943
Iteration 415, loss = 0.41657616
Iteration 416, loss = 0.41747934
Iteration 417, loss = 0.41656941
Iteration 418, loss = 0.41674401
Iteration 419, loss = 0.41630259
Iteration 420, loss = 0.41636563
Iteration 421, loss = 0.41601379
Iteration 422, loss = 0.41599757
Iteration 423, loss = 0.41637519
Iteration 424, loss = 0.41615239
Iteration 425, loss = 0.41630544
Iteration 426, loss = 0.41602445
Iteration 427, loss = 0.41562028
Iteration 428, loss = 0.41650423
Iteration 429, loss = 0.41572005
Iteration 430, loss = 0.41599922
Iteration 431, loss = 0.41637910
Iteration 432, loss = 0.41534378
Iteration 433, loss = 0.41526798
Iteration 434, loss = 0.41555286
Iteration 435, loss = 0.41536852
Iteration 436, loss = 0.41547582
Iteration 437, loss = 0.41524784
Iteration 438, loss = 0.41485957
Iteration 439, loss = 0.41488121
Iteration 440, loss = 0.41547155
Iteration 441, loss = 0.41458705
Iteration 442, loss = 0.41480215
Iteration 443, loss = 0.41464603
Iteration 444, loss = 0.41469542
Iteration 445, loss = 0.41515971
Iteration 446, loss = 0.41600275
Iteration 447, loss = 0.41467064
Iteration 448, loss = 0.41478080
Iteration 449, loss = 0.41449239
Iteration 450, loss = 0.41490812
```

```
Iteration 451, loss = 0.41422301
Iteration 452, loss = 0.41459521
Iteration 453, loss = 0.41462301
Iteration 454, loss = 0.41414766
Iteration 455, loss = 0.41386427
Iteration 456, loss = 0.41404783
Iteration 457, loss = 0.41410216
Iteration 458, loss = 0.41388360
Iteration 459, loss = 0.41373852
Iteration 460, loss = 0.41453978
Iteration 461, loss = 0.41401590
Iteration 462, loss = 0.41351324
Iteration 463, loss = 0.41343349
Iteration 464, loss = 0.41315023
Iteration 465, loss = 0.41344311
Iteration 466, loss = 0.41320490
Iteration 467, loss = 0.41310779
Iteration 468, loss = 0.41307987
Iteration 469, loss = 0.41307346
Iteration 470, loss = 0.41309560
Iteration 471, loss = 0.41310686
Iteration 472, loss = 0.41276790
Iteration 473, loss = 0.41328892
Iteration 474, loss = 0.41264432
Iteration 475, loss = 0.41298649
Iteration 476, loss = 0.41267763
Iteration 477, loss = 0.41271218
Iteration 478, loss = 0.41338474
Iteration 479, loss = 0.41251115
Iteration 480, loss = 0.41272127
Iteration 481, loss = 0.41272798
Iteration 482, loss = 0.41259333
Iteration 483, loss = 0.41232543
Iteration 484, loss = 0.41288573
Iteration 485, loss = 0.41242938
Iteration 486, loss = 0.41215202
Iteration 487, loss = 0.41207929
Iteration 488, loss = 0.41208441
Iteration 489, loss = 0.41221784
Iteration 490, loss = 0.41221137
Iteration 491, loss = 0.41192020
Iteration 492, loss = 0.41198604
Iteration 493, loss = 0.41210221
Iteration 494, loss = 0.41187938
Iteration 495, loss = 0.41169877
Iteration 496, loss = 0.41160497
Iteration 497, loss = 0.41200962
Iteration 498, loss = 0.41131772
Iteration 499, loss = 0.41143445
Iteration 500, loss = 0.41116016
Iteration 1, loss = 0.78389366
Iteration 2, loss = 0.71467443
Iteration 3, loss = 0.65814497
Iteration 4, loss = 0.61549531
Iteration 5, loss = 0.58271435
Iteration 6, loss = 0.55829717
Iteration 7, loss = 0.53959121
Iteration 8, loss = 0.52533638
Iteration 9, loss = 0.51519221
Iteration 10, loss = 0.50777528
Iteration 11, loss = 0.50238456
```

```
Iteration 12, loss = 0.49827949
Iteration 13, loss = 0.49555037
Iteration 14, loss = 0.49327678
Iteration 15, loss = 0.49176956
Iteration 16, loss = 0.49029712
Iteration 17, loss = 0.48905038
Iteration 18, loss = 0.48815040
Iteration 19, loss = 0.48715207

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 20, loss = 0.48637938
Iteration 21, loss = 0.48557328
Iteration 22, loss = 0.48505878
Iteration 23, loss = 0.48423424
Iteration 24, loss = 0.48368817
Iteration 25, loss = 0.48305663
Iteration 26, loss = 0.48218243
Iteration 27, loss = 0.48176429
Iteration 28, loss = 0.48112335
Iteration 29, loss = 0.48051369
Iteration 30, loss = 0.47995935
Iteration 31, loss = 0.47950879
Iteration 32, loss = 0.47888473
Iteration 33, loss = 0.47843883
Iteration 34, loss = 0.47802721
Iteration 35, loss = 0.47752907
Iteration 36, loss = 0.47700099
Iteration 37, loss = 0.47649773
Iteration 38, loss = 0.47606856
Iteration 39, loss = 0.47559142
Iteration 40, loss = 0.47510491
Iteration 41, loss = 0.47510453
Iteration 42, loss = 0.47440410
Iteration 43, loss = 0.47395065
Iteration 44, loss = 0.47345370
Iteration 45, loss = 0.47320906
Iteration 46, loss = 0.47268309
Iteration 47, loss = 0.47230941
Iteration 48, loss = 0.47199862
Iteration 49, loss = 0.47135691
Iteration 50, loss = 0.47113799
Iteration 51, loss = 0.47073283
Iteration 52, loss = 0.47021489
Iteration 53, loss = 0.47004883
Iteration 54, loss = 0.46966284
Iteration 55, loss = 0.46926734
Iteration 56, loss = 0.46904798
Iteration 57, loss = 0.46902757
Iteration 58, loss = 0.46853009
Iteration 59, loss = 0.46785917
Iteration 60, loss = 0.46763163
Iteration 61, loss = 0.46725752
Iteration 62, loss = 0.46698682
Iteration 63, loss = 0.46685065
Iteration 64, loss = 0.46632744
Iteration 65, loss = 0.46595855
Iteration 66, loss = 0.46572081
Iteration 67, loss = 0.46521258
Iteration 68, loss = 0.46506269
Iteration 69, loss = 0.46476811
Iteration 70, loss = 0.46446262
Iteration 71, loss = 0.46410090
Iteration 72, loss = 0.46391600
Iteration 73, loss = 0.46348836
Iteration 74, loss = 0.46311598
Iteration 75, loss = 0.46275735
Iteration 76, loss = 0.46244698
Iteration 77, loss = 0.46229941
Iteration 78, loss = 0.46191651
Iteration 79, loss = 0.46170199
Iteration 80, loss = 0.46122259
```

```
Iteration 81, loss = 0.46099894
Iteration 82, loss = 0.46098636
Iteration 83, loss = 0.46053226
Iteration 84, loss = 0.46002456
Iteration 85, loss = 0.45971524
Iteration 86, loss = 0.45939177
Iteration 87, loss = 0.45935294
Iteration 88, loss = 0.45895607
Iteration 89, loss = 0.45852278
Iteration 90, loss = 0.45829431
Iteration 91, loss = 0.45781040
Iteration 92, loss = 0.45769552
Iteration 93, loss = 0.45760753
Iteration 94, loss = 0.45712647
Iteration 95, loss = 0.45697625
Iteration 96, loss = 0.45664250
Iteration 97, loss = 0.45634767
Iteration 98, loss = 0.45601176
Iteration 99, loss = 0.45576525
Iteration 100, loss = 0.45550041
Iteration 101, loss = 0.45520029
Iteration 102, loss = 0.45489714
Iteration 103, loss = 0.45487741
Iteration 104, loss = 0.45445877
Iteration 105, loss = 0.45424833
Iteration 106, loss = 0.45400130
Iteration 107, loss = 0.45375504
Iteration 108, loss = 0.45338091
Iteration 109, loss = 0.45345304
Iteration 110, loss = 0.45322391
Iteration 111, loss = 0.45296899
Iteration 112, loss = 0.45261084
Iteration 113, loss = 0.45223585
Iteration 114, loss = 0.45210611
Iteration 115, loss = 0.45180566
Iteration 116, loss = 0.45152299
Iteration 117, loss = 0.45144974
Iteration 118, loss = 0.45198620
Iteration 119, loss = 0.45128982
Iteration 120, loss = 0.45086831
Iteration 121, loss = 0.45076266
Iteration 122, loss = 0.45029107
Iteration 123, loss = 0.45030678
Iteration 124, loss = 0.45026782
Iteration 125, loss = 0.44980010
Iteration 126, loss = 0.44981009
Iteration 127, loss = 0.44937615
Iteration 128, loss = 0.44920569
Iteration 129, loss = 0.44897248
Iteration 130, loss = 0.44869727
Iteration 131, loss = 0.44864077
Iteration 132, loss = 0.44876383
Iteration 133, loss = 0.44854435
Iteration 134, loss = 0.44797429
Iteration 135, loss = 0.44806220
Iteration 136, loss = 0.44771245
Iteration 137, loss = 0.44728324
Iteration 138, loss = 0.44760934
Iteration 139, loss = 0.44726834
Iteration 140, loss = 0.44685825
Iteration 141, loss = 0.44712852
```

```
Iteration 142, loss = 0.44658615
Iteration 143, loss = 0.44647165
Iteration 144, loss = 0.44637201
Iteration 145, loss = 0.44607723
Iteration 146, loss = 0.44574680
Iteration 147, loss = 0.44558723
Iteration 148, loss = 0.44575829
Iteration 149, loss = 0.44534856
Iteration 150, loss = 0.44537180
Iteration 151, loss = 0.44519064
Iteration 152, loss = 0.44477425
Iteration 153, loss = 0.44467234
Iteration 154, loss = 0.44455167
Iteration 155, loss = 0.44460278
Iteration 156, loss = 0.44438471
Iteration 157, loss = 0.44396139
Iteration 158, loss = 0.44393771
Iteration 159, loss = 0.44358890
Iteration 160, loss = 0.44365481
Iteration 161, loss = 0.44365216
Iteration 162, loss = 0.44347705
Iteration 163, loss = 0.44324452
Iteration 164, loss = 0.44272848
Iteration 165, loss = 0.44297752
Iteration 166, loss = 0.44259602
Iteration 167, loss = 0.44222392
Iteration 168, loss = 0.44210957
Iteration 169, loss = 0.44204139
Iteration 170, loss = 0.44179960
Iteration 171, loss = 0.44202653
Iteration 172, loss = 0.44158640
Iteration 173, loss = 0.44147901
Iteration 174, loss = 0.44136517
Iteration 175, loss = 0.44103683
Iteration 176, loss = 0.44116762
Iteration 177, loss = 0.44093540
Iteration 178, loss = 0.44077637
Iteration 179, loss = 0.44053648
Iteration 180, loss = 0.44034160
Iteration 181, loss = 0.44044628
Iteration 182, loss = 0.43986371
Iteration 183, loss = 0.44021666
Iteration 184, loss = 0.43966231
Iteration 185, loss = 0.43951790
Iteration 186, loss = 0.43963362
Iteration 187, loss = 0.43926904
Iteration 188, loss = 0.43944477
Iteration 189, loss = 0.43896626
Iteration 190, loss = 0.43926477
Iteration 191, loss = 0.43875156
Iteration 192, loss = 0.43912752
Iteration 193, loss = 0.43853308
Iteration 194, loss = 0.43862102
Iteration 195, loss = 0.43818603
Iteration 196, loss = 0.43795892
Iteration 197, loss = 0.43811198
Iteration 198, loss = 0.43784457
Iteration 199, loss = 0.43737114
Iteration 200, loss = 0.43738650
Iteration 201, loss = 0.43720927
Iteration 202, loss = 0.43700369
```

```
Iteration 203, loss = 0.43697886
Iteration 204, loss = 0.43695027
Iteration 205, loss = 0.43698968
Iteration 206, loss = 0.43680811
Iteration 207, loss = 0.43651334
Iteration 208, loss = 0.43645453
Iteration 209, loss = 0.43636044
Iteration 210, loss = 0.43617375
Iteration 211, loss = 0.43618866
Iteration 212, loss = 0.43607120
Iteration 213, loss = 0.43585428
Iteration 214, loss = 0.43626324
Iteration 215, loss = 0.43566028
Iteration 216, loss = 0.43567206
Iteration 217, loss = 0.43530518
Iteration 218, loss = 0.43507938
Iteration 219, loss = 0.43489093
Iteration 220, loss = 0.43514459
Iteration 221, loss = 0.43535032
Iteration 222, loss = 0.43453773
Iteration 223, loss = 0.43434509
Iteration 224, loss = 0.43435634
Iteration 225, loss = 0.43415204
Iteration 226, loss = 0.43393478
Iteration 227, loss = 0.43394172
Iteration 228, loss = 0.43392727
Iteration 229, loss = 0.43378281
Iteration 230, loss = 0.43364566
Iteration 231, loss = 0.43360534
Iteration 232, loss = 0.43329124
Iteration 233, loss = 0.43342791
Iteration 234, loss = 0.43300234
Iteration 235, loss = 0.43293633
Iteration 236, loss = 0.43284372
Iteration 237, loss = 0.43268815
Iteration 238, loss = 0.43261165
Iteration 239, loss = 0.43249384
Iteration 240, loss = 0.43237656
Iteration 241, loss = 0.43225380
Iteration 242, loss = 0.43220416
Iteration 243, loss = 0.43226398
Iteration 244, loss = 0.43191151
Iteration 245, loss = 0.43156763
Iteration 246, loss = 0.43168663
Iteration 247, loss = 0.43141070
Iteration 248, loss = 0.43130906
Iteration 249, loss = 0.43128130
Iteration 250, loss = 0.43107837
Iteration 251, loss = 0.43153186
Iteration 252, loss = 0.43068514
Iteration 253, loss = 0.43106016
Iteration 254, loss = 0.43069588
Iteration 255, loss = 0.43050580
Iteration 256, loss = 0.43039465
Iteration 257, loss = 0.43038144
Iteration 258, loss = 0.43007230
Iteration 259, loss = 0.43013220
Iteration 260, loss = 0.42989628
Iteration 261, loss = 0.42987366
Iteration 262, loss = 0.42955302
Iteration 263, loss = 0.42977422
```

```
Iteration 264, loss = 0.42953021
Iteration 265, loss = 0.42950202
Iteration 266, loss = 0.42894346
Iteration 267, loss = 0.42913743
Iteration 268, loss = 0.42891353
Iteration 269, loss = 0.42896987
Iteration 270, loss = 0.42893369
Iteration 271, loss = 0.42860295
Iteration 272, loss = 0.42845553
Iteration 273, loss = 0.42862143
Iteration 274, loss = 0.42808306
Iteration 275, loss = 0.42813408
Iteration 276, loss = 0.42822626
Iteration 277, loss = 0.42815112
Iteration 278, loss = 0.42815804
Iteration 279, loss = 0.42766122
Iteration 280, loss = 0.42801437
Iteration 281, loss = 0.42779110
Iteration 282, loss = 0.42793345
Iteration 283, loss = 0.42724513
Iteration 284, loss = 0.42762744
Iteration 285, loss = 0.42713435
Iteration 286, loss = 0.42704172
Iteration 287, loss = 0.42709651
Iteration 288, loss = 0.42682629
Iteration 289, loss = 0.42661920
Iteration 290, loss = 0.42657182
Iteration 291, loss = 0.42646930
Iteration 292, loss = 0.42676962
Iteration 293, loss = 0.42617581
Iteration 294, loss = 0.42632562
Iteration 295, loss = 0.42589095
Iteration 296, loss = 0.42637919
Iteration 297, loss = 0.42575543
Iteration 298, loss = 0.42539548
Iteration 299, loss = 0.42559777
Iteration 300, loss = 0.42529767
Iteration 301, loss = 0.42529194
Iteration 302, loss = 0.42519622
Iteration 303, loss = 0.42491999
Iteration 304, loss = 0.42551788
Iteration 305, loss = 0.42492246
Iteration 306, loss = 0.42500355
Iteration 307, loss = 0.42470097
Iteration 308, loss = 0.42456385
Iteration 309, loss = 0.42462464
Iteration 310, loss = 0.42418528
Iteration 311, loss = 0.42436183
Iteration 312, loss = 0.42429641
Iteration 313, loss = 0.42387420
Iteration 314, loss = 0.42396319
Iteration 315, loss = 0.42399750
Iteration 316, loss = 0.42391073
Iteration 317, loss = 0.42413870
Iteration 318, loss = 0.42394282
Iteration 319, loss = 0.42330478
Iteration 320, loss = 0.42338221
Iteration 321, loss = 0.42388883
Iteration 322, loss = 0.42323257
Iteration 323, loss = 0.42318169
Iteration 324, loss = 0.42303528
```

```
Iteration 325, loss = 0.42296278
Iteration 326, loss = 0.42267302
Iteration 327, loss = 0.42293224
Iteration 328, loss = 0.42227673
Iteration 329, loss = 0.42241852
Iteration 330, loss = 0.42225042
Iteration 331, loss = 0.42227365
Iteration 332, loss = 0.42209027
Iteration 333, loss = 0.42220395
Iteration 334, loss = 0.42196026
Iteration 335, loss = 0.42186120
Iteration 336, loss = 0.42157905
Iteration 337, loss = 0.42167309
Iteration 338, loss = 0.42158932
Iteration 339, loss = 0.42180851
Iteration 340, loss = 0.42152467
Iteration 341, loss = 0.42170672
Iteration 342, loss = 0.42195793
Iteration 343, loss = 0.42128872
Iteration 344, loss = 0.42121982
Iteration 345, loss = 0.42105517
Iteration 346, loss = 0.42147761
Iteration 347, loss = 0.42046002
Iteration 348, loss = 0.42035262
Iteration 349, loss = 0.42019955
Iteration 350, loss = 0.42028385
Iteration 351, loss = 0.42109931
Iteration 352, loss = 0.42026991
Iteration 353, loss = 0.41981105
Iteration 354, loss = 0.42024274
Iteration 355, loss = 0.41973451
Iteration 356, loss = 0.41974004
Iteration 357, loss = 0.41974444
Iteration 358, loss = 0.41943378
Iteration 359, loss = 0.41964143
Iteration 360, loss = 0.41965981
Iteration 361, loss = 0.42005769
Iteration 362, loss = 0.41904442
Iteration 363, loss = 0.41889522
Iteration 364, loss = 0.41875922
Iteration 365, loss = 0.41923421
Iteration 366, loss = 0.41869222
Iteration 367, loss = 0.41873186
Iteration 368, loss = 0.41938992
Iteration 369, loss = 0.41831913
Iteration 370, loss = 0.41920817
Iteration 371, loss = 0.41867386
Iteration 372, loss = 0.41824633
Iteration 373, loss = 0.41844666
Iteration 374, loss = 0.41833217
Iteration 375, loss = 0.41806416
Iteration 376, loss = 0.41783626
Iteration 377, loss = 0.41805944
Iteration 378, loss = 0.41794850
Iteration 379, loss = 0.41746404
Iteration 380, loss = 0.41747598
Iteration 381, loss = 0.41839224
Iteration 382, loss = 0.41722748
Iteration 383, loss = 0.41704268
Iteration 384, loss = 0.41750515
Iteration 385, loss = 0.41717462
```

```
Iteration 386, loss = 0.41686380
Iteration 387, loss = 0.41683957
Iteration 388, loss = 0.41649554
Iteration 389, loss = 0.41660061
Iteration 390, loss = 0.41643007
Iteration 391, loss = 0.41686877
Iteration 392, loss = 0.41624195
Iteration 393, loss = 0.41626274
Iteration 394, loss = 0.41635017
Iteration 395, loss = 0.41614121
Iteration 396, loss = 0.41604738
Iteration 397, loss = 0.41613055
Iteration 398, loss = 0.41597831
Iteration 399, loss = 0.41571693
Iteration 400, loss = 0.41543291
Iteration 401, loss = 0.41564157
Iteration 402, loss = 0.41573258
Iteration 403, loss = 0.41533125
Iteration 404, loss = 0.41547502
Iteration 405, loss = 0.41599128
Iteration 406, loss = 0.41556638
Iteration 407, loss = 0.41490705
Iteration 408, loss = 0.41476196
Iteration 409, loss = 0.41443884
Iteration 410, loss = 0.41513129
Iteration 411, loss = 0.41461076
Iteration 412, loss = 0.41521471
Iteration 413, loss = 0.41469748
Iteration 414, loss = 0.41419447
Iteration 415, loss = 0.41427728
Iteration 416, loss = 0.41440952
Iteration 417, loss = 0.41393320
Iteration 418, loss = 0.41378194
Iteration 419, loss = 0.41410997
Iteration 420, loss = 0.41364025
Iteration 421, loss = 0.41367511
Iteration 422, loss = 0.41400448
Iteration 423, loss = 0.41391015
Iteration 424, loss = 0.41425346
Iteration 425, loss = 0.41327143
Iteration 426, loss = 0.41291640
Iteration 427, loss = 0.41313733
Iteration 428, loss = 0.41336942
Iteration 429, loss = 0.41304866
Iteration 430, loss = 0.41334583
Iteration 431, loss = 0.41355274
Iteration 432, loss = 0.41252496
Iteration 433, loss = 0.41284073
Iteration 434, loss = 0.41314395
Iteration 435, loss = 0.41260977
Iteration 436, loss = 0.41230721
Iteration 437, loss = 0.41218060
Iteration 438, loss = 0.41208507
Iteration 439, loss = 0.41194044
Iteration 440, loss = 0.41184151
Iteration 441, loss = 0.41200350
Iteration 442, loss = 0.41305726
Iteration 443, loss = 0.41174014
Iteration 444, loss = 0.41233763
Iteration 445, loss = 0.41223453
Iteration 446, loss = 0.41170381
```

```
Iteration 447, loss = 0.41150130
Iteration 448, loss = 0.41129237
Iteration 449, loss = 0.41151133
Iteration 450, loss = 0.41109801
Iteration 451, loss = 0.41097768
Iteration 452, loss = 0.41108790
Iteration 453, loss = 0.41120106
Iteration 454, loss = 0.41078396
Iteration 455, loss = 0.41073744
Iteration 456, loss = 0.41070537
Iteration 457, loss = 0.41063306
Iteration 458, loss = 0.41031554
Iteration 459, loss = 0.41018336
Iteration 460, loss = 0.41029837
Iteration 461, loss = 0.41023724
Iteration 462, loss = 0.41079348
Iteration 463, loss = 0.40989690
Iteration 464, loss = 0.41065559
Iteration 465, loss = 0.40976734
Iteration 466, loss = 0.41002875
Iteration 467, loss = 0.40943419
Iteration 468, loss = 0.40954463
Iteration 469, loss = 0.40945403
Iteration 470, loss = 0.41000074
Iteration 471, loss = 0.40916604
Iteration 472, loss = 0.40953486
Iteration 473, loss = 0.40895488
Iteration 474, loss = 0.40930093
Iteration 475, loss = 0.40964099
Iteration 476, loss = 0.40864846
Iteration 477, loss = 0.40865491
Iteration 478, loss = 0.40840708
Iteration 479, loss = 0.40830068
Iteration 480, loss = 0.40862762
Iteration 481, loss = 0.40872442
Iteration 482, loss = 0.40850551
Iteration 483, loss = 0.40795501
Iteration 484, loss = 0.40813831
Iteration 485, loss = 0.40780283
Iteration 486, loss = 0.40790864
Iteration 487, loss = 0.40793199
Iteration 488, loss = 0.40803405
Iteration 489, loss = 0.40788841
Iteration 490, loss = 0.40720366
Iteration 491, loss = 0.40737493
Iteration 492, loss = 0.40750353
Iteration 493, loss = 0.40696911
Iteration 494, loss = 0.40708531
Iteration 495, loss = 0.40830608
Iteration 496, loss = 0.40672998
Iteration 497, loss = 0.40712132
Iteration 498, loss = 0.40676197
Iteration 499, loss = 0.40665377
Iteration 500, loss = 0.40657137
Iteration 1, loss = 0.61278296
Iteration 2, loss = 0.57513515
Iteration 3, loss = 0.54918132
Iteration 4, loss = 0.52961829
Iteration 5, loss = 0.51627337
Iteration 6, loss = 0.50701422
Iteration 7, loss = 0.50088775
```

```
Iteration 8, loss = 0.49657383
Iteration 9, loss = 0.49414346
Iteration 10, loss = 0.49166402

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 11, loss = 0.49026658
Iteration 12, loss = 0.48888281
Iteration 13, loss = 0.48772194
Iteration 14, loss = 0.48707190
Iteration 15, loss = 0.48607781
Iteration 16, loss = 0.48542716
Iteration 17, loss = 0.48450298
Iteration 18, loss = 0.48398641
Iteration 19, loss = 0.48321147
Iteration 20, loss = 0.48271506
Iteration 21, loss = 0.48217118
Iteration 22, loss = 0.48153810
Iteration 23, loss = 0.48125343
Iteration 24, loss = 0.48032027
Iteration 25, loss = 0.47978666
Iteration 26, loss = 0.47948893
Iteration 27, loss = 0.47881236
Iteration 28, loss = 0.47819075
Iteration 29, loss = 0.47773568
Iteration 30, loss = 0.47723888
Iteration 31, loss = 0.47686935
Iteration 32, loss = 0.47633791
Iteration 33, loss = 0.47580853
Iteration 34, loss = 0.47528573
Iteration 35, loss = 0.47483645
Iteration 36, loss = 0.47440026
Iteration 37, loss = 0.47399994
Iteration 38, loss = 0.47358348
Iteration 39, loss = 0.47322342
Iteration 40, loss = 0.47286927
Iteration 41, loss = 0.47235564
Iteration 42, loss = 0.47195618
Iteration 43, loss = 0.47147713
Iteration 44, loss = 0.47106908
Iteration 45, loss = 0.47073944
Iteration 46, loss = 0.47050592
Iteration 47, loss = 0.46981405
Iteration 48, loss = 0.46945602
Iteration 49, loss = 0.46912558
Iteration 50, loss = 0.46875465
Iteration 51, loss = 0.46833083
Iteration 52, loss = 0.46786652
Iteration 53, loss = 0.46753891
Iteration 54, loss = 0.46739704
Iteration 55, loss = 0.46700407
Iteration 56, loss = 0.46676283
Iteration 57, loss = 0.46615944
Iteration 58, loss = 0.46597319
Iteration 59, loss = 0.46582618
Iteration 60, loss = 0.46503325
Iteration 61, loss = 0.46496393
Iteration 62, loss = 0.46457608
Iteration 63, loss = 0.46429497
Iteration 64, loss = 0.46396260
Iteration 65, loss = 0.46365431
Iteration 66, loss = 0.46345087
Iteration 67, loss = 0.46323759
Iteration 68, loss = 0.46283216
Iteration 69, loss = 0.46251843
Iteration 70, loss = 0.46269125
Iteration 71, loss = 0.46217175
```

```
Iteration 72, loss = 0.46155354
Iteration 73, loss = 0.46163493
Iteration 74, loss = 0.46115509
Iteration 75, loss = 0.46079646
Iteration 76, loss = 0.46073958
Iteration 77, loss = 0.46032684
Iteration 78, loss = 0.45995130
Iteration 79, loss = 0.45969889
Iteration 80, loss = 0.45966716
Iteration 81, loss = 0.45932167
Iteration 82, loss = 0.45905235
Iteration 83, loss = 0.45904786
Iteration 84, loss = 0.45844463
Iteration 85, loss = 0.45818452
Iteration 86, loss = 0.45801567
Iteration 87, loss = 0.45788558
Iteration 88, loss = 0.45765955
Iteration 89, loss = 0.45740914
Iteration 90, loss = 0.45712239
Iteration 91, loss = 0.45701187
Iteration 92, loss = 0.45636807
Iteration 93, loss = 0.45642334
Iteration 94, loss = 0.45611113
Iteration 95, loss = 0.45579796
Iteration 96, loss = 0.45551925
Iteration 97, loss = 0.45564905
Iteration 98, loss = 0.45598448
Iteration 99, loss = 0.45521377
Iteration 100, loss = 0.45528725
Iteration 101, loss = 0.45467670
Iteration 102, loss = 0.45485705
Iteration 103, loss = 0.45443473
Iteration 104, loss = 0.45425530
Iteration 105, loss = 0.45416804
Iteration 106, loss = 0.45349569
Iteration 107, loss = 0.45345261
Iteration 108, loss = 0.45339927
Iteration 109, loss = 0.45334728
Iteration 110, loss = 0.45288696
Iteration 111, loss = 0.45278642
Iteration 112, loss = 0.45256206
Iteration 113, loss = 0.45233598
Iteration 114, loss = 0.45210782
Iteration 115, loss = 0.45176014
Iteration 116, loss = 0.45176351
Iteration 117, loss = 0.45153544
Iteration 118, loss = 0.45125278
Iteration 119, loss = 0.45147808
Iteration 120, loss = 0.45121049
Iteration 121, loss = 0.45069561
Iteration 122, loss = 0.45069956
Iteration 123, loss = 0.45057024
Iteration 124, loss = 0.45034995
Iteration 125, loss = 0.45101268
Iteration 126, loss = 0.45004736
Iteration 127, loss = 0.44970567
Iteration 128, loss = 0.44981463
Iteration 129, loss = 0.44960636
Iteration 130, loss = 0.44938210
Iteration 131, loss = 0.44929711
Iteration 132, loss = 0.44907222
```

```
Iteration 133, loss = 0.44903182
Iteration 134, loss = 0.44861732
Iteration 135, loss = 0.44850191
Iteration 136, loss = 0.44867509
Iteration 137, loss = 0.44843021
Iteration 138, loss = 0.44824882
Iteration 139, loss = 0.44789101
Iteration 140, loss = 0.44763884
Iteration 141, loss = 0.44773588
Iteration 142, loss = 0.44731508
Iteration 143, loss = 0.44741140
Iteration 144, loss = 0.44716208
Iteration 145, loss = 0.44735533
Iteration 146, loss = 0.44691324
Iteration 147, loss = 0.44659055
Iteration 148, loss = 0.44646639
Iteration 149, loss = 0.44663857
Iteration 150, loss = 0.44631449
Iteration 151, loss = 0.44601219
Iteration 152, loss = 0.44583908
Iteration 153, loss = 0.44566823
Iteration 154, loss = 0.44568676
Iteration 155, loss = 0.44535261
Iteration 156, loss = 0.44507681
Iteration 157, loss = 0.44532376
Iteration 158, loss = 0.44488918
Iteration 159, loss = 0.44476631
Iteration 160, loss = 0.44487152
Iteration 161, loss = 0.44493967
Iteration 162, loss = 0.44426491
Iteration 163, loss = 0.44396391
Iteration 164, loss = 0.44409429
Iteration 165, loss = 0.44409789
Iteration 166, loss = 0.44367520
Iteration 167, loss = 0.44362744
Iteration 168, loss = 0.44333389
Iteration 169, loss = 0.44320490
Iteration 170, loss = 0.44353649
Iteration 171, loss = 0.44316482
Iteration 172, loss = 0.44267643
Iteration 173, loss = 0.44272147
Iteration 174, loss = 0.44304423
Iteration 175, loss = 0.44247829
Iteration 176, loss = 0.44205941
Iteration 177, loss = 0.44204857
Iteration 178, loss = 0.44193050
Iteration 179, loss = 0.44191495
Iteration 180, loss = 0.44178292
Iteration 181, loss = 0.44145217
Iteration 182, loss = 0.44134637
Iteration 183, loss = 0.44110935
Iteration 184, loss = 0.44091968
Iteration 185, loss = 0.44122713
Iteration 186, loss = 0.44101610
Iteration 187, loss = 0.44059326
Iteration 188, loss = 0.44069124
Iteration 189, loss = 0.44057434
Iteration 190, loss = 0.44066026
Iteration 191, loss = 0.44035760
Iteration 192, loss = 0.44054946
Iteration 193, loss = 0.44021263
```

```
Iteration 194, loss = 0.43980896
Iteration 195, loss = 0.43991178
Iteration 196, loss = 0.43953070
Iteration 197, loss = 0.43952753
Iteration 198, loss = 0.43981904
Iteration 199, loss = 0.43907982
Iteration 200, loss = 0.43929865
Iteration 201, loss = 0.43891594
Iteration 202, loss = 0.43889297
Iteration 203, loss = 0.43912462
Iteration 204, loss = 0.43852035
Iteration 205, loss = 0.43851221
Iteration 206, loss = 0.43838137
Iteration 207, loss = 0.43803396
Iteration 208, loss = 0.43818209
Iteration 209, loss = 0.43792074
Iteration 210, loss = 0.43793995
Iteration 211, loss = 0.43787310
Iteration 212, loss = 0.43753680
Iteration 213, loss = 0.43734737
Iteration 214, loss = 0.43736532
Iteration 215, loss = 0.43711929
Iteration 216, loss = 0.43758161
Iteration 217, loss = 0.43731368
Iteration 218, loss = 0.43744696
Iteration 219, loss = 0.43650128
Iteration 220, loss = 0.43657878
Iteration 221, loss = 0.43638938
Iteration 222, loss = 0.43663759
Iteration 223, loss = 0.43621488
Iteration 224, loss = 0.43599165
Iteration 225, loss = 0.43594762
Iteration 226, loss = 0.43579505
Iteration 227, loss = 0.43586355
Iteration 228, loss = 0.43596883
Iteration 229, loss = 0.43516891
Iteration 230, loss = 0.43552113
Iteration 231, loss = 0.43535419
Iteration 232, loss = 0.43519142
Iteration 233, loss = 0.43554585
Iteration 234, loss = 0.43484938
Iteration 235, loss = 0.43525800
Iteration 236, loss = 0.43469396
Iteration 237, loss = 0.43433044
Iteration 238, loss = 0.43451835
Iteration 239, loss = 0.43427325
Iteration 240, loss = 0.43458719
Iteration 241, loss = 0.43399233
Iteration 242, loss = 0.43361135
Iteration 243, loss = 0.43376365
Iteration 244, loss = 0.43365714
Iteration 245, loss = 0.43349607
Iteration 246, loss = 0.43343926
Iteration 247, loss = 0.43325474
Iteration 248, loss = 0.43347985
Iteration 249, loss = 0.43301946
Iteration 250, loss = 0.43319680
Iteration 251, loss = 0.43290627
Iteration 252, loss = 0.43268970
Iteration 253, loss = 0.43276346
Iteration 254, loss = 0.43276923
```

```
Iteration 255, loss = 0.43254325
Iteration 256, loss = 0.43216984
Iteration 257, loss = 0.43295086
Iteration 258, loss = 0.43201943
Iteration 259, loss = 0.43192093
Iteration 260, loss = 0.43189808
Iteration 261, loss = 0.43151605
Iteration 262, loss = 0.43167760
Iteration 263, loss = 0.43143800
Iteration 264, loss = 0.43188717
Iteration 265, loss = 0.43178174
Iteration 266, loss = 0.43120479
Iteration 267, loss = 0.43105573
Iteration 268, loss = 0.43076893
Iteration 269, loss = 0.43099551
Iteration 270, loss = 0.43091848
Iteration 271, loss = 0.43059305
Iteration 272, loss = 0.43076801
Iteration 273, loss = 0.43031587
Iteration 274, loss = 0.43044404
Iteration 275, loss = 0.43013488
Iteration 276, loss = 0.43032757
Iteration 277, loss = 0.42995621
Iteration 278, loss = 0.42963808
Iteration 279, loss = 0.43016560
Iteration 280, loss = 0.42986359
Iteration 281, loss = 0.42962540
Iteration 282, loss = 0.42965761
Iteration 283, loss = 0.42988748
Iteration 284, loss = 0.42928809
Iteration 285, loss = 0.42942324
Iteration 286, loss = 0.42914598
Iteration 287, loss = 0.42874893
Iteration 288, loss = 0.42893349
Iteration 289, loss = 0.42895473
Iteration 290, loss = 0.42884606
Iteration 291, loss = 0.42851115
Iteration 292, loss = 0.42867548
Iteration 293, loss = 0.42829306
Iteration 294, loss = 0.42843135
Iteration 295, loss = 0.42823187
Iteration 296, loss = 0.42823510
Iteration 297, loss = 0.42821087
Iteration 298, loss = 0.42788181
Iteration 299, loss = 0.42771397
Iteration 300, loss = 0.42774365
Iteration 301, loss = 0.42768304
Iteration 302, loss = 0.42730372
Iteration 303, loss = 0.42734385
Iteration 304, loss = 0.42732154
Iteration 305, loss = 0.42715017
Iteration 306, loss = 0.42685384
Iteration 307, loss = 0.42687680
Iteration 308, loss = 0.42696918
Iteration 309, loss = 0.42704533
Iteration 310, loss = 0.42694540
Iteration 311, loss = 0.42668083
Iteration 312, loss = 0.42659637
Iteration 313, loss = 0.42686250
Iteration 314, loss = 0.42643704
Iteration 315, loss = 0.42643292
```

```
Iteration 316, loss = 0.42640947
Iteration 317, loss = 0.42646569
Iteration 318, loss = 0.42657022
Iteration 319, loss = 0.42607340
Iteration 320, loss = 0.42583494
Iteration 321, loss = 0.42643988
Iteration 322, loss = 0.42576915
Iteration 323, loss = 0.42590677
Iteration 324, loss = 0.42564579
Iteration 325, loss = 0.42533728
Iteration 326, loss = 0.42533495
Iteration 327, loss = 0.42541006
Iteration 328, loss = 0.42568718
Iteration 329, loss = 0.42551413
Iteration 330, loss = 0.42516062
Iteration 331, loss = 0.42492632
Iteration 332, loss = 0.42486165
Iteration 333, loss = 0.42465647
Iteration 334, loss = 0.42457385
Iteration 335, loss = 0.42460885
Iteration 336, loss = 0.42463643
Iteration 337, loss = 0.42441035
Iteration 338, loss = 0.42444373
Iteration 339, loss = 0.42426770
Iteration 340, loss = 0.42416226
Iteration 341, loss = 0.42393862
Iteration 342, loss = 0.42410323
Iteration 343, loss = 0.42402141
Iteration 344, loss = 0.42385330
Iteration 345, loss = 0.42376230
Iteration 346, loss = 0.42389005
Iteration 347, loss = 0.42348354
Iteration 348, loss = 0.42329059
Iteration 349, loss = 0.42313235
Iteration 350, loss = 0.42361958
Iteration 351, loss = 0.42352051
Iteration 352, loss = 0.42326089
Iteration 353, loss = 0.42318984
Iteration 354, loss = 0.42276516
Iteration 355, loss = 0.42281140
Iteration 356, loss = 0.42249904
Iteration 357, loss = 0.42257291
Iteration 358, loss = 0.42237653
Iteration 359, loss = 0.42250076
Iteration 360, loss = 0.42254381
Iteration 361, loss = 0.42263208
Iteration 362, loss = 0.42239370
Iteration 363, loss = 0.42230782
Iteration 364, loss = 0.42258151
Iteration 365, loss = 0.42224563
Iteration 366, loss = 0.42237176
Iteration 367, loss = 0.42212505
Iteration 368, loss = 0.42219026
Iteration 369, loss = 0.42223487
Iteration 370, loss = 0.42180351
Iteration 371, loss = 0.42168438
Iteration 372, loss = 0.42179928
Iteration 373, loss = 0.42152285
Iteration 374, loss = 0.42133829
Iteration 375, loss = 0.42103205
Iteration 376, loss = 0.42169209
```

```
Iteration 377, loss = 0.42151327
Iteration 378, loss = 0.42122792
Iteration 379, loss = 0.42075341
Iteration 380, loss = 0.42103034
Iteration 381, loss = 0.42074632
Iteration 382, loss = 0.42071996
Iteration 383, loss = 0.42109395
Iteration 384, loss = 0.42062220
Iteration 385, loss = 0.42042351
Iteration 386, loss = 0.42077194
Iteration 387, loss = 0.42044047
Iteration 388, loss = 0.42025806
Iteration 389, loss = 0.41998946
Iteration 390, loss = 0.42026685
Iteration 391, loss = 0.42039735
Iteration 392, loss = 0.42026583
Iteration 393, loss = 0.41974239
Iteration 394, loss = 0.41994438
Iteration 395, loss = 0.41991194
Iteration 396, loss = 0.42001695
Iteration 397, loss = 0.41966473
Iteration 398, loss = 0.41949603
Iteration 399, loss = 0.41995170
Iteration 400, loss = 0.41925443
Iteration 401, loss = 0.41950944
Iteration 402, loss = 0.41948142
Iteration 403, loss = 0.41924542
Iteration 404, loss = 0.41914577
Iteration 405, loss = 0.41909859
Iteration 406, loss = 0.41884860
Iteration 407, loss = 0.41906635
Iteration 408, loss = 0.41914103
Iteration 409, loss = 0.41863779
Iteration 410, loss = 0.41860128
Iteration 411, loss = 0.41889399
Iteration 412, loss = 0.41838803
Iteration 413, loss = 0.41839001
Iteration 414, loss = 0.41833736
Iteration 415, loss = 0.41830838
Iteration 416, loss = 0.41815149
Iteration 417, loss = 0.41775733
Iteration 418, loss = 0.41797355
Iteration 419, loss = 0.41798601
Iteration 420, loss = 0.41781519
Iteration 421, loss = 0.41745973
Iteration 422, loss = 0.41762676
Iteration 423, loss = 0.41725736
Iteration 424, loss = 0.41731595
Iteration 425, loss = 0.41751359
Iteration 426, loss = 0.41722955
Iteration 427, loss = 0.41772046
Iteration 428, loss = 0.41684612
Iteration 429, loss = 0.41744144
Iteration 430, loss = 0.41677651
Iteration 431, loss = 0.41704516
Iteration 432, loss = 0.41688728
Iteration 433, loss = 0.41644077
Iteration 434, loss = 0.41704293
Iteration 435, loss = 0.41662028
Iteration 436, loss = 0.41631808
Iteration 437, loss = 0.41668141
```

```
Iteration 438, loss = 0.41625361
Iteration 439, loss = 0.41694386
Iteration 440, loss = 0.41670775
Iteration 441, loss = 0.41628763
Iteration 442, loss = 0.41622921
Iteration 443, loss = 0.41592721
Iteration 444, loss = 0.41643387
Iteration 445, loss = 0.41608863
Iteration 446, loss = 0.41626382
Iteration 447, loss = 0.41571567
Iteration 448, loss = 0.41579389
Iteration 449, loss = 0.41550844
Iteration 450, loss = 0.41537174
Iteration 451, loss = 0.41508550
Iteration 452, loss = 0.41537958
Iteration 453, loss = 0.41503784
Iteration 454, loss = 0.41514342
Iteration 455, loss = 0.41488086
Iteration 456, loss = 0.41505854
Iteration 457, loss = 0.41493460
Iteration 458, loss = 0.41562633
Iteration 459, loss = 0.41594570
Iteration 460, loss = 0.41544862
Iteration 461, loss = 0.41464156
Iteration 462, loss = 0.41438260
Iteration 463, loss = 0.41498762
Iteration 464, loss = 0.41437267
Iteration 465, loss = 0.41469099
Iteration 466, loss = 0.41445102
Iteration 467, loss = 0.41421886
Iteration 468, loss = 0.41398333
Iteration 469, loss = 0.41439720
Iteration 470, loss = 0.41391335
Iteration 471, loss = 0.41413460
Iteration 472, loss = 0.41376969
Iteration 473, loss = 0.41351237
Iteration 474, loss = 0.41401248
Iteration 475, loss = 0.41378908
Iteration 476, loss = 0.41342126
Iteration 477, loss = 0.41342076
Iteration 478, loss = 0.41357408
Iteration 479, loss = 0.41336690
Iteration 480, loss = 0.41298191
Iteration 481, loss = 0.41317453
Iteration 482, loss = 0.41291243
Iteration 483, loss = 0.41297797
Iteration 484, loss = 0.41343240
Iteration 485, loss = 0.41371362
Iteration 486, loss = 0.41362068
Iteration 487, loss = 0.41319089
Iteration 488, loss = 0.41286345
Iteration 489, loss = 0.41307808
Iteration 490, loss = 0.41240049
Iteration 491, loss = 0.41215177
Iteration 492, loss = 0.41243222
Iteration 493, loss = 0.41239396
Iteration 494, loss = 0.41266785
Iteration 495, loss = 0.41233409
Iteration 496, loss = 0.41224066
Iteration 497, loss = 0.41185543
Iteration 498, loss = 0.41171717
```

```
Iteration 499, loss = 0.41218876
Iteration 500, loss = 0.41178969
Iteration 1, loss = 0.59800190
Iteration 2, loss = 0.53673879
Iteration 3, loss = 0.50567194
Iteration 4, loss = 0.49261220
Iteration 5, loss = 0.48748673
Iteration 6, loss = 0.48369453
Iteration 7, loss = 0.48159773

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 8, loss = 0.47987736
Iteration 9, loss = 0.47862452
Iteration 10, loss = 0.47712823
Iteration 11, loss = 0.47605385
Iteration 12, loss = 0.47517208
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.62805716
Iteration 2, loss = 0.55981069
Iteration 3, loss = 0.52379328
Iteration 4, loss = 0.50746259
Iteration 5, loss = 0.50065867
Iteration 6, loss = 0.49690598
Iteration 7, loss = 0.49508530
Iteration 8, loss = 0.49232605
Iteration 9, loss = 0.49081778
Iteration 10, loss = 0.48881896
Iteration 11, loss = 0.48743242
Iteration 12, loss = 0.48622119
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.61487900
Iteration 2, loss = 0.55260379
Iteration 3, loss = 0.51881695
Iteration 4, loss = 0.50257711
Iteration 5, loss = 0.49467481
Iteration 6, loss = 0.49095241
Iteration 7, loss = 0.48898448
Iteration 8, loss = 0.48739296
Iteration 9, loss = 0.48642977
Iteration 10, loss = 0.48513475
Iteration 11, loss = 0.48382394
Iteration 12, loss = 0.48266984
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.70199709
Iteration 2, loss = 0.60623986
Iteration 3, loss = 0.54855067
Iteration 4, loss = 0.51722161
Iteration 5, loss = 0.50040550
Iteration 6, loss = 0.49150357
Iteration 7, loss = 0.48561836
Iteration 8, loss = 0.48361008
Iteration 9, loss = 0.48141296
Iteration 10, loss = 0.48034491
Iteration 11, loss = 0.47870532
Iteration 12, loss = 0.47736763
Iteration 13, loss = 0.47638166
Iteration 14, loss = 0.47534724
Iteration 15, loss = 0.47410666
Iteration 16, loss = 0.47366681
Iteration 17, loss = 0.47256283
Iteration 18, loss = 0.47190757
Iteration 19, loss = 0.47083206
Iteration 20, loss = 0.47001430
Iteration 21, loss = 0.46910763
Iteration 22, loss = 0.46853905
Iteration 23, loss = 0.46801633
Iteration 24, loss = 0.46704682
Iteration 25, loss = 0.46644403
Iteration 26, loss = 0.46566024
```

```
Iteration 27, loss = 0.46498846
Iteration 28, loss = 0.46456511
Iteration 29, loss = 0.46371364
Iteration 30, loss = 0.46354596
Iteration 31, loss = 0.46331542
Iteration 32, loss = 0.46245558
Iteration 33, loss = 0.46151491
Iteration 34, loss = 0.46114480
Iteration 35, loss = 0.46064699
Iteration 36, loss = 0.46003663
Iteration 37, loss = 0.45986706
Iteration 38, loss = 0.45921976
Iteration 39, loss = 0.45886767
Iteration 40, loss = 0.45836387
Iteration 41, loss = 0.45771824
Iteration 42, loss = 0.45757674
Iteration 43, loss = 0.45701442
Iteration 44, loss = 0.45639700
Iteration 45, loss = 0.45651159
Iteration 46, loss = 0.45601240
Iteration 47, loss = 0.45552436
Iteration 48, loss = 0.45505261
Iteration 49, loss = 0.45427544
Iteration 50, loss = 0.45436308
Iteration 51, loss = 0.45411410
Iteration 52, loss = 0.45323772
Iteration 53, loss = 0.45300129
Iteration 54, loss = 0.45284959
Iteration 55, loss = 0.45247492
Iteration 56, loss = 0.45208721
Iteration 57, loss = 0.45204411
Iteration 58, loss = 0.45170180
Iteration 59, loss = 0.45090736
Iteration 60, loss = 0.45062394
Iteration 61, loss = 0.45064186
Iteration 62, loss = 0.44998215
Iteration 63, loss = 0.44984682
Iteration 64, loss = 0.44919126
Iteration 65, loss = 0.44957818
Iteration 66, loss = 0.44919136
Iteration 67, loss = 0.44886040
Iteration 68, loss = 0.44800554
Iteration 69, loss = 0.44778414
Iteration 70, loss = 0.44750162
Iteration 71, loss = 0.44697250
Iteration 72, loss = 0.44671015
Iteration 73, loss = 0.44673205
Iteration 74, loss = 0.44624490
Iteration 75, loss = 0.44616754
Iteration 76, loss = 0.44615195
Iteration 77, loss = 0.44537962
Iteration 78, loss = 0.44550616
Iteration 79, loss = 0.44512496
Iteration 80, loss = 0.44463920
Iteration 81, loss = 0.44453595
Iteration 82, loss = 0.44486105
Iteration 83, loss = 0.44416076
Iteration 84, loss = 0.44352706
Iteration 85, loss = 0.44409830
Iteration 86, loss = 0.44399294
Iteration 87, loss = 0.44380457
```

```
Iteration 88, loss = 0.44298442
Iteration 89, loss = 0.44215324
Iteration 90, loss = 0.44197865
Iteration 91, loss = 0.44198011
Iteration 92, loss = 0.44171987
Iteration 93, loss = 0.44135163
Iteration 94, loss = 0.44140965
Iteration 95, loss = 0.44103810
Iteration 96, loss = 0.44117058
Iteration 97, loss = 0.44065262
Iteration 98, loss = 0.44015953
Iteration 99, loss = 0.44022270
Iteration 100, loss = 0.43978396
Iteration 101, loss = 0.43974260
Iteration 102, loss = 0.43937353
Iteration 103, loss = 0.43925473
Iteration 104, loss = 0.43970536
Iteration 105, loss = 0.43854119
Iteration 106, loss = 0.43822631
Iteration 107, loss = 0.43810104
Iteration 108, loss = 0.43841383
Iteration 109, loss = 0.43805760
Iteration 110, loss = 0.43801491
Iteration 111, loss = 0.43737289
Iteration 112, loss = 0.43754122
Iteration 113, loss = 0.43662769
Iteration 114, loss = 0.43658398
Iteration 115, loss = 0.43667094
Iteration 116, loss = 0.43641354
Iteration 117, loss = 0.43608833
Iteration 118, loss = 0.43573625
Iteration 119, loss = 0.43587482
Iteration 120, loss = 0.43506544
Iteration 121, loss = 0.43521825
Iteration 122, loss = 0.43471576
Iteration 123, loss = 0.43463147
Iteration 124, loss = 0.43419423
Iteration 125, loss = 0.43403035
Iteration 126, loss = 0.43479804
Iteration 127, loss = 0.43387973
Iteration 128, loss = 0.43355869
Iteration 129, loss = 0.43378538
Iteration 130, loss = 0.43334619
Iteration 131, loss = 0.43276215
Iteration 132, loss = 0.43278910
Iteration 133, loss = 0.43267042
Iteration 134, loss = 0.43271865
Iteration 135, loss = 0.43171046
Iteration 136, loss = 0.43199066
Iteration 137, loss = 0.43208845
Iteration 138, loss = 0.43182657
Iteration 139, loss = 0.43232819
Iteration 140, loss = 0.43121268
Iteration 141, loss = 0.43264928
Iteration 142, loss = 0.43040201
Iteration 143, loss = 0.43085587
Iteration 144, loss = 0.43130680
Iteration 145, loss = 0.43024194
Iteration 146, loss = 0.43039951
Iteration 147, loss = 0.43035921
Iteration 148, loss = 0.43071045
```

```
Iteration 149, loss = 0.42995965
Iteration 150, loss = 0.42942339
Iteration 151, loss = 0.42966885
Iteration 152, loss = 0.42932584
Iteration 153, loss = 0.42901288
Iteration 154, loss = 0.42882480
Iteration 155, loss = 0.42886646
Iteration 156, loss = 0.42859859
Iteration 157, loss = 0.42908251
Iteration 158, loss = 0.42874710
Iteration 159, loss = 0.42814192
Iteration 160, loss = 0.42843201
Iteration 161, loss = 0.42855631
Iteration 162, loss = 0.42855426
Iteration 163, loss = 0.42696147
Iteration 164, loss = 0.42727749
Iteration 165, loss = 0.42697696
Iteration 166, loss = 0.42674193
Iteration 167, loss = 0.42643749
Iteration 168, loss = 0.42630593
Iteration 169, loss = 0.42662264
Iteration 170, loss = 0.42717325
Iteration 171, loss = 0.42569140
Iteration 172, loss = 0.42532144
Iteration 173, loss = 0.42577753
Iteration 174, loss = 0.42552930
Iteration 175, loss = 0.42586952
Iteration 176, loss = 0.42554147
Iteration 177, loss = 0.42470873
Iteration 178, loss = 0.42475845
Iteration 179, loss = 0.42513437
Iteration 180, loss = 0.42403301
Iteration 181, loss = 0.42414081
Iteration 182, loss = 0.42470188
Iteration 183, loss = 0.42509917
Iteration 184, loss = 0.42446442
Iteration 185, loss = 0.42368511
Iteration 186, loss = 0.42332434
Iteration 187, loss = 0.42311256
Iteration 188, loss = 0.42306618
Iteration 189, loss = 0.42278511
Iteration 190, loss = 0.42289810
Iteration 191, loss = 0.42344702
Iteration 192, loss = 0.42299090
Iteration 193, loss = 0.42242447
Iteration 194, loss = 0.42347282
Iteration 195, loss = 0.42186149
Iteration 196, loss = 0.42167821
Iteration 197, loss = 0.42185271
Iteration 198, loss = 0.42136653
Iteration 199, loss = 0.42157584
Iteration 200, loss = 0.42147068
Iteration 201, loss = 0.42258057
Iteration 202, loss = 0.42131225
Iteration 203, loss = 0.42122896
Iteration 204, loss = 0.42132125
Iteration 205, loss = 0.42148822
Iteration 206, loss = 0.42022704
Iteration 207, loss = 0.42098668
Iteration 208, loss = 0.42006821
Iteration 209, loss = 0.42144114
```

```
Iteration 210, loss = 0.42003832
Iteration 211, loss = 0.42104032
Iteration 212, loss = 0.41998939
Iteration 213, loss = 0.42121144
Iteration 214, loss = 0.41864141
Iteration 215, loss = 0.42001166
Iteration 216, loss = 0.42061605
Iteration 217, loss = 0.41938790
Iteration 218, loss = 0.41856971
Iteration 219, loss = 0.41846780
Iteration 220, loss = 0.41999468
Iteration 221, loss = 0.41917908
Iteration 222, loss = 0.41861830
Iteration 223, loss = 0.41853747
Iteration 224, loss = 0.41814245
Iteration 225, loss = 0.41827002
Iteration 226, loss = 0.41774427
Iteration 227, loss = 0.41871441
Iteration 228, loss = 0.41801109
Iteration 229, loss = 0.41772846
Iteration 230, loss = 0.41782089
Iteration 231, loss = 0.41724281
Iteration 232, loss = 0.41741487
Iteration 233, loss = 0.41682702
Iteration 234, loss = 0.41721905
Iteration 235, loss = 0.41653724
Iteration 236, loss = 0.41766426
Iteration 237, loss = 0.41719458
Iteration 238, loss = 0.41661931
Iteration 239, loss = 0.41666096
Iteration 240, loss = 0.41589630
Iteration 241, loss = 0.41638575
Iteration 242, loss = 0.41622348
Iteration 243, loss = 0.41636298
Iteration 244, loss = 0.41568652
Iteration 245, loss = 0.41706053
Iteration 246, loss = 0.41483012
Iteration 247, loss = 0.41593031
Iteration 248, loss = 0.41547696
Iteration 249, loss = 0.41514489
Iteration 250, loss = 0.41491136
Iteration 251, loss = 0.41531816
Iteration 252, loss = 0.41407400
Iteration 253, loss = 0.41413580
Iteration 254, loss = 0.41535295
Iteration 255, loss = 0.41469016
Iteration 256, loss = 0.41429748
Iteration 257, loss = 0.41352715
Iteration 258, loss = 0.41478522
Iteration 259, loss = 0.41422488
Iteration 260, loss = 0.41418525
Iteration 261, loss = 0.41399202
Iteration 262, loss = 0.41307145
Iteration 263, loss = 0.41346286
Iteration 264, loss = 0.41384122
Iteration 265, loss = 0.41336652
Iteration 266, loss = 0.41259963
Iteration 267, loss = 0.41298332
Iteration 268, loss = 0.41267585
Iteration 269, loss = 0.41277444
Iteration 270, loss = 0.41342491
```

```
Iteration 271, loss = 0.41202280
Iteration 272, loss = 0.41259404
Iteration 273, loss = 0.41193007
Iteration 274, loss = 0.41259949
Iteration 275, loss = 0.41260761
Iteration 276, loss = 0.41194109
Iteration 277, loss = 0.41213796
Iteration 278, loss = 0.41149294
Iteration 279, loss = 0.41193078
Iteration 280, loss = 0.41127470
Iteration 281, loss = 0.41193214
Iteration 282, loss = 0.41165808
Iteration 283, loss = 0.41145745
Iteration 284, loss = 0.41147578
Iteration 285, loss = 0.41161824
Iteration 286, loss = 0.41114649
Iteration 287, loss = 0.41067657
Iteration 288, loss = 0.41049034
Iteration 289, loss = 0.41061704
Iteration 290, loss = 0.41069769
Iteration 291, loss = 0.41070909
Iteration 292, loss = 0.41073182
Iteration 293, loss = 0.41102261
Iteration 294, loss = 0.41000430
Iteration 295, loss = 0.41060282
Iteration 296, loss = 0.41041770
Iteration 297, loss = 0.41067628
Iteration 298, loss = 0.40981059
Iteration 299, loss = 0.40975504
Iteration 300, loss = 0.40963940
Iteration 1, loss = 0.67755459
Iteration 2, loss = 0.59503581
Iteration 3, loss = 0.54472907

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 4, loss = 0.51738436
Iteration 5, loss = 0.50404487
Iteration 6, loss = 0.49705721
Iteration 7, loss = 0.49371258
Iteration 8, loss = 0.49086631
Iteration 9, loss = 0.48903754
Iteration 10, loss = 0.48770665
Iteration 11, loss = 0.48607752
Iteration 12, loss = 0.48465175
Iteration 13, loss = 0.48371692
Iteration 14, loss = 0.48204856
Iteration 15, loss = 0.48109837
Iteration 16, loss = 0.48051585
Iteration 17, loss = 0.47965213
Iteration 18, loss = 0.47844453
Iteration 19, loss = 0.47753729
Iteration 20, loss = 0.47674696
Iteration 21, loss = 0.47583513
Iteration 22, loss = 0.47526513
Iteration 23, loss = 0.47454465
Iteration 24, loss = 0.47363853
Iteration 25, loss = 0.47319107
Iteration 26, loss = 0.47257351
Iteration 27, loss = 0.47166570
Iteration 28, loss = 0.47113631
Iteration 29, loss = 0.47104795
Iteration 30, loss = 0.46980836
Iteration 31, loss = 0.46927637
Iteration 32, loss = 0.46864431
Iteration 33, loss = 0.46810706
Iteration 34, loss = 0.46733507
Iteration 35, loss = 0.46677445
Iteration 36, loss = 0.46680387
Iteration 37, loss = 0.46560305
Iteration 38, loss = 0.46527700
Iteration 39, loss = 0.46458559
Iteration 40, loss = 0.46410201
Iteration 41, loss = 0.46386483
Iteration 42, loss = 0.46304079
Iteration 43, loss = 0.46268576
Iteration 44, loss = 0.46225374
Iteration 45, loss = 0.46147940
Iteration 46, loss = 0.46082801
Iteration 47, loss = 0.46075681
Iteration 48, loss = 0.45995058
Iteration 49, loss = 0.45975191
Iteration 50, loss = 0.45894209
Iteration 51, loss = 0.45850870
Iteration 52, loss = 0.45801791
Iteration 53, loss = 0.45771933
Iteration 54, loss = 0.45725944
Iteration 55, loss = 0.45720517
Iteration 56, loss = 0.45636491
Iteration 57, loss = 0.45632244
Iteration 58, loss = 0.45535649
Iteration 59, loss = 0.45501365
Iteration 60, loss = 0.45493988
Iteration 61, loss = 0.45454139
Iteration 62, loss = 0.45386452
Iteration 63, loss = 0.45380708
Iteration 64, loss = 0.45312647
```

```
Iteration 65, loss = 0.45292757
Iteration 66, loss = 0.45232958
Iteration 67, loss = 0.45200031
Iteration 68, loss = 0.45159027
Iteration 69, loss = 0.45135458
Iteration 70, loss = 0.45159110
Iteration 71, loss = 0.45117413
Iteration 72, loss = 0.45055157
Iteration 73, loss = 0.44990104
Iteration 74, loss = 0.44990742
Iteration 75, loss = 0.44988259
Iteration 76, loss = 0.44908584
Iteration 77, loss = 0.44885949
Iteration 78, loss = 0.44885457
Iteration 79, loss = 0.44915776
Iteration 80, loss = 0.44815955
Iteration 81, loss = 0.44766009
Iteration 82, loss = 0.44721136
Iteration 83, loss = 0.44707636
Iteration 84, loss = 0.44733252
Iteration 85, loss = 0.44613182
Iteration 86, loss = 0.44591445
Iteration 87, loss = 0.44614802
Iteration 88, loss = 0.44511703
Iteration 89, loss = 0.44606061
Iteration 90, loss = 0.44547560
Iteration 91, loss = 0.44524463
Iteration 92, loss = 0.44431781
Iteration 93, loss = 0.44434355
Iteration 94, loss = 0.44365080
Iteration 95, loss = 0.44317750
Iteration 96, loss = 0.44364494
Iteration 97, loss = 0.44271927
Iteration 98, loss = 0.44238796
Iteration 99, loss = 0.44290539
Iteration 100, loss = 0.44189769
Iteration 101, loss = 0.44293422
Iteration 102, loss = 0.44186765
Iteration 103, loss = 0.44167499
Iteration 104, loss = 0.44145570
Iteration 105, loss = 0.44079922
Iteration 106, loss = 0.44161246
Iteration 107, loss = 0.44080161
Iteration 108, loss = 0.44054912
Iteration 109, loss = 0.44080920
Iteration 110, loss = 0.44046512
Iteration 111, loss = 0.43951646
Iteration 112, loss = 0.43928794
Iteration 113, loss = 0.43883153
Iteration 114, loss = 0.43877085
Iteration 115, loss = 0.43837599
Iteration 116, loss = 0.43810489
Iteration 117, loss = 0.43886246
Iteration 118, loss = 0.43810726
Iteration 119, loss = 0.43726355
Iteration 120, loss = 0.43692776
Iteration 121, loss = 0.43670924
Iteration 122, loss = 0.43721160
Iteration 123, loss = 0.43689090
Iteration 124, loss = 0.43618761
Iteration 125, loss = 0.43641262
```

```
Iteration 126, loss = 0.43588344
Iteration 127, loss = 0.43543591
Iteration 128, loss = 0.43508550
Iteration 129, loss = 0.43511010
Iteration 130, loss = 0.43457999
Iteration 131, loss = 0.43442643
Iteration 132, loss = 0.43430631
Iteration 133, loss = 0.43403136
Iteration 134, loss = 0.43347251
Iteration 135, loss = 0.43345420
Iteration 136, loss = 0.43382274
Iteration 137, loss = 0.43332600
Iteration 138, loss = 0.43398130
Iteration 139, loss = 0.43303517
Iteration 140, loss = 0.43320561
Iteration 141, loss = 0.43261402
Iteration 142, loss = 0.43185063
Iteration 143, loss = 0.43236045
Iteration 144, loss = 0.43141470
Iteration 145, loss = 0.43192417
Iteration 146, loss = 0.43120470
Iteration 147, loss = 0.43132775
Iteration 148, loss = 0.43068189
Iteration 149, loss = 0.43081834
Iteration 150, loss = 0.43134396
Iteration 151, loss = 0.43061951
Iteration 152, loss = 0.43054067
Iteration 153, loss = 0.42939172
Iteration 154, loss = 0.43000863
Iteration 155, loss = 0.42965619
Iteration 156, loss = 0.42971580
Iteration 157, loss = 0.42925894
Iteration 158, loss = 0.42935300
Iteration 159, loss = 0.42939577
Iteration 160, loss = 0.42924207
Iteration 161, loss = 0.42868529
Iteration 162, loss = 0.42862802
Iteration 163, loss = 0.42794199
Iteration 164, loss = 0.42824165
Iteration 165, loss = 0.42717709
Iteration 166, loss = 0.42776233
Iteration 167, loss = 0.42673829
Iteration 168, loss = 0.42716456
Iteration 169, loss = 0.42704380
Iteration 170, loss = 0.42716478
Iteration 171, loss = 0.42801844
Iteration 172, loss = 0.42656134
Iteration 173, loss = 0.42604106
Iteration 174, loss = 0.42551330
Iteration 175, loss = 0.42549201
Iteration 176, loss = 0.42580836
Iteration 177, loss = 0.42532193
Iteration 178, loss = 0.42502644
Iteration 179, loss = 0.42511636
Iteration 180, loss = 0.42524775
Iteration 181, loss = 0.42427552
Iteration 182, loss = 0.42444728
Iteration 183, loss = 0.42387706
Iteration 184, loss = 0.42367255
Iteration 185, loss = 0.42401706
Iteration 186, loss = 0.42334387
```

```
Iteration 187, loss = 0.42467017
Iteration 188, loss = 0.42265847
Iteration 189, loss = 0.42421168
Iteration 190, loss = 0.42286849
Iteration 191, loss = 0.42288733
Iteration 192, loss = 0.42233435
Iteration 193, loss = 0.42259107
Iteration 194, loss = 0.42202334
Iteration 195, loss = 0.42205500
Iteration 196, loss = 0.42127547
Iteration 197, loss = 0.42196725
Iteration 198, loss = 0.42172707
Iteration 199, loss = 0.42151384
Iteration 200, loss = 0.42146550
Iteration 201, loss = 0.42143465
Iteration 202, loss = 0.42108041
Iteration 203, loss = 0.42056703
Iteration 204, loss = 0.42105735
Iteration 205, loss = 0.42065691
Iteration 206, loss = 0.42053970
Iteration 207, loss = 0.42023089
Iteration 208, loss = 0.42130225
Iteration 209, loss = 0.42137118
Iteration 210, loss = 0.42032732
Iteration 211, loss = 0.41904685
Iteration 212, loss = 0.42018212
Iteration 213, loss = 0.41875676
Iteration 214, loss = 0.41971414
Iteration 215, loss = 0.41782936
Iteration 216, loss = 0.41898551
Iteration 217, loss = 0.41872462
Iteration 218, loss = 0.41824654
Iteration 219, loss = 0.41818083
Iteration 220, loss = 0.41758931
Iteration 221, loss = 0.41742751
Iteration 222, loss = 0.41858889
Iteration 223, loss = 0.41783954
Iteration 224, loss = 0.41717166
Iteration 225, loss = 0.41637024
Iteration 226, loss = 0.41793102
Iteration 227, loss = 0.41707357
Iteration 228, loss = 0.41763396
Iteration 229, loss = 0.41629121
Iteration 230, loss = 0.41638596
Iteration 231, loss = 0.41645218
Iteration 232, loss = 0.41657944
Iteration 233, loss = 0.41561604
Iteration 234, loss = 0.41652518
Iteration 235, loss = 0.41685323
Iteration 236, loss = 0.41566737
Iteration 237, loss = 0.41571496
Iteration 238, loss = 0.41542081
Iteration 239, loss = 0.41527651
Iteration 240, loss = 0.41601265
Iteration 241, loss = 0.41615130
Iteration 242, loss = 0.41403663
Iteration 243, loss = 0.41456951
Iteration 244, loss = 0.41598256
Iteration 245, loss = 0.41298966
Iteration 246, loss = 0.41444800
Iteration 247, loss = 0.41323350
```

```
Iteration 248, loss = 0.41333942
Iteration 249, loss = 0.41291926
Iteration 250, loss = 0.41286688
Iteration 251, loss = 0.41339747
Iteration 252, loss = 0.41188107
Iteration 253, loss = 0.41270944
Iteration 254, loss = 0.41188804
Iteration 255, loss = 0.41257289
Iteration 256, loss = 0.41199247
Iteration 257, loss = 0.41294466
Iteration 258, loss = 0.41161388
Iteration 259, loss = 0.41150812
Iteration 260, loss = 0.41132117
Iteration 261, loss = 0.41124892
Iteration 262, loss = 0.41052716
Iteration 263, loss = 0.41246568
Iteration 264, loss = 0.41030832
Iteration 265, loss = 0.41108521
Iteration 266, loss = 0.41096708
Iteration 267, loss = 0.41055450
Iteration 268, loss = 0.41000236
Iteration 269, loss = 0.41011824
Iteration 270, loss = 0.41021699
Iteration 271, loss = 0.41095942
Iteration 272, loss = 0.40938046
Iteration 273, loss = 0.40907059
Iteration 274, loss = 0.41011676
Iteration 275, loss = 0.40910643
Iteration 276, loss = 0.40911822
Iteration 277, loss = 0.40854559
Iteration 278, loss = 0.40837835
Iteration 279, loss = 0.40824986
Iteration 280, loss = 0.40850131
Iteration 281, loss = 0.40826261
Iteration 282, loss = 0.40769230
Iteration 283, loss = 0.40761799
Iteration 284, loss = 0.40738713
Iteration 285, loss = 0.40783282
Iteration 286, loss = 0.40888912
Iteration 287, loss = 0.40734724
Iteration 288, loss = 0.40712479
Iteration 289, loss = 0.40823243
Iteration 290, loss = 0.40700799
Iteration 291, loss = 0.40698478
Iteration 292, loss = 0.40893951
Iteration 293, loss = 0.40729376
Iteration 294, loss = 0.40812703
Iteration 295, loss = 0.40711640
Iteration 296, loss = 0.40648879
Iteration 297, loss = 0.40649125
Iteration 298, loss = 0.40578128
Iteration 299, loss = 0.40575997
Iteration 300, loss = 0.40598228
Iteration 1, loss = 0.62259431
Iteration 2, loss = 0.54894778
Iteration 3, loss = 0.51566751
Iteration 4, loss = 0.50001843
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 5, loss = 0.49463503
Iteration 6, loss = 0.49236299
Iteration 7, loss = 0.49009066
Iteration 8, loss = 0.48841763
Iteration 9, loss = 0.48728535
Iteration 10, loss = 0.48574612
Iteration 11, loss = 0.48420210
Iteration 12, loss = 0.48335490
Iteration 13, loss = 0.48214447
Iteration 14, loss = 0.48120675
Iteration 15, loss = 0.48010069
Iteration 16, loss = 0.47908842
Iteration 17, loss = 0.47861701
Iteration 18, loss = 0.47766380
Iteration 19, loss = 0.47662287
Iteration 20, loss = 0.47602265
Iteration 21, loss = 0.47512318
Iteration 22, loss = 0.47440001
Iteration 23, loss = 0.47361313
Iteration 24, loss = 0.47325448
Iteration 25, loss = 0.47246430
Iteration 26, loss = 0.47164413
Iteration 27, loss = 0.47096265
Iteration 28, loss = 0.47040531
Iteration 29, loss = 0.46941590
Iteration 30, loss = 0.46928066
Iteration 31, loss = 0.46810426
Iteration 32, loss = 0.46750919
Iteration 33, loss = 0.46674889
Iteration 34, loss = 0.46651316
Iteration 35, loss = 0.46569772
Iteration 36, loss = 0.46510038
Iteration 37, loss = 0.46480757
Iteration 38, loss = 0.46406489
Iteration 39, loss = 0.46344414
Iteration 40, loss = 0.46288813
Iteration 41, loss = 0.46251860
Iteration 42, loss = 0.46147919
Iteration 43, loss = 0.46149839
Iteration 44, loss = 0.46151141
Iteration 45, loss = 0.46062160
Iteration 46, loss = 0.45995333
Iteration 47, loss = 0.45938112
Iteration 48, loss = 0.45902813
Iteration 49, loss = 0.45833602
Iteration 50, loss = 0.45809705
Iteration 51, loss = 0.45768078
Iteration 52, loss = 0.45715470
Iteration 53, loss = 0.45701008
Iteration 54, loss = 0.45640024
Iteration 55, loss = 0.45607053
Iteration 56, loss = 0.45556838
Iteration 57, loss = 0.45530458
Iteration 58, loss = 0.45486234
Iteration 59, loss = 0.45439971
Iteration 60, loss = 0.45398013
Iteration 61, loss = 0.45352432
Iteration 62, loss = 0.45348414
Iteration 63, loss = 0.45276859
Iteration 64, loss = 0.45343970
Iteration 65, loss = 0.45246088
```

```
Iteration 66, loss = 0.45184573
Iteration 67, loss = 0.45140951
Iteration 68, loss = 0.45152449
Iteration 69, loss = 0.45085510
Iteration 70, loss = 0.45036689
Iteration 71, loss = 0.45007387
Iteration 72, loss = 0.44987574
Iteration 73, loss = 0.44932572
Iteration 74, loss = 0.44885884
Iteration 75, loss = 0.44882864
Iteration 76, loss = 0.44881476
Iteration 77, loss = 0.44806191
Iteration 78, loss = 0.44815989
Iteration 79, loss = 0.44728524
Iteration 80, loss = 0.44763497
Iteration 81, loss = 0.44672861
Iteration 82, loss = 0.44685840
Iteration 83, loss = 0.44615054
Iteration 84, loss = 0.44569959
Iteration 85, loss = 0.44538346
Iteration 86, loss = 0.44647031
Iteration 87, loss = 0.44500182
Iteration 88, loss = 0.44571283
Iteration 89, loss = 0.44444858
Iteration 90, loss = 0.44413611
Iteration 91, loss = 0.44430367
Iteration 92, loss = 0.44352636
Iteration 93, loss = 0.44309583
Iteration 94, loss = 0.44375274
Iteration 95, loss = 0.44299787
Iteration 96, loss = 0.44294895
Iteration 97, loss = 0.44250915
Iteration 98, loss = 0.44203915
Iteration 99, loss = 0.44196094
Iteration 100, loss = 0.44164451
Iteration 101, loss = 0.44090102
Iteration 102, loss = 0.44177166
Iteration 103, loss = 0.44053777
Iteration 104, loss = 0.44053185
Iteration 105, loss = 0.44024042
Iteration 106, loss = 0.44048331
Iteration 107, loss = 0.44011567
Iteration 108, loss = 0.43971416
Iteration 109, loss = 0.43869890
Iteration 110, loss = 0.43888637
Iteration 111, loss = 0.43874619
Iteration 112, loss = 0.44092960
Iteration 113, loss = 0.43831010
Iteration 114, loss = 0.43892825
Iteration 115, loss = 0.43813383
Iteration 116, loss = 0.43708556
Iteration 117, loss = 0.43718815
Iteration 118, loss = 0.43759299
Iteration 119, loss = 0.43710136
Iteration 120, loss = 0.43685931
Iteration 121, loss = 0.43607095
Iteration 122, loss = 0.43620442
Iteration 123, loss = 0.43576600
Iteration 124, loss = 0.43657042
Iteration 125, loss = 0.43727623
Iteration 126, loss = 0.43597794
```

```
Iteration 127, loss = 0.43529250
Iteration 128, loss = 0.43460823
Iteration 129, loss = 0.43441503
Iteration 130, loss = 0.43418074
Iteration 131, loss = 0.43396091
Iteration 132, loss = 0.43404726
Iteration 133, loss = 0.43359148
Iteration 134, loss = 0.43363295
Iteration 135, loss = 0.43382226
Iteration 136, loss = 0.43354150
Iteration 137, loss = 0.43299421
Iteration 138, loss = 0.43325421
Iteration 139, loss = 0.43246505
Iteration 140, loss = 0.43178364
Iteration 141, loss = 0.43197530
Iteration 142, loss = 0.43154155
Iteration 143, loss = 0.43138259
Iteration 144, loss = 0.43175297
Iteration 145, loss = 0.43183836
Iteration 146, loss = 0.43104290
Iteration 147, loss = 0.43183383
Iteration 148, loss = 0.43044081
Iteration 149, loss = 0.43066664
Iteration 150, loss = 0.43054133
Iteration 151, loss = 0.43079657
Iteration 152, loss = 0.42959709
Iteration 153, loss = 0.42941785
Iteration 154, loss = 0.42952100
Iteration 155, loss = 0.42897661
Iteration 156, loss = 0.42892753
Iteration 157, loss = 0.42859200
Iteration 158, loss = 0.42917014
Iteration 159, loss = 0.42910374
Iteration 160, loss = 0.42808609
Iteration 161, loss = 0.42760175
Iteration 162, loss = 0.42814130
Iteration 163, loss = 0.42745504
Iteration 164, loss = 0.42779332
Iteration 165, loss = 0.42708354
Iteration 166, loss = 0.42776317
Iteration 167, loss = 0.42755189
Iteration 168, loss = 0.42686277
Iteration 169, loss = 0.42733397
Iteration 170, loss = 0.42742995
Iteration 171, loss = 0.42701306
Iteration 172, loss = 0.42746276
Iteration 173, loss = 0.42552602
Iteration 174, loss = 0.42606386
Iteration 175, loss = 0.42500139
Iteration 176, loss = 0.42591113
Iteration 177, loss = 0.42496117
Iteration 178, loss = 0.42578786
Iteration 179, loss = 0.42464237
Iteration 180, loss = 0.42460402
Iteration 181, loss = 0.42479607
Iteration 182, loss = 0.42442972
Iteration 183, loss = 0.42462224
Iteration 184, loss = 0.42465211
Iteration 185, loss = 0.42407544
Iteration 186, loss = 0.42330210
Iteration 187, loss = 0.42358714
```

```
Iteration 188, loss = 0.42296420
Iteration 189, loss = 0.42342821
Iteration 190, loss = 0.42291901
Iteration 191, loss = 0.42343934
Iteration 192, loss = 0.42244321
Iteration 193, loss = 0.42351694
Iteration 194, loss = 0.42388568
Iteration 195, loss = 0.42177353
Iteration 196, loss = 0.42244566
Iteration 197, loss = 0.42294526
Iteration 198, loss = 0.42212496
Iteration 199, loss = 0.42123578
Iteration 200, loss = 0.42096387
Iteration 201, loss = 0.42143422
Iteration 202, loss = 0.42042879
Iteration 203, loss = 0.42088333
Iteration 204, loss = 0.42148641
Iteration 205, loss = 0.42053224
Iteration 206, loss = 0.42053900
Iteration 207, loss = 0.42018978
Iteration 208, loss = 0.42054712
Iteration 209, loss = 0.41989513
Iteration 210, loss = 0.41996285
Iteration 211, loss = 0.41962304
Iteration 212, loss = 0.41995998
Iteration 213, loss = 0.41935564
Iteration 214, loss = 0.41894521
Iteration 215, loss = 0.41889311
Iteration 216, loss = 0.41883342
Iteration 217, loss = 0.42077602
Iteration 218, loss = 0.41898500
Iteration 219, loss = 0.41870937
Iteration 220, loss = 0.41968395
Iteration 221, loss = 0.41751184
Iteration 222, loss = 0.41758109
Iteration 223, loss = 0.41740922
Iteration 224, loss = 0.41746997
Iteration 225, loss = 0.41764623
Iteration 226, loss = 0.41691411
Iteration 227, loss = 0.41863153
Iteration 228, loss = 0.41821411
Iteration 229, loss = 0.41663789
Iteration 230, loss = 0.41759840
Iteration 231, loss = 0.41647455
Iteration 232, loss = 0.41659695
Iteration 233, loss = 0.41715755
Iteration 234, loss = 0.41554776
Iteration 235, loss = 0.41728450
Iteration 236, loss = 0.41573996
Iteration 237, loss = 0.41643140
Iteration 238, loss = 0.41657170
Iteration 239, loss = 0.41638502
Iteration 240, loss = 0.41591218
Iteration 241, loss = 0.41560497
Iteration 242, loss = 0.41532781
Iteration 243, loss = 0.41498276
Iteration 244, loss = 0.41497922
Iteration 245, loss = 0.41477997
Iteration 246, loss = 0.41480615
Iteration 247, loss = 0.41406982
Iteration 248, loss = 0.41497500
```

```
Iteration 249, loss = 0.41400477
Iteration 250, loss = 0.41406379
Iteration 251, loss = 0.41383069
Iteration 252, loss = 0.41371055
Iteration 253, loss = 0.41360837
Iteration 254, loss = 0.41368322
Iteration 255, loss = 0.41302495
Iteration 256, loss = 0.41296752
Iteration 257, loss = 0.41303710
Iteration 258, loss = 0.41329353
Iteration 259, loss = 0.41310196
Iteration 260, loss = 0.41334334
Iteration 261, loss = 0.41273739
Iteration 262, loss = 0.41301150
Iteration 263, loss = 0.41215485
Iteration 264, loss = 0.41280236
Iteration 265, loss = 0.41177360
Iteration 266, loss = 0.41188284
Iteration 267, loss = 0.41406938
Iteration 268, loss = 0.41303662
Iteration 269, loss = 0.41118489
Iteration 270, loss = 0.41162680
Iteration 271, loss = 0.41204779
Iteration 272, loss = 0.41237349
Iteration 273, loss = 0.41062417
Iteration 274, loss = 0.41093457
Iteration 275, loss = 0.41061756
Iteration 276, loss = 0.41050252
Iteration 277, loss = 0.41052620
Iteration 278, loss = 0.41040049
Iteration 279, loss = 0.41016830
Iteration 280, loss = 0.41032701
Iteration 281, loss = 0.41074775
Iteration 282, loss = 0.40979021
Iteration 283, loss = 0.41292702
Iteration 284, loss = 0.41202562
Iteration 285, loss = 0.41088527
Iteration 286, loss = 0.40911591
Iteration 287, loss = 0.40881658
Iteration 288, loss = 0.40885228
Iteration 289, loss = 0.40935070
Iteration 290, loss = 0.41155142
Iteration 291, loss = 0.40826178
Iteration 292, loss = 0.40846693
Iteration 293, loss = 0.40899991
Iteration 294, loss = 0.40798498
Iteration 295, loss = 0.40887715
Iteration 296, loss = 0.40737321
Iteration 297, loss = 0.40837431
Iteration 298, loss = 0.40824063
Iteration 299, loss = 0.40766063
Iteration 300, loss = 0.40806897
Iteration 1, loss = 0.69105230
Iteration 2, loss = 0.60506783
Iteration 3, loss = 0.55050626
Iteration 4, loss = 0.51929946
Iteration 5, loss = 0.50090024
Iteration 6, loss = 0.49162798
Iteration 7, loss = 0.48639816
Iteration 8, loss = 0.48323471
Iteration 9, loss = 0.48154331
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (300) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 10, loss = 0.48007175
Iteration 11, loss = 0.47884530
Iteration 12, loss = 0.47773123
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.61649757
Iteration 2, loss = 0.55791777
Iteration 3, loss = 0.52316519
Iteration 4, loss = 0.50631783
Iteration 5, loss = 0.49845502
Iteration 6, loss = 0.49392022
Iteration 7, loss = 0.49134925
Iteration 8, loss = 0.48987722
Iteration 9, loss = 0.48797077
Iteration 10, loss = 0.48668115
Iteration 11, loss = 0.48547811
Iteration 12, loss = 0.48434990
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.65441362
Iteration 2, loss = 0.57845330
Iteration 3, loss = 0.53596807
Iteration 4, loss = 0.51201911
Iteration 5, loss = 0.49887935
Iteration 6, loss = 0.49182107
Iteration 7, loss = 0.48809982
Iteration 8, loss = 0.48663552
Iteration 9, loss = 0.48515239
Iteration 10, loss = 0.48406585
Iteration 11, loss = 0.48318366
Iteration 12, loss = 0.48202288
Training loss did not improve more than tol=0.100000 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.68959871
Iteration 2, loss = 0.60447535
Iteration 3, loss = 0.55242003
Iteration 4, loss = 0.52246310
Iteration 5, loss = 0.50508820
Iteration 6, loss = 0.49386288
Iteration 7, loss = 0.48725060
Iteration 8, loss = 0.48418627
Iteration 9, loss = 0.48122351
Iteration 10, loss = 0.47999676
Iteration 11, loss = 0.47864362
Iteration 12, loss = 0.47764839
Iteration 13, loss = 0.47651802
Iteration 14, loss = 0.47526795
Iteration 15, loss = 0.47400905
Iteration 16, loss = 0.47315271
Iteration 17, loss = 0.47240530
Iteration 18, loss = 0.47136647
Iteration 19, loss = 0.47072978
Iteration 20, loss = 0.46978529
Iteration 21, loss = 0.46958482
Iteration 22, loss = 0.46867203
Iteration 23, loss = 0.46750342
Iteration 24, loss = 0.46707011
Iteration 25, loss = 0.46643900
Iteration 26, loss = 0.46601083
Iteration 27, loss = 0.46527771
Iteration 28, loss = 0.46491170
```

```
Iteration 29, loss = 0.46395719
Iteration 30, loss = 0.46335473
Iteration 31, loss = 0.46299323
Iteration 32, loss = 0.46232403
Iteration 33, loss = 0.46187975
Iteration 34, loss = 0.46159072
Iteration 35, loss = 0.46102809
Iteration 36, loss = 0.46023243
Iteration 37, loss = 0.45984126
Iteration 38, loss = 0.45929128
Iteration 39, loss = 0.45863504
Iteration 40, loss = 0.45827830
Iteration 41, loss = 0.45795880
Iteration 42, loss = 0.45736465
Iteration 43, loss = 0.45676496
Iteration 44, loss = 0.45627962
Iteration 45, loss = 0.45611797
Iteration 46, loss = 0.45545689
Iteration 47, loss = 0.45490812
Iteration 48, loss = 0.45494518
Iteration 49, loss = 0.45403457
Iteration 50, loss = 0.45399250
Iteration 51, loss = 0.45347584
Iteration 52, loss = 0.45319586
Iteration 53, loss = 0.45296946
Iteration 54, loss = 0.45199716
Iteration 55, loss = 0.45240040
Iteration 56, loss = 0.45119773
Iteration 57, loss = 0.45155501
Iteration 58, loss = 0.45111673
Iteration 59, loss = 0.45040039
Iteration 60, loss = 0.45000592
Iteration 61, loss = 0.44976468
Iteration 62, loss = 0.44987722
Iteration 63, loss = 0.44929057
Iteration 64, loss = 0.44888430
Iteration 65, loss = 0.44869873
Iteration 66, loss = 0.44875315
Iteration 67, loss = 0.44760828
Iteration 68, loss = 0.44741339
Iteration 69, loss = 0.44757200
Iteration 70, loss = 0.44717714
Iteration 71, loss = 0.44674309
Iteration 72, loss = 0.44585451
Iteration 73, loss = 0.44553993
Iteration 74, loss = 0.44558435
Iteration 75, loss = 0.44547869
Iteration 76, loss = 0.44495931
Iteration 77, loss = 0.44456973
Iteration 78, loss = 0.44468491
Iteration 79, loss = 0.44410335
Iteration 80, loss = 0.44374671
Iteration 81, loss = 0.44362981
Iteration 82, loss = 0.44324120
Iteration 83, loss = 0.44334866
Iteration 84, loss = 0.44324689
Iteration 85, loss = 0.44233113
Iteration 86, loss = 0.44223112
Iteration 87, loss = 0.44191075
Iteration 88, loss = 0.44121862
Iteration 89, loss = 0.44119149
```

```
Iteration 90, loss = 0.44162668
Iteration 91, loss = 0.44091039
Iteration 92, loss = 0.44020646
Iteration 93, loss = 0.44029329
Iteration 94, loss = 0.44011542
Iteration 95, loss = 0.43992633
Iteration 96, loss = 0.44005477
Iteration 97, loss = 0.43921443
Iteration 98, loss = 0.43941358
Iteration 99, loss = 0.43887384
Iteration 100, loss = 0.43855969
Iteration 101, loss = 0.43786821
Iteration 102, loss = 0.43815617
Iteration 103, loss = 0.43825434
Iteration 104, loss = 0.43767826
Iteration 105, loss = 0.43707632
Iteration 106, loss = 0.43697547
Iteration 107, loss = 0.43705531
Iteration 108, loss = 0.43698659
Iteration 109, loss = 0.43630203
Iteration 110, loss = 0.43622759
Iteration 111, loss = 0.43579340
Iteration 112, loss = 0.43582319
Iteration 113, loss = 0.43606408
Iteration 114, loss = 0.43563560
Iteration 115, loss = 0.43509981
Iteration 116, loss = 0.43521716
Iteration 117, loss = 0.43476977
Iteration 118, loss = 0.43442563
Iteration 119, loss = 0.43397911
Iteration 120, loss = 0.43434148
Iteration 121, loss = 0.43435401
Iteration 122, loss = 0.43351984
Iteration 123, loss = 0.43386575
Iteration 124, loss = 0.43398164
Iteration 125, loss = 0.43313817
Iteration 126, loss = 0.43308007
Iteration 127, loss = 0.43268131
Iteration 128, loss = 0.43318018
Iteration 129, loss = 0.43245015
Iteration 130, loss = 0.43210129
Iteration 131, loss = 0.43173929
Iteration 132, loss = 0.43154926
Iteration 133, loss = 0.43189470
Iteration 134, loss = 0.43206986
Iteration 135, loss = 0.43134970
Iteration 136, loss = 0.43136910
Iteration 137, loss = 0.43077877
Iteration 138, loss = 0.43098552
Iteration 139, loss = 0.43081261
Iteration 140, loss = 0.43040618
Iteration 141, loss = 0.43006232
Iteration 142, loss = 0.42988608
Iteration 143, loss = 0.42968338
Iteration 144, loss = 0.42959576
Iteration 145, loss = 0.42876927
Iteration 146, loss = 0.42902559
Iteration 147, loss = 0.42872856
Iteration 148, loss = 0.42875970
Iteration 149, loss = 0.42908126
Iteration 150, loss = 0.42849471
```

```
Iteration 151, loss = 0.42847763
Iteration 152, loss = 0.42786631
Iteration 153, loss = 0.42799518
Iteration 154, loss = 0.42837790
Iteration 155, loss = 0.42782640
Iteration 156, loss = 0.42788784
Iteration 157, loss = 0.42766734
Iteration 158, loss = 0.42740402
Iteration 159, loss = 0.42752096
Iteration 160, loss = 0.42762265
Iteration 161, loss = 0.42667295
Iteration 162, loss = 0.42720852
Iteration 163, loss = 0.42684462
Iteration 164, loss = 0.42610729
Iteration 165, loss = 0.42550612
Iteration 166, loss = 0.42576033
Iteration 167, loss = 0.42629018
Iteration 168, loss = 0.42594763
Iteration 169, loss = 0.42560073
Iteration 170, loss = 0.42559723
Iteration 171, loss = 0.42476505
Iteration 172, loss = 0.42456765
Iteration 173, loss = 0.42579026
Iteration 174, loss = 0.42490952
Iteration 175, loss = 0.42518893
Iteration 176, loss = 0.42375476
Iteration 177, loss = 0.42370349
Iteration 178, loss = 0.42367207
Iteration 179, loss = 0.42376490
Iteration 180, loss = 0.42440441
Iteration 181, loss = 0.42301013
Iteration 182, loss = 0.42318762
Iteration 183, loss = 0.42344602
Iteration 184, loss = 0.42359695
Iteration 185, loss = 0.42355664
Iteration 186, loss = 0.42326579
Iteration 187, loss = 0.42227571
Iteration 188, loss = 0.42408618
Iteration 189, loss = 0.42338754
Iteration 190, loss = 0.42262685
Iteration 191, loss = 0.42196435
Iteration 192, loss = 0.42292019
Iteration 193, loss = 0.42202131
Iteration 194, loss = 0.42142095
Iteration 195, loss = 0.42151222
Iteration 196, loss = 0.42177972
Iteration 197, loss = 0.42132299
Iteration 198, loss = 0.42054328
Iteration 199, loss = 0.42117416
Iteration 200, loss = 0.42029476
Iteration 201, loss = 0.42066868
Iteration 202, loss = 0.42216356
Iteration 203, loss = 0.42012804
Iteration 204, loss = 0.42109073
Iteration 205, loss = 0.41988061
Iteration 206, loss = 0.42067259
Iteration 207, loss = 0.41933655
Iteration 208, loss = 0.41949962
Iteration 209, loss = 0.42015413
Iteration 210, loss = 0.41932831
Iteration 211, loss = 0.41920984
```

```
Iteration 212, loss = 0.41933523
Iteration 213, loss = 0.41886159
Iteration 214, loss = 0.41842008
Iteration 215, loss = 0.41846859
Iteration 216, loss = 0.41945648
Iteration 217, loss = 0.41780818
Iteration 218, loss = 0.41812849
Iteration 219, loss = 0.41843772
Iteration 220, loss = 0.41828542
Iteration 221, loss = 0.41873456
Iteration 222, loss = 0.41808413
Iteration 223, loss = 0.41756379
Iteration 224, loss = 0.41729233
Iteration 225, loss = 0.41715834
Iteration 226, loss = 0.41826174
Iteration 227, loss = 0.41667507
Iteration 228, loss = 0.41633990
Iteration 229, loss = 0.41670981
Iteration 230, loss = 0.41622501
Iteration 231, loss = 0.41621752
Iteration 232, loss = 0.41617812
Iteration 233, loss = 0.41675938
Iteration 234, loss = 0.41548219
Iteration 235, loss = 0.41594692
Iteration 236, loss = 0.41555557
Iteration 237, loss = 0.41549201
Iteration 238, loss = 0.41583402
Iteration 239, loss = 0.41547952
Iteration 240, loss = 0.41538027
Iteration 241, loss = 0.41485612
Iteration 242, loss = 0.41584610
Iteration 243, loss = 0.41483295
Iteration 244, loss = 0.41457315
Iteration 245, loss = 0.41455670
Iteration 246, loss = 0.41469037
Iteration 247, loss = 0.41426809
Iteration 248, loss = 0.41433919
Iteration 249, loss = 0.41390006
Iteration 250, loss = 0.41386250
Iteration 251, loss = 0.41402391
Iteration 252, loss = 0.41393419
Iteration 253, loss = 0.41325081
Iteration 254, loss = 0.41333427
Iteration 255, loss = 0.41445847
Iteration 256, loss = 0.41344678
Iteration 257, loss = 0.41338048
Iteration 258, loss = 0.41368014
Iteration 259, loss = 0.41340376
Iteration 260, loss = 0.41359215
Iteration 261, loss = 0.41218237
Iteration 262, loss = 0.41300239
Iteration 263, loss = 0.41282628
Iteration 264, loss = 0.41279456
Iteration 265, loss = 0.41220238
Iteration 266, loss = 0.41220002
Iteration 267, loss = 0.41222530
Iteration 268, loss = 0.41115853
Iteration 269, loss = 0.41166172
Iteration 270, loss = 0.41089098
Iteration 271, loss = 0.41079603
Iteration 272, loss = 0.41140668
```

```
Iteration 273, loss = 0.41200970
Iteration 274, loss = 0.41143516
Iteration 275, loss = 0.41114819
Iteration 276, loss = 0.41128116
Iteration 277, loss = 0.41003695
Iteration 278, loss = 0.41059785
Iteration 279, loss = 0.41097740
Iteration 280, loss = 0.41019261
Iteration 281, loss = 0.41032456
Iteration 282, loss = 0.41038769
Iteration 283, loss = 0.41048093
Iteration 284, loss = 0.41012098
Iteration 285, loss = 0.41073772
Iteration 286, loss = 0.41004275
Iteration 287, loss = 0.40964005
Iteration 288, loss = 0.41019066
Iteration 289, loss = 0.40957262
Iteration 290, loss = 0.41008500
Iteration 291, loss = 0.40936498
Iteration 292, loss = 0.40948445
Iteration 293, loss = 0.40838437
Iteration 294, loss = 0.40870642
Iteration 295, loss = 0.40861720
Iteration 296, loss = 0.40944740
Iteration 297, loss = 0.40896778
Iteration 298, loss = 0.40939366
Iteration 299, loss = 0.40821632
Iteration 300, loss = 0.40864125
Iteration 301, loss = 0.40763399
Iteration 302, loss = 0.40779257
Iteration 303, loss = 0.40820481
Iteration 304, loss = 0.40800331
Iteration 305, loss = 0.40857895
Iteration 306, loss = 0.40737397
Iteration 307, loss = 0.40821840
Iteration 308, loss = 0.40744497
Iteration 309, loss = 0.40687171
Iteration 310, loss = 0.40748479
Iteration 311, loss = 0.40750001
Iteration 312, loss = 0.40800969
Iteration 313, loss = 0.40713118
Iteration 314, loss = 0.40631164
Iteration 315, loss = 0.40690300
Iteration 316, loss = 0.40725937
Iteration 317, loss = 0.40600710
Iteration 318, loss = 0.40614422
Iteration 319, loss = 0.40713094
Iteration 320, loss = 0.40730497
Iteration 321, loss = 0.40583628
Iteration 322, loss = 0.40642724
Iteration 323, loss = 0.40717697
Iteration 324, loss = 0.40572427
Iteration 325, loss = 0.40690223
Iteration 326, loss = 0.40566177
Iteration 327, loss = 0.40486177
Iteration 328, loss = 0.40445415
Iteration 329, loss = 0.40498282
Iteration 330, loss = 0.40460127
Iteration 331, loss = 0.40494083
Iteration 332, loss = 0.40498703
Iteration 333, loss = 0.40484328
```

```
Iteration 334, loss = 0.40431180
Iteration 335, loss = 0.40552162
Iteration 336, loss = 0.40380413
Iteration 337, loss = 0.40428231
Iteration 338, loss = 0.40430722
Iteration 339, loss = 0.40401885
Iteration 340, loss = 0.40418448
Iteration 341, loss = 0.40400494
Iteration 342, loss = 0.40449577
Iteration 343, loss = 0.40323018
Iteration 344, loss = 0.40340973
Iteration 345, loss = 0.40257981
Iteration 346, loss = 0.40452870
Iteration 347, loss = 0.40332776
Iteration 348, loss = 0.40282145
Iteration 349, loss = 0.40295946
Iteration 350, loss = 0.40244966
Iteration 351, loss = 0.40216950
Iteration 352, loss = 0.40259676
Iteration 353, loss = 0.40350401
Iteration 354, loss = 0.40247834
Iteration 355, loss = 0.40286484
Iteration 356, loss = 0.40148197
Iteration 357, loss = 0.40169706
Iteration 358, loss = 0.40193600
Iteration 359, loss = 0.40193999
Iteration 360, loss = 0.40231445
Iteration 361, loss = 0.40196850
Iteration 362, loss = 0.40163229
Iteration 363, loss = 0.40383220
Iteration 364, loss = 0.40166360
Iteration 365, loss = 0.40118376
Iteration 366, loss = 0.40104146
Iteration 367, loss = 0.40104415
Iteration 368, loss = 0.40074503
Iteration 369, loss = 0.40124011
Iteration 370, loss = 0.40111888
Iteration 371, loss = 0.40119976
Iteration 372, loss = 0.40091556
Iteration 373, loss = 0.40128404
Iteration 374, loss = 0.40092900
Iteration 375, loss = 0.40104734
Iteration 376, loss = 0.40128468
Iteration 377, loss = 0.40086989
Iteration 378, loss = 0.40085442
Iteration 379, loss = 0.39949733
Iteration 380, loss = 0.39973438
Iteration 381, loss = 0.40011096
Iteration 382, loss = 0.39945865
Iteration 383, loss = 0.39945908
Iteration 384, loss = 0.39968724
Iteration 385, loss = 0.40008907
Iteration 386, loss = 0.39884164
Iteration 387, loss = 0.40041623
Iteration 388, loss = 0.39920365
Iteration 389, loss = 0.40049950
Iteration 390, loss = 0.39968302
Iteration 391, loss = 0.39981323
Iteration 392, loss = 0.39889068
Iteration 393, loss = 0.39878887
Iteration 394, loss = 0.39962405
```

```
Iteration 395, loss = 0.39822727
Iteration 396, loss = 0.39851803
Iteration 397, loss = 0.39819783
Iteration 398, loss = 0.39896325
Iteration 399, loss = 0.39911387
Iteration 400, loss = 0.39820524
Iteration 401, loss = 0.39795432
Iteration 402, loss = 0.39994928
Iteration 403, loss = 0.39726154
Iteration 404, loss = 0.39813167
Iteration 405, loss = 0.39693983
Iteration 406, loss = 0.39784051
Iteration 407, loss = 0.39683083
Iteration 408, loss = 0.39753698
Iteration 409, loss = 0.39681262
Iteration 410, loss = 0.39698175
Iteration 411, loss = 0.39747453
Iteration 412, loss = 0.39742314
Iteration 413, loss = 0.39708279
Iteration 414, loss = 0.39643424
Iteration 415, loss = 0.39646842
Iteration 416, loss = 0.39581253
Iteration 417, loss = 0.39586776
Iteration 418, loss = 0.39751097
Iteration 419, loss = 0.39651259
Iteration 420, loss = 0.39666633
Iteration 421, loss = 0.39595431
Iteration 422, loss = 0.39660313
Iteration 423, loss = 0.39564694
Iteration 424, loss = 0.39500883
Iteration 425, loss = 0.39566605
Iteration 426, loss = 0.39489822
Iteration 427, loss = 0.39528847
Iteration 428, loss = 0.39578020
Iteration 429, loss = 0.39458398
Iteration 430, loss = 0.39432843
Iteration 431, loss = 0.39495596
Iteration 432, loss = 0.39546633
Iteration 433, loss = 0.39456618
Iteration 434, loss = 0.39513014
Iteration 435, loss = 0.39515533
Iteration 436, loss = 0.39532402
Iteration 437, loss = 0.39568910
Iteration 438, loss = 0.39580106
Iteration 439, loss = 0.39506771
Iteration 440, loss = 0.39489654
Iteration 441, loss = 0.39531401
Training loss did not improve more than tol=0.000001 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.62322411
Iteration 2, loss = 0.55972838
Iteration 3, loss = 0.52407060
Iteration 4, loss = 0.50838967
Iteration 5, loss = 0.50167734
Iteration 6, loss = 0.49656301
Iteration 7, loss = 0.49364963
Iteration 8, loss = 0.49157446
Iteration 9, loss = 0.48961346
Iteration 10, loss = 0.48809823
Iteration 11, loss = 0.48650103
Iteration 12, loss = 0.48502781
```

```
Iteration 13, loss = 0.48370642
Iteration 14, loss = 0.48305336
Iteration 15, loss = 0.48171473
Iteration 16, loss = 0.48048942
Iteration 17, loss = 0.47979214
Iteration 18, loss = 0.47900039
Iteration 19, loss = 0.47817922
Iteration 20, loss = 0.47722284
Iteration 21, loss = 0.47622172
Iteration 22, loss = 0.47579486
Iteration 23, loss = 0.47482545
Iteration 24, loss = 0.47453642
Iteration 25, loss = 0.47345375
Iteration 26, loss = 0.47308561
Iteration 27, loss = 0.47180545
Iteration 28, loss = 0.47143377
Iteration 29, loss = 0.47063679
Iteration 30, loss = 0.46987850
Iteration 31, loss = 0.46945425
Iteration 32, loss = 0.46862682
Iteration 33, loss = 0.46811550
Iteration 34, loss = 0.46753886
Iteration 35, loss = 0.46651636
Iteration 36, loss = 0.46606461
Iteration 37, loss = 0.46531100
Iteration 38, loss = 0.46494806
Iteration 39, loss = 0.46473945
Iteration 40, loss = 0.46477346
Iteration 41, loss = 0.46312975
Iteration 42, loss = 0.46293178
Iteration 43, loss = 0.46194270
Iteration 44, loss = 0.46120268
Iteration 45, loss = 0.46077789
Iteration 46, loss = 0.46071454
Iteration 47, loss = 0.46052076
Iteration 48, loss = 0.45923658
Iteration 49, loss = 0.45896009
Iteration 50, loss = 0.45893641
Iteration 51, loss = 0.45794525
Iteration 52, loss = 0.45779968
Iteration 53, loss = 0.45726674
Iteration 54, loss = 0.45683646
Iteration 55, loss = 0.45628785
Iteration 56, loss = 0.45667418
Iteration 57, loss = 0.45553361
Iteration 58, loss = 0.45502124
Iteration 59, loss = 0.45481011
Iteration 60, loss = 0.45470558
Iteration 61, loss = 0.45358250
Iteration 62, loss = 0.45338738
Iteration 63, loss = 0.45323721
Iteration 64, loss = 0.45242725
Iteration 65, loss = 0.45253539
Iteration 66, loss = 0.45215247
Iteration 67, loss = 0.45179584
Iteration 68, loss = 0.45140037
Iteration 69, loss = 0.45122566
Iteration 70, loss = 0.45053704
Iteration 71, loss = 0.44977014
Iteration 72, loss = 0.44948510
Iteration 73, loss = 0.44937904
```

```
Iteration 74, loss = 0.44877185
Iteration 75, loss = 0.44837898
Iteration 76, loss = 0.44837089
Iteration 77, loss = 0.44782430
Iteration 78, loss = 0.44763409
Iteration 79, loss = 0.44711357
Iteration 80, loss = 0.44691091
Iteration 81, loss = 0.44677441
Iteration 82, loss = 0.44620023
Iteration 83, loss = 0.44622648
Iteration 84, loss = 0.44567643
Iteration 85, loss = 0.44529417
Iteration 86, loss = 0.44529632
Iteration 87, loss = 0.44479900
Iteration 88, loss = 0.44433493
Iteration 89, loss = 0.44465423
Iteration 90, loss = 0.44400170
Iteration 91, loss = 0.44420458
Iteration 92, loss = 0.44344921
Iteration 93, loss = 0.44386066
Iteration 94, loss = 0.44264002
Iteration 95, loss = 0.44239306
Iteration 96, loss = 0.44205706
Iteration 97, loss = 0.44227847
Iteration 98, loss = 0.44151808
Iteration 99, loss = 0.44154604
Iteration 100, loss = 0.44084015
Iteration 101, loss = 0.44075834
Iteration 102, loss = 0.44052936
Iteration 103, loss = 0.44013393
Iteration 104, loss = 0.44032331
Iteration 105, loss = 0.43944068
Iteration 106, loss = 0.43964861
Iteration 107, loss = 0.43954776
Iteration 108, loss = 0.43848811
Iteration 109, loss = 0.43855348
Iteration 110, loss = 0.43818095
Iteration 111, loss = 0.43787733
Iteration 112, loss = 0.43814768
Iteration 113, loss = 0.43697470
Iteration 114, loss = 0.43751960
Iteration 115, loss = 0.43693599
Iteration 116, loss = 0.43650933
Iteration 117, loss = 0.43619322
Iteration 118, loss = 0.43666625
Iteration 119, loss = 0.43564731
Iteration 120, loss = 0.43602286
Iteration 121, loss = 0.43496024
Iteration 122, loss = 0.43630860
Iteration 123, loss = 0.43451648
Iteration 124, loss = 0.43595917
Iteration 125, loss = 0.43486497
Iteration 126, loss = 0.43409439
Iteration 127, loss = 0.43414066
Iteration 128, loss = 0.43371413
Iteration 129, loss = 0.43343193
Iteration 130, loss = 0.43347987
Iteration 131, loss = 0.43328770
Iteration 132, loss = 0.43248612
Iteration 133, loss = 0.43289149
Iteration 134, loss = 0.43199779
```

```
Iteration 135, loss = 0.43153463
Iteration 136, loss = 0.43173530
Iteration 137, loss = 0.43303981
Iteration 138, loss = 0.43126653
Iteration 139, loss = 0.43076822
Iteration 140, loss = 0.43063676
Iteration 141, loss = 0.43024908
Iteration 142, loss = 0.43000899
Iteration 143, loss = 0.43026472
Iteration 144, loss = 0.42981977
Iteration 145, loss = 0.42985003
Iteration 146, loss = 0.42982463
Iteration 147, loss = 0.42878703
Iteration 148, loss = 0.42956152
Iteration 149, loss = 0.42855163
Iteration 150, loss = 0.42812424
Iteration 151, loss = 0.42886124
Iteration 152, loss = 0.42793043
Iteration 153, loss = 0.42774943
Iteration 154, loss = 0.42734637
Iteration 155, loss = 0.42760751
Iteration 156, loss = 0.42741283
Iteration 157, loss = 0.42685328
Iteration 158, loss = 0.42659377
Iteration 159, loss = 0.42629659
Iteration 160, loss = 0.42670888
Iteration 161, loss = 0.42586359
Iteration 162, loss = 0.42584293
Iteration 163, loss = 0.42542804
Iteration 164, loss = 0.42579205
Iteration 165, loss = 0.42550814
Iteration 166, loss = 0.42446969
Iteration 167, loss = 0.42440864
Iteration 168, loss = 0.42443949
Iteration 169, loss = 0.42470147
Iteration 170, loss = 0.42496798
Iteration 171, loss = 0.42483283
Iteration 172, loss = 0.42386920
Iteration 173, loss = 0.42348413
Iteration 174, loss = 0.42279611
Iteration 175, loss = 0.42324493
Iteration 176, loss = 0.42318116
Iteration 177, loss = 0.42277320
Iteration 178, loss = 0.42359261
Iteration 179, loss = 0.42175762
Iteration 180, loss = 0.42395565
Iteration 181, loss = 0.42185493
Iteration 182, loss = 0.42253581
Iteration 183, loss = 0.42115649
Iteration 184, loss = 0.42235080
Iteration 185, loss = 0.42163681
Iteration 186, loss = 0.42126473
Iteration 187, loss = 0.42081386
Iteration 188, loss = 0.42043286
Iteration 189, loss = 0.42034416
Iteration 190, loss = 0.42011459
Iteration 191, loss = 0.42048920
Iteration 192, loss = 0.41945390
Iteration 193, loss = 0.41928330
Iteration 194, loss = 0.41918840
Iteration 195, loss = 0.41944153
```

```
Iteration 196, loss = 0.41930961
Iteration 197, loss = 0.41965380
Iteration 198, loss = 0.41873580
Iteration 199, loss = 0.41945875
Iteration 200, loss = 0.41877184
Iteration 201, loss = 0.41858487
Iteration 202, loss = 0.41871569
Iteration 203, loss = 0.41752103
Iteration 204, loss = 0.41719677
Iteration 205, loss = 0.41744763
Iteration 206, loss = 0.41747027
Iteration 207, loss = 0.41711041
Iteration 208, loss = 0.41726086
Iteration 209, loss = 0.41678046
Iteration 210, loss = 0.41609689
Iteration 211, loss = 0.41641762
Iteration 212, loss = 0.41666354
Iteration 213, loss = 0.41573738
Iteration 214, loss = 0.41625440
Iteration 215, loss = 0.41605061
Iteration 216, loss = 0.41618079
Iteration 217, loss = 0.41750051
Iteration 218, loss = 0.41629039
Iteration 219, loss = 0.41577224
Iteration 220, loss = 0.41471358
Iteration 221, loss = 0.41471400
Iteration 222, loss = 0.41482074
Iteration 223, loss = 0.41503459
Iteration 224, loss = 0.41402385
Iteration 225, loss = 0.41749504
Iteration 226, loss = 0.41519243
Iteration 227, loss = 0.41527644
Iteration 228, loss = 0.41297231
Iteration 229, loss = 0.41471656
Iteration 230, loss = 0.41476194
Iteration 231, loss = 0.41320355
Iteration 232, loss = 0.41383161
Iteration 233, loss = 0.41403656
Iteration 234, loss = 0.41326506
Iteration 235, loss = 0.41228846
Iteration 236, loss = 0.41257558
Iteration 237, loss = 0.41285418
Iteration 238, loss = 0.41223953
Iteration 239, loss = 0.41188832
Iteration 240, loss = 0.41146626
Iteration 241, loss = 0.41145415
Iteration 242, loss = 0.41192459
Iteration 243, loss = 0.41157167
Iteration 244, loss = 0.41160156
Iteration 245, loss = 0.41030618
Iteration 246, loss = 0.41161801
Iteration 247, loss = 0.41108573
Iteration 248, loss = 0.41154588
Iteration 249, loss = 0.41056126
Iteration 250, loss = 0.41097445
Iteration 251, loss = 0.40970596
Iteration 252, loss = 0.40953694
Iteration 253, loss = 0.41049688
Iteration 254, loss = 0.40979392
Iteration 255, loss = 0.40979483
Iteration 256, loss = 0.40924834
```

```
Iteration 257, loss = 0.40910192
Iteration 258, loss = 0.40906012
Iteration 259, loss = 0.41052968
Iteration 260, loss = 0.40856674
Iteration 261, loss = 0.40847955
Iteration 262, loss = 0.40877881
Iteration 263, loss = 0.40806918
Iteration 264, loss = 0.40823629
Iteration 265, loss = 0.40775083
Iteration 266, loss = 0.40728685
Iteration 267, loss = 0.40743662
Iteration 268, loss = 0.40821636
Iteration 269, loss = 0.40798309
Iteration 270, loss = 0.40763149
Iteration 271, loss = 0.40694287
Iteration 272, loss = 0.40709972
Iteration 273, loss = 0.40681168
Iteration 274, loss = 0.40615345
Iteration 275, loss = 0.40738233
Iteration 276, loss = 0.40700545
Iteration 277, loss = 0.40618802
Iteration 278, loss = 0.40607723
Iteration 279, loss = 0.40692024
Iteration 280, loss = 0.40594814
Iteration 281, loss = 0.40561311
Iteration 282, loss = 0.40568333
Iteration 283, loss = 0.40521211
Iteration 284, loss = 0.40506168
Iteration 285, loss = 0.40501110
Iteration 286, loss = 0.40551291
Iteration 287, loss = 0.40571583
Iteration 288, loss = 0.40453692
Iteration 289, loss = 0.40414248
Iteration 290, loss = 0.40472771
Iteration 291, loss = 0.40417815
Iteration 292, loss = 0.40457739
Iteration 293, loss = 0.40350201
Iteration 294, loss = 0.40369894
Iteration 295, loss = 0.40473825
Iteration 296, loss = 0.40493723
Iteration 297, loss = 0.40307418
Iteration 298, loss = 0.40333362
Iteration 299, loss = 0.40472152
Iteration 300, loss = 0.40400642
Iteration 301, loss = 0.40342923
Iteration 302, loss = 0.40233617
Iteration 303, loss = 0.40277381
Iteration 304, loss = 0.40240057
Iteration 305, loss = 0.40248575
Iteration 306, loss = 0.40243739
Iteration 307, loss = 0.40317107
Iteration 308, loss = 0.40162242
Iteration 309, loss = 0.40234280
Iteration 310, loss = 0.40134978
Iteration 311, loss = 0.40145621
Iteration 312, loss = 0.40100076
Iteration 313, loss = 0.40136671
Iteration 314, loss = 0.40132725
Iteration 315, loss = 0.40135132
Iteration 316, loss = 0.40097807
Iteration 317, loss = 0.40043090
```

```
Iteration 318, loss = 0.40073580
Iteration 319, loss = 0.40030140
Iteration 320, loss = 0.40008650
Iteration 321, loss = 0.40057649
Iteration 322, loss = 0.40057041
Iteration 323, loss = 0.40125875
Iteration 324, loss = 0.39975464
Iteration 325, loss = 0.39976951
Iteration 326, loss = 0.39892429
Iteration 327, loss = 0.39962908
Iteration 328, loss = 0.39871480
Iteration 329, loss = 0.39892162
Iteration 330, loss = 0.39919390
Iteration 331, loss = 0.39804257
Iteration 332, loss = 0.39921654
Iteration 333, loss = 0.39950724
Iteration 334, loss = 0.39884251
Iteration 335, loss = 0.39853605
Iteration 336, loss = 0.39787863
Iteration 337, loss = 0.39860273
Iteration 338, loss = 0.39706267
Iteration 339, loss = 0.39846827
Iteration 340, loss = 0.39763595
Iteration 341, loss = 0.39768744
Iteration 342, loss = 0.39797129
Iteration 343, loss = 0.39723525
Iteration 344, loss = 0.39779315
Iteration 345, loss = 0.39755884
Iteration 346, loss = 0.39639996
Iteration 347, loss = 0.39773640
Iteration 348, loss = 0.39650884
Iteration 349, loss = 0.39799131
Iteration 350, loss = 0.39712720
Iteration 351, loss = 0.39612281
Iteration 352, loss = 0.39690361
Iteration 353, loss = 0.39700085
Iteration 354, loss = 0.39703884
Iteration 355, loss = 0.39666249
Iteration 356, loss = 0.39670939
Iteration 357, loss = 0.39671786
Iteration 358, loss = 0.39806040
Iteration 359, loss = 0.39549932
Iteration 360, loss = 0.39915843
Iteration 361, loss = 0.39510949
Iteration 362, loss = 0.39540140
Iteration 363, loss = 0.39468420
Iteration 364, loss = 0.39474500
Iteration 365, loss = 0.39424971
Iteration 366, loss = 0.39489834
Iteration 367, loss = 0.39454045
Iteration 368, loss = 0.39453469
Iteration 369, loss = 0.39467760
Iteration 370, loss = 0.39402962
Iteration 371, loss = 0.39456681
Iteration 372, loss = 0.39323158
Iteration 373, loss = 0.39410591
Iteration 374, loss = 0.39333796
Iteration 375, loss = 0.39377038
Iteration 376, loss = 0.39431749
Iteration 377, loss = 0.39580774
Iteration 378, loss = 0.39363181
```

```
Iteration 379, loss = 0.39312690
Iteration 380, loss = 0.39360134
Iteration 381, loss = 0.39273253
Iteration 382, loss = 0.39349492
Iteration 383, loss = 0.39272218
Iteration 384, loss = 0.39249596
Iteration 385, loss = 0.39145897
Iteration 386, loss = 0.39166348
Iteration 387, loss = 0.39158616
Iteration 388, loss = 0.39158462
Iteration 389, loss = 0.39182921
Iteration 390, loss = 0.39141604
Iteration 391, loss = 0.39358720
Iteration 392, loss = 0.39441098
Iteration 393, loss = 0.39210806
Iteration 394, loss = 0.39141984
Iteration 395, loss = 0.39135960
Iteration 396, loss = 0.39177242
Iteration 397, loss = 0.39179498
Iteration 398, loss = 0.39141795
Iteration 399, loss = 0.39139497
Iteration 400, loss = 0.39099447
Iteration 401, loss = 0.39243711
Iteration 402, loss = 0.39090068
Iteration 403, loss = 0.39200250
Iteration 404, loss = 0.39088945
Iteration 405, loss = 0.39023684
Iteration 406, loss = 0.39001544
Iteration 407, loss = 0.38977105
Iteration 408, loss = 0.38941065
Iteration 409, loss = 0.38953365
Iteration 410, loss = 0.38921486
Iteration 411, loss = 0.38986528
Iteration 412, loss = 0.38961766
Iteration 413, loss = 0.38972724
Iteration 414, loss = 0.38852681
Iteration 415, loss = 0.38947512
Iteration 416, loss = 0.39023973
Iteration 417, loss = 0.38908743
Iteration 418, loss = 0.38879600
Iteration 419, loss = 0.38855741
Iteration 420, loss = 0.38854185
Iteration 421, loss = 0.38818336
Iteration 422, loss = 0.38829001
Iteration 423, loss = 0.38741699
Iteration 424, loss = 0.38989488
Iteration 425, loss = 0.38729257
Iteration 426, loss = 0.38793509
Iteration 427, loss = 0.38720233
Iteration 428, loss = 0.38748259
Iteration 429, loss = 0.38757154
Iteration 430, loss = 0.38695062
Iteration 431, loss = 0.38709671
Iteration 432, loss = 0.38670280
Iteration 433, loss = 0.38699635
Iteration 434, loss = 0.38747978
Iteration 435, loss = 0.38745208
Iteration 436, loss = 0.38709774
Iteration 437, loss = 0.38700625
Iteration 438, loss = 0.38799129
Iteration 439, loss = 0.38713079
```

```
Iteration 440, loss = 0.38672617
Iteration 441, loss = 0.38685819
Iteration 442, loss = 0.38840714
Iteration 443, loss = 0.38783312
Training loss did not improve more than tol=0.000001 for 10 consecutive ep
ochs. Stopping.
Iteration 1, loss = 0.66171873
Iteration 2, loss = 0.57968207
Iteration 3, loss = 0.53462373
Iteration 4, loss = 0.51132736
Iteration 5, loss = 0.49966246
Iteration 6, loss = 0.49392235
Iteration 7, loss = 0.49116885
Iteration 8, loss = 0.48937134
Iteration 9, loss = 0.48801914
Iteration 10, loss = 0.48665594
Iteration 11, loss = 0.48541132
Iteration 12, loss = 0.48440045
Iteration 13, loss = 0.48353448
Iteration 14, loss = 0.48247223
Iteration 15, loss = 0.48152458
Iteration 16, loss = 0.48041023
Iteration 17, loss = 0.47954881
Iteration 18, loss = 0.47862974
Iteration 19, loss = 0.47801125
Iteration 20, loss = 0.47690280
Iteration 21, loss = 0.47644688
Iteration 22, loss = 0.47557836
Iteration 23, loss = 0.47480111
Iteration 24, loss = 0.47448853
Iteration 25, loss = 0.47372048
Iteration 26, loss = 0.47275432
Iteration 27, loss = 0.47224216
Iteration 28, loss = 0.47167849
Iteration 29, loss = 0.47093645
Iteration 30, loss = 0.47021516
Iteration 31, loss = 0.46958735
Iteration 32, loss = 0.46945458
Iteration 33, loss = 0.46930778
Iteration 34, loss = 0.46807342
Iteration 35, loss = 0.46735593
Iteration 36, loss = 0.46674526
Iteration 37, loss = 0.46612456
Iteration 38, loss = 0.46562832
Iteration 39, loss = 0.46515120
Iteration 40, loss = 0.46449731
Iteration 41, loss = 0.46396027
Iteration 42, loss = 0.46343682
Iteration 43, loss = 0.46329933
Iteration 44, loss = 0.46281040
Iteration 45, loss = 0.46218997
Iteration 46, loss = 0.46156158
Iteration 47, loss = 0.46139769
Iteration 48, loss = 0.46071437
Iteration 49, loss = 0.46059692
Iteration 50, loss = 0.46000731
Iteration 51, loss = 0.45966384
Iteration 52, loss = 0.45912255
Iteration 53, loss = 0.45891461
Iteration 54, loss = 0.45814323
Iteration 55, loss = 0.45779045
```

```
Iteration 56, loss = 0.45713196
Iteration 57, loss = 0.45704651
Iteration 58, loss = 0.45693680
Iteration 59, loss = 0.45696224
Iteration 60, loss = 0.45563689
Iteration 61, loss = 0.45598246
Iteration 62, loss = 0.45502471
Iteration 63, loss = 0.45437309
Iteration 64, loss = 0.45453215
Iteration 65, loss = 0.45382355
Iteration 66, loss = 0.45377656
Iteration 67, loss = 0.45310552
Iteration 68, loss = 0.45287842
Iteration 69, loss = 0.45243229
Iteration 70, loss = 0.45239984
Iteration 71, loss = 0.45345300
Iteration 72, loss = 0.45164764
Iteration 73, loss = 0.45141372
Iteration 74, loss = 0.45127851
Iteration 75, loss = 0.45076893
Iteration 76, loss = 0.45145772
Iteration 77, loss = 0.45039804
Iteration 78, loss = 0.44985347
Iteration 79, loss = 0.44910898
Iteration 80, loss = 0.44950442
Iteration 81, loss = 0.44924635
Iteration 82, loss = 0.44903326
Iteration 83, loss = 0.44760119
Iteration 84, loss = 0.44793764
Iteration 85, loss = 0.44735648
Iteration 86, loss = 0.44725248
Iteration 87, loss = 0.44740519
Iteration 88, loss = 0.44654069
Iteration 89, loss = 0.44702597
Iteration 90, loss = 0.44621032
Iteration 91, loss = 0.44577341
Iteration 92, loss = 0.44614706
Iteration 93, loss = 0.44565823
Iteration 94, loss = 0.44491385
Iteration 95, loss = 0.44548936
Iteration 96, loss = 0.44509832
Iteration 97, loss = 0.44439949
Iteration 98, loss = 0.44425242
Iteration 99, loss = 0.44398453
Iteration 100, loss = 0.44370458
Iteration 101, loss = 0.44357465
Iteration 102, loss = 0.44310805
Iteration 103, loss = 0.44265050
Iteration 104, loss = 0.44255283
Iteration 105, loss = 0.44256740
Iteration 106, loss = 0.44164605
Iteration 107, loss = 0.44225427
Iteration 108, loss = 0.44234501
Iteration 109, loss = 0.44160163
Iteration 110, loss = 0.44089323
Iteration 111, loss = 0.44118760
Iteration 112, loss = 0.44021054
Iteration 113, loss = 0.44038553
Iteration 114, loss = 0.44111211
Iteration 115, loss = 0.44139118
Iteration 116, loss = 0.43966039
```

```
Iteration 117, loss = 0.43929804
Iteration 118, loss = 0.44029277
Iteration 119, loss = 0.43977748
Iteration 120, loss = 0.43880568
Iteration 121, loss = 0.43963677
Iteration 122, loss = 0.43876799
Iteration 123, loss = 0.43801529
Iteration 124, loss = 0.43801479
Iteration 125, loss = 0.43753735
Iteration 126, loss = 0.43781323
Iteration 127, loss = 0.43708833
Iteration 128, loss = 0.43718249
Iteration 129, loss = 0.43675963
Iteration 130, loss = 0.43723976
Iteration 131, loss = 0.43612451
Iteration 132, loss = 0.43799445
Iteration 133, loss = 0.43644058
Iteration 134, loss = 0.43653573
Iteration 135, loss = 0.43488607
Iteration 136, loss = 0.43538415
Iteration 137, loss = 0.43518528
Iteration 138, loss = 0.43487774
Iteration 139, loss = 0.43535821
Iteration 140, loss = 0.43526172
Iteration 141, loss = 0.43429133
Iteration 142, loss = 0.43459013
Iteration 143, loss = 0.43392517
Iteration 144, loss = 0.43383194
Iteration 145, loss = 0.43323704
Iteration 146, loss = 0.43344434
Iteration 147, loss = 0.43301548
Iteration 148, loss = 0.43256505
Iteration 149, loss = 0.43235893
Iteration 150, loss = 0.43263049
Iteration 151, loss = 0.43175979
Iteration 152, loss = 0.43181711
Iteration 153, loss = 0.43236490
Iteration 154, loss = 0.43306405
Iteration 155, loss = 0.43174916
Iteration 156, loss = 0.43142686
Iteration 157, loss = 0.43194766
Iteration 158, loss = 0.43091726
Iteration 159, loss = 0.43106617
Iteration 160, loss = 0.43058867
Iteration 161, loss = 0.43063134
Iteration 162, loss = 0.43011053
Iteration 163, loss = 0.42998136
Iteration 164, loss = 0.42999711
Iteration 165, loss = 0.43042127
Iteration 166, loss = 0.43044870
Iteration 167, loss = 0.43030699
Iteration 168, loss = 0.42903586
Iteration 169, loss = 0.42903563
Iteration 170, loss = 0.42859317
Iteration 171, loss = 0.42830461
Iteration 172, loss = 0.42894015
Iteration 173, loss = 0.42792237
Iteration 174, loss = 0.42794847
Iteration 175, loss = 0.42813457
Iteration 176, loss = 0.42744897
Iteration 177, loss = 0.42811325
```

```
Iteration 178, loss = 0.42753289
Iteration 179, loss = 0.42703103
Iteration 180, loss = 0.42787334
Iteration 181, loss = 0.42675418
Iteration 182, loss = 0.42716442
Iteration 183, loss = 0.42706118
Iteration 184, loss = 0.42695353
Iteration 185, loss = 0.42736942
Iteration 186, loss = 0.42635547
Iteration 187, loss = 0.42615216
Iteration 188, loss = 0.42547687
Iteration 189, loss = 0.42622008
Iteration 190, loss = 0.42520624
Iteration 191, loss = 0.42528304
Iteration 192, loss = 0.42456895
Iteration 193, loss = 0.42472700
Iteration 194, loss = 0.42473626
Iteration 195, loss = 0.42417055
Iteration 196, loss = 0.42539314
Iteration 197, loss = 0.42473362
Iteration 198, loss = 0.42590218
Iteration 199, loss = 0.42461282
Iteration 200, loss = 0.42379354
Iteration 201, loss = 0.42331231
Iteration 202, loss = 0.42314262
Iteration 203, loss = 0.42347701
Iteration 204, loss = 0.42253679
Iteration 205, loss = 0.42291877
Iteration 206, loss = 0.42278462
Iteration 207, loss = 0.42309573
Iteration 208, loss = 0.42272740
Iteration 209, loss = 0.42204027
Iteration 210, loss = 0.42189734
Iteration 211, loss = 0.42179733
Iteration 212, loss = 0.42180847
Iteration 213, loss = 0.42138075
Iteration 214, loss = 0.42230180
Iteration 215, loss = 0.42098564
Iteration 216, loss = 0.42148637
Iteration 217, loss = 0.42091276
Iteration 218, loss = 0.42057112
Iteration 219, loss = 0.42040808
Iteration 220, loss = 0.42016474
Iteration 221, loss = 0.42093318
Iteration 222, loss = 0.42032244
Iteration 223, loss = 0.42055667
Iteration 224, loss = 0.42028739
Iteration 225, loss = 0.41942228
Iteration 226, loss = 0.42082808
Iteration 227, loss = 0.41963117
Iteration 228, loss = 0.41983846
Iteration 229, loss = 0.41932953
Iteration 230, loss = 0.41925335
Iteration 231, loss = 0.41935823
Iteration 232, loss = 0.41865915
Iteration 233, loss = 0.41856807
Iteration 234, loss = 0.41824973
Iteration 235, loss = 0.41989494
Iteration 236, loss = 0.42058949
Iteration 237, loss = 0.41842018
Iteration 238, loss = 0.41911851
```

```
Iteration 239, loss = 0.41827426
Iteration 240, loss = 0.41765712
Iteration 241, loss = 0.41906936
Iteration 242, loss = 0.41694753
Iteration 243, loss = 0.41753372
Iteration 244, loss = 0.41726932
Iteration 245, loss = 0.41714443
Iteration 246, loss = 0.41708933
Iteration 247, loss = 0.41727876
Iteration 248, loss = 0.41651476
Iteration 249, loss = 0.41639701
Iteration 250, loss = 0.41704534
Iteration 251, loss = 0.41656833
Iteration 252, loss = 0.41599984
Iteration 253, loss = 0.41603585
Iteration 254, loss = 0.41628725
Iteration 255, loss = 0.41597429
Iteration 256, loss = 0.41560006
Iteration 257, loss = 0.41597706
Iteration 258, loss = 0.41573943
Iteration 259, loss = 0.41557962
Iteration 260, loss = 0.41588960
Iteration 261, loss = 0.41504402
Iteration 262, loss = 0.41457342
Iteration 263, loss = 0.41489483
Iteration 264, loss = 0.41465802
Iteration 265, loss = 0.41384909
Iteration 266, loss = 0.41468858
Iteration 267, loss = 0.41480918
Iteration 268, loss = 0.41418492
Iteration 269, loss = 0.41451789
Iteration 270, loss = 0.41385639
Iteration 271, loss = 0.41315253
Iteration 272, loss = 0.41384841
Iteration 273, loss = 0.41312895
Iteration 274, loss = 0.41339423
Iteration 275, loss = 0.41266224
Iteration 276, loss = 0.41330541
Iteration 277, loss = 0.41252983
Iteration 278, loss = 0.41315607
Iteration 279, loss = 0.41370201
Iteration 280, loss = 0.41392107
Iteration 281, loss = 0.41209030
Iteration 282, loss = 0.41359359
Iteration 283, loss = 0.41191359
Iteration 284, loss = 0.41255531
Iteration 285, loss = 0.41182378
Iteration 286, loss = 0.41186207
Iteration 287, loss = 0.41189205
Iteration 288, loss = 0.41105749
Iteration 289, loss = 0.41121096
Iteration 290, loss = 0.41211907
Iteration 291, loss = 0.41172603
Iteration 292, loss = 0.41161211
Iteration 293, loss = 0.41092226
Iteration 294, loss = 0.41092106
Iteration 295, loss = 0.41146690
Iteration 296, loss = 0.40990038
Iteration 297, loss = 0.41008297
Iteration 298, loss = 0.41036161
Iteration 299, loss = 0.40985993
```

```
Iteration 300, loss = 0.41095978
Iteration 301, loss = 0.40989823
Iteration 302, loss = 0.40965470
Iteration 303, loss = 0.40897062
Iteration 304, loss = 0.41005989
Iteration 305, loss = 0.41012691
Iteration 306, loss = 0.41066355
Iteration 307, loss = 0.40928035
Iteration 308, loss = 0.40927602
Iteration 309, loss = 0.40978749
Iteration 310, loss = 0.40834218
Iteration 311, loss = 0.41049145
Iteration 312, loss = 0.40829253
Iteration 313, loss = 0.40858820
Iteration 314, loss = 0.40923820
Iteration 315, loss = 0.40830288
Iteration 316, loss = 0.40888012
Iteration 317, loss = 0.40813485
Iteration 318, loss = 0.40818329
Iteration 319, loss = 0.40828163
Iteration 320, loss = 0.40871203
Iteration 321, loss = 0.40789130
Iteration 322, loss = 0.40786684
Iteration 323, loss = 0.40691525
Iteration 324, loss = 0.40768525
Iteration 325, loss = 0.40753536
Iteration 326, loss = 0.40735850
Iteration 327, loss = 0.40630679
Iteration 328, loss = 0.40685634
Iteration 329, loss = 0.40639721
Iteration 330, loss = 0.40708227
Iteration 331, loss = 0.40676651
Iteration 332, loss = 0.40569818
Iteration 333, loss = 0.40682985
Iteration 334, loss = 0.40646764
Iteration 335, loss = 0.40658230
Iteration 336, loss = 0.40558636
Iteration 337, loss = 0.40842068
Iteration 338, loss = 0.40572420
Iteration 339, loss = 0.40725413
Iteration 340, loss = 0.40444582
Iteration 341, loss = 0.40668217
Iteration 342, loss = 0.40585698
Iteration 343, loss = 0.40586965
Iteration 344, loss = 0.40535886
Iteration 345, loss = 0.40544858
Iteration 346, loss = 0.40524741
Iteration 347, loss = 0.40526095
Iteration 348, loss = 0.40518487
Iteration 349, loss = 0.40431915
Iteration 350, loss = 0.40437759
Iteration 351, loss = 0.40462554
Iteration 352, loss = 0.40373478
Iteration 353, loss = 0.40437551
Iteration 354, loss = 0.40410866
Iteration 355, loss = 0.40372799
Iteration 356, loss = 0.40388163
Iteration 357, loss = 0.40389831
Iteration 358, loss = 0.40476095
Iteration 359, loss = 0.40358975
Iteration 360, loss = 0.40448031
```

```
Iteration 361, loss = 0.40336983
Iteration 362, loss = 0.40375065
Iteration 363, loss = 0.40368702
Iteration 364, loss = 0.40306679
Iteration 365, loss = 0.40397728
Iteration 366, loss = 0.40357041
Iteration 367, loss = 0.40347239
Iteration 368, loss = 0.40418405
Iteration 369, loss = 0.40310454
Iteration 370, loss = 0.40391824
Iteration 371, loss = 0.40441854
Iteration 372, loss = 0.40241401
Iteration 373, loss = 0.40275829
Iteration 374, loss = 0.40210246
Iteration 375, loss = 0.40226551
Iteration 376, loss = 0.40406066
Iteration 377, loss = 0.40306808
Iteration 378, loss = 0.40124555
Iteration 379, loss = 0.40129258
Iteration 380, loss = 0.40223652
Iteration 381, loss = 0.40198760
Iteration 382, loss = 0.40149827
Iteration 383, loss = 0.40256542
Iteration 384, loss = 0.40061837
Iteration 385, loss = 0.40123760
Iteration 386, loss = 0.40274359
Iteration 387, loss = 0.40324986
Iteration 388, loss = 0.40200610
Iteration 389, loss = 0.40237685
Iteration 390, loss = 0.40043823
Iteration 391, loss = 0.40020824
Iteration 392, loss = 0.40106839
Iteration 393, loss = 0.39958468
Iteration 394, loss = 0.40107827
Iteration 395, loss = 0.40029350
Iteration 396, loss = 0.40089610
Iteration 397, loss = 0.40045705
Iteration 398, loss = 0.40134526
Iteration 399, loss = 0.40049962
Iteration 400, loss = 0.40098099
Iteration 401, loss = 0.40005151
Iteration 402, loss = 0.40022577
Iteration 403, loss = 0.39909331
Iteration 404, loss = 0.39915436
Iteration 405, loss = 0.39944273
Iteration 406, loss = 0.39939459
Iteration 407, loss = 0.39945053
Iteration 408, loss = 0.39856996
Iteration 409, loss = 0.39901960
Iteration 410, loss = 0.39928010
Iteration 411, loss = 0.40033906
Iteration 412, loss = 0.39832427
Iteration 413, loss = 0.39963046
Iteration 414, loss = 0.39934755
Iteration 415, loss = 0.40002833
Iteration 416, loss = 0.39802291
Iteration 417, loss = 0.39968315
Iteration 418, loss = 0.40144940
Iteration 419, loss = 0.39825682
Iteration 420, loss = 0.39808517
Iteration 421, loss = 0.39897515
```

```
Iteration 422, loss = 0.39835308
Iteration 423, loss = 0.39824200
Iteration 424, loss = 0.39767228
Iteration 425, loss = 0.39829566
Iteration 426, loss = 0.39812883
Iteration 427, loss = 0.39734995
Iteration 428, loss = 0.39721890
Iteration 429, loss = 0.39684890
Iteration 430, loss = 0.39733245
Iteration 431, loss = 0.39774042
Iteration 432, loss = 0.39749676
Iteration 433, loss = 0.39819412
Iteration 434, loss = 0.39677179
Iteration 435, loss = 0.39709736
Iteration 436, loss = 0.39746884
Iteration 437, loss = 0.39619212
Iteration 438, loss = 0.39685872
Iteration 439, loss = 0.39763585
Iteration 440, loss = 0.39623509
Iteration 441, loss = 0.39679390
Iteration 442, loss = 0.39757126
Iteration 443, loss = 0.39634116
Iteration 444, loss = 0.39847891
Iteration 445, loss = 0.39584896
Iteration 446, loss = 0.39681052
Iteration 447, loss = 0.39519101
Iteration 448, loss = 0.39564513
Iteration 449, loss = 0.39678136
Iteration 450, loss = 0.39529767
Iteration 451, loss = 0.39688171
Iteration 452, loss = 0.39565212
Iteration 453, loss = 0.39595885
Iteration 454, loss = 0.39450468
Iteration 455, loss = 0.39624290
Iteration 456, loss = 0.39493869
Iteration 457, loss = 0.39535063
Iteration 458, loss = 0.39636266
Iteration 459, loss = 0.39559570
Iteration 460, loss = 0.39563760
Iteration 461, loss = 0.39478733
Iteration 462, loss = 0.39557489
Iteration 463, loss = 0.39431067
Iteration 464, loss = 0.39524248
Iteration 465, loss = 0.39538076
Iteration 466, loss = 0.39529476
Iteration 467, loss = 0.39515327
Iteration 468, loss = 0.39418543
Iteration 469, loss = 0.39413188
Iteration 470, loss = 0.39361192
Iteration 471, loss = 0.39443300
Iteration 472, loss = 0.39421320
Iteration 473, loss = 0.39331287
Iteration 474, loss = 0.39402876
Iteration 475, loss = 0.39456860
Iteration 476, loss = 0.39381011
Iteration 477, loss = 0.39351039
Iteration 478, loss = 0.39295685
Iteration 479, loss = 0.39303892
Iteration 480, loss = 0.39407060
Iteration 481, loss = 0.39306350
Iteration 482, loss = 0.39494779
```

```
Iteration 483, loss = 0.39303576
Iteration 484, loss = 0.39286760
Iteration 485, loss = 0.39390991
Iteration 486, loss = 0.39267690
Iteration 487, loss = 0.39271961
Iteration 488, loss = 0.39376286
Iteration 489, loss = 0.39223246
Iteration 490, loss = 0.39437527
Iteration 491, loss = 0.39278677
Iteration 492, loss = 0.39370182
Iteration 493, loss = 0.39251579
Iteration 494, loss = 0.39148227
Iteration 495, loss = 0.39227320
Iteration 496, loss = 0.39226397
Iteration 497, loss = 0.39190526
Iteration 498, loss = 0.39192081
Iteration 499, loss = 0.39160343
Iteration 500, loss = 0.39192577

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Iteration 1, loss = 0.67006897
Iteration 2, loss = 0.58681633
Iteration 3, loss = 0.53920839
Iteration 4, loss = 0.51393121
Iteration 5, loss = 0.50142647
Iteration 6, loss = 0.49457766
Iteration 7, loss = 0.49100552
Iteration 8, loss = 0.48890300
Iteration 9, loss = 0.48744068
Iteration 10, loss = 0.48677922
Iteration 11, loss = 0.48577613
Iteration 12, loss = 0.48484613
Iteration 13, loss = 0.48370904
Iteration 14, loss = 0.48318429
Iteration 15, loss = 0.48297755
Iteration 16, loss = 0.48197874
Iteration 17, loss = 0.48115214
Iteration 18, loss = 0.48052450
Iteration 19, loss = 0.48007558
Iteration 20, loss = 0.47951391
Iteration 21, loss = 0.47903577
Iteration 22, loss = 0.47839616
Iteration 23, loss = 0.47793688
Iteration 24, loss = 0.47769253
Iteration 25, loss = 0.47717818
Iteration 26, loss = 0.47684406
Iteration 27, loss = 0.47629084
Iteration 28, loss = 0.47602651
Iteration 29, loss = 0.47548856
Iteration 30, loss = 0.47504805
Iteration 31, loss = 0.47465371
Iteration 32, loss = 0.47421830
Iteration 33, loss = 0.47392416
Iteration 34, loss = 0.47366706
Iteration 35, loss = 0.47321522
Iteration 36, loss = 0.47318851
Iteration 37, loss = 0.47241880
Iteration 38, loss = 0.47216180
Iteration 39, loss = 0.47184123
Iteration 40, loss = 0.47178338
Iteration 41, loss = 0.47169183
Iteration 42, loss = 0.47115280
Iteration 43, loss = 0.47058632
Iteration 44, loss = 0.47074491
Iteration 45, loss = 0.47009595
Iteration 46, loss = 0.47004928
Iteration 47, loss = 0.46993091
Iteration 48, loss = 0.46917179
Iteration 49, loss = 0.46871246
Iteration 50, loss = 0.46909419
Iteration 51, loss = 0.46877284
Iteration 52, loss = 0.46804914
Iteration 53, loss = 0.46844450
Iteration 54, loss = 0.46803214
Iteration 55, loss = 0.46745846
Iteration 56, loss = 0.46715322
Iteration 57, loss = 0.46717949
Iteration 58, loss = 0.46700730
Iteration 59, loss = 0.46653581
Iteration 60, loss = 0.46644627
Iteration 61, loss = 0.46591540
```

```
Iteration 62, loss = 0.46594909
Iteration 63, loss = 0.46575490
Iteration 64, loss = 0.46536989
Iteration 65, loss = 0.46577235
Iteration 66, loss = 0.46524243
Iteration 67, loss = 0.46490995
Iteration 68, loss = 0.46485736
Iteration 69, loss = 0.46442841
Iteration 70, loss = 0.46437446
Iteration 71, loss = 0.46398872
Iteration 72, loss = 0.46442703
Iteration 73, loss = 0.46360853
Iteration 74, loss = 0.46339823
Iteration 75, loss = 0.46308871
Iteration 76, loss = 0.46286793
Iteration 77, loss = 0.46270811
Iteration 78, loss = 0.46259578
Iteration 79, loss = 0.46233204
Iteration 80, loss = 0.46226247
Iteration 81, loss = 0.46173010
Iteration 82, loss = 0.46167422
Iteration 83, loss = 0.46189441
Iteration 84, loss = 0.46126475
Iteration 85, loss = 0.46145931
Iteration 86, loss = 0.46087437
Iteration 87, loss = 0.46094959
Iteration 88, loss = 0.46039434
Iteration 89, loss = 0.46014797
Iteration 90, loss = 0.46033477
Iteration 91, loss = 0.46024311
Iteration 92, loss = 0.45997921
Iteration 93, loss = 0.45967082
Iteration 94, loss = 0.45907994
Iteration 95, loss = 0.45895664
Iteration 96, loss = 0.45940572
Iteration 97, loss = 0.45915889
Iteration 98, loss = 0.45891797
Iteration 99, loss = 0.45847444
Iteration 100, loss = 0.45886119
Iteration 101, loss = 0.45871746
Iteration 102, loss = 0.45805829
Iteration 103, loss = 0.45772903
Iteration 104, loss = 0.45866296
Iteration 105, loss = 0.45773146
Iteration 106, loss = 0.45795647
Iteration 107, loss = 0.45726038
Iteration 108, loss = 0.45789032
Iteration 109, loss = 0.45693626
Iteration 110, loss = 0.45706952
Iteration 111, loss = 0.45679228
Iteration 112, loss = 0.45638598
Iteration 113, loss = 0.45673297
Iteration 114, loss = 0.45694934
Iteration 115, loss = 0.45611257
Iteration 116, loss = 0.45639420
Iteration 117, loss = 0.45615014
Iteration 118, loss = 0.45576490
Iteration 119, loss = 0.45564153
Iteration 120, loss = 0.45565093
Iteration 121, loss = 0.45543370
Iteration 122, loss = 0.45576457
```

```
Iteration 123, loss = 0.45471208
Iteration 124, loss = 0.45516356
Iteration 125, loss = 0.45485477
Iteration 126, loss = 0.45476449
Iteration 127, loss = 0.45417761
Iteration 128, loss = 0.45453220
Iteration 129, loss = 0.45391870
Iteration 130, loss = 0.45428679
Iteration 131, loss = 0.45444043
Iteration 132, loss = 0.45358254
Iteration 133, loss = 0.45350843
Iteration 134, loss = 0.45344064
Iteration 135, loss = 0.45336926
Iteration 136, loss = 0.45296763
Iteration 137, loss = 0.45296365
Iteration 138, loss = 0.45309536
Iteration 139, loss = 0.45260042
Iteration 140, loss = 0.45227005
Iteration 141, loss = 0.45289859
Iteration 142, loss = 0.45334750
Iteration 143, loss = 0.45270164
Iteration 144, loss = 0.45288663
Iteration 145, loss = 0.45203721
Iteration 146, loss = 0.45191858
Iteration 147, loss = 0.45149123
Iteration 148, loss = 0.45167364
Iteration 149, loss = 0.45131354
Iteration 150, loss = 0.45121724
Iteration 151, loss = 0.45136694
Iteration 152, loss = 0.45177262
Iteration 153, loss = 0.45237254
Iteration 154, loss = 0.45118876
Iteration 155, loss = 0.45085391
Iteration 156, loss = 0.45066492
Iteration 157, loss = 0.45070685
Iteration 158, loss = 0.45048707
Iteration 159, loss = 0.45104006
Iteration 160, loss = 0.45071881
Iteration 161, loss = 0.45011476
Iteration 162, loss = 0.45025811
Iteration 163, loss = 0.44994081
Iteration 164, loss = 0.44982499
Iteration 165, loss = 0.45004496
Iteration 166, loss = 0.44979045
Iteration 167, loss = 0.44949088
Iteration 168, loss = 0.44993765
Iteration 169, loss = 0.44957623
Iteration 170, loss = 0.44905731
Iteration 171, loss = 0.44908916
Iteration 172, loss = 0.44884736
Iteration 173, loss = 0.44889676
Iteration 174, loss = 0.44884551
Iteration 175, loss = 0.44904190
Iteration 176, loss = 0.44882156
Iteration 177, loss = 0.44815407
Iteration 178, loss = 0.44891833
Iteration 179, loss = 0.44820507
Iteration 180, loss = 0.44808960
Iteration 181, loss = 0.44760904
Iteration 182, loss = 0.44920554
Iteration 183, loss = 0.44766855
```

```
Iteration 184, loss = 0.44831462
Iteration 185, loss = 0.44800986
Iteration 186, loss = 0.44738494
Iteration 187, loss = 0.44774977
Iteration 188, loss = 0.44713838
Iteration 189, loss = 0.44700448
Iteration 190, loss = 0.44697666
Iteration 191, loss = 0.44659582
Iteration 192, loss = 0.44693665
Iteration 193, loss = 0.44699783
Iteration 194, loss = 0.44671569
Iteration 195, loss = 0.44636847
Iteration 196, loss = 0.44635588
Iteration 197, loss = 0.44629661
Iteration 198, loss = 0.44634599
Iteration 199, loss = 0.44569481
Iteration 200, loss = 0.44614766
Iteration 201, loss = 0.44620847
Iteration 202, loss = 0.44559528
Iteration 203, loss = 0.44534444
Iteration 204, loss = 0.44567643
Iteration 205, loss = 0.44549825
Iteration 206, loss = 0.44578399
Iteration 207, loss = 0.44613533
Iteration 208, loss = 0.44540663
Iteration 209, loss = 0.44528018
Iteration 210, loss = 0.44472900
Iteration 211, loss = 0.44478270
Iteration 212, loss = 0.44481551
Iteration 213, loss = 0.44462484
Iteration 214, loss = 0.44440144
Iteration 215, loss = 0.44451867
Iteration 216, loss = 0.44435779
Iteration 217, loss = 0.44431666
Iteration 218, loss = 0.44416423
Iteration 219, loss = 0.44394249
Iteration 220, loss = 0.44377471
Iteration 221, loss = 0.44383592
Iteration 222, loss = 0.44516260
Iteration 223, loss = 0.44364211
Iteration 224, loss = 0.44419440
Iteration 225, loss = 0.44347755
Iteration 226, loss = 0.44357249
Iteration 227, loss = 0.44345833
Iteration 228, loss = 0.44279156
Iteration 229, loss = 0.44375112
Iteration 230, loss = 0.44307224
Iteration 231, loss = 0.44291541
Iteration 232, loss = 0.44254482
Iteration 233, loss = 0.44251327
Iteration 234, loss = 0.44276658
Iteration 235, loss = 0.44312722
Iteration 236, loss = 0.44258282
Iteration 237, loss = 0.44265600
Iteration 238, loss = 0.44285077
Iteration 239, loss = 0.44211748
Iteration 240, loss = 0.44232080
Iteration 241, loss = 0.44260018
Iteration 242, loss = 0.44182296
Iteration 243, loss = 0.44169549
Iteration 244, loss = 0.44149249
```

```
Iteration 245, loss = 0.44168525
Iteration 246, loss = 0.44133360
Iteration 247, loss = 0.44103861
Iteration 248, loss = 0.44188409
Iteration 249, loss = 0.44110593
Iteration 250, loss = 0.44189873
Iteration 251, loss = 0.44116284
Iteration 252, loss = 0.44093293
Iteration 253, loss = 0.44074960
Iteration 254, loss = 0.44098141
Iteration 255, loss = 0.44083418
Iteration 256, loss = 0.44091541
Iteration 257, loss = 0.44037966
Iteration 258, loss = 0.44057201
Iteration 259, loss = 0.44033274
Iteration 260, loss = 0.44008429
Iteration 261, loss = 0.44045585
Iteration 262, loss = 0.44009587
Iteration 263, loss = 0.44056081
Iteration 264, loss = 0.44081680
Iteration 265, loss = 0.44031914
Iteration 266, loss = 0.43976607
Iteration 267, loss = 0.43984674
Iteration 268, loss = 0.43955556
Iteration 269, loss = 0.43974183
Iteration 270, loss = 0.43989798
Iteration 271, loss = 0.43958918
Iteration 272, loss = 0.43958370
Iteration 273, loss = 0.43930902
Iteration 274, loss = 0.43947110
Iteration 275, loss = 0.43936144
Iteration 276, loss = 0.43890814
Iteration 277, loss = 0.43875458
Iteration 278, loss = 0.43879896
Iteration 279, loss = 0.43933236
Iteration 280, loss = 0.43851311
Iteration 281, loss = 0.43891215
Iteration 282, loss = 0.43888968
Iteration 283, loss = 0.43892452
Iteration 284, loss = 0.43840539
Iteration 285, loss = 0.43835750
Iteration 286, loss = 0.43830347
Iteration 287, loss = 0.43797909
Iteration 288, loss = 0.43794645
Iteration 289, loss = 0.43749646
Iteration 290, loss = 0.43769798
Iteration 291, loss = 0.43769876
Iteration 292, loss = 0.43767729
Iteration 293, loss = 0.43765863
Iteration 294, loss = 0.43726584
Iteration 295, loss = 0.43768609
Iteration 296, loss = 0.43813140
Iteration 297, loss = 0.43749923
Iteration 298, loss = 0.43764654
Iteration 299, loss = 0.43663132
Iteration 300, loss = 0.43728134
Iteration 301, loss = 0.43656637
Iteration 302, loss = 0.43659735
Iteration 303, loss = 0.43687521
Iteration 304, loss = 0.43647528
Iteration 305, loss = 0.43631715
```

```
Iteration 306, loss = 0.43655843
Iteration 307, loss = 0.43617117
Iteration 308, loss = 0.43627427
Iteration 309, loss = 0.43632334
Iteration 310, loss = 0.43619807
Iteration 311, loss = 0.43596359
Iteration 312, loss = 0.43651193
Iteration 313, loss = 0.43605696
Iteration 314, loss = 0.43584238
Iteration 315, loss = 0.43566049
Iteration 316, loss = 0.43533812
Iteration 317, loss = 0.43550692
Iteration 318, loss = 0.43550387
Iteration 319, loss = 0.43576480
Iteration 320, loss = 0.43530177
Iteration 321, loss = 0.43501177
Iteration 322, loss = 0.43596726
Iteration 323, loss = 0.43515113
Iteration 324, loss = 0.43541947
Iteration 325, loss = 0.43453280
Iteration 326, loss = 0.43513729
Iteration 327, loss = 0.43442673
Iteration 328, loss = 0.43461892
Iteration 329, loss = 0.43439021
Iteration 330, loss = 0.43460529
Iteration 331, loss = 0.43402549
Iteration 332, loss = 0.43406422
Iteration 333, loss = 0.43457232
Iteration 334, loss = 0.43508105
Iteration 335, loss = 0.43437074
Iteration 336, loss = 0.43440732
Iteration 337, loss = 0.43442254
Iteration 338, loss = 0.43381139
Iteration 339, loss = 0.43375981
Iteration 340, loss = 0.43421399
Iteration 341, loss = 0.43357164
Iteration 342, loss = 0.43373028
Iteration 343, loss = 0.43432599
Iteration 344, loss = 0.43353094
Iteration 345, loss = 0.43318961
Iteration 346, loss = 0.43379849
Iteration 347, loss = 0.43340701
Iteration 348, loss = 0.43335318
Iteration 349, loss = 0.43329704
Iteration 350, loss = 0.43326750
Iteration 351, loss = 0.43366012
Iteration 352, loss = 0.43303431
Iteration 353, loss = 0.43360645
Iteration 354, loss = 0.43329384
Iteration 355, loss = 0.43402302
Iteration 356, loss = 0.43310757
Iteration 357, loss = 0.43268438
Iteration 358, loss = 0.43291926
Iteration 359, loss = 0.43275938
Iteration 360, loss = 0.43255597
Iteration 361, loss = 0.43314507
Iteration 362, loss = 0.43276047
Iteration 363, loss = 0.43242658
Iteration 364, loss = 0.43248626
Iteration 365, loss = 0.43224287
Iteration 366, loss = 0.43199025
```

```
Iteration 367, loss = 0.43279481
Iteration 368, loss = 0.43282201
Iteration 369, loss = 0.43266549
Iteration 370, loss = 0.43161654
Iteration 371, loss = 0.43180304
Iteration 372, loss = 0.43192180
Iteration 373, loss = 0.43189777
Iteration 374, loss = 0.43163689
Iteration 375, loss = 0.43223497
Iteration 376, loss = 0.43295459
Iteration 377, loss = 0.43150329
Iteration 378, loss = 0.43176854
Iteration 379, loss = 0.43171138
Iteration 380, loss = 0.43107366
Iteration 381, loss = 0.43090286
Iteration 382, loss = 0.43131712
Iteration 383, loss = 0.43108997
Iteration 384, loss = 0.43092951
Iteration 385, loss = 0.43120585
Iteration 386, loss = 0.43068606
Iteration 387, loss = 0.43085873
Iteration 388, loss = 0.43063791
Iteration 389, loss = 0.43065250
Iteration 390, loss = 0.43050123
Iteration 391, loss = 0.43064201
Iteration 392, loss = 0.43044406
Iteration 393, loss = 0.43080872
Iteration 394, loss = 0.43033096
Iteration 395, loss = 0.43014741
Iteration 396, loss = 0.43012643
Iteration 397, loss = 0.43012513
Iteration 398, loss = 0.43004987
Iteration 399, loss = 0.43039645
Iteration 400, loss = 0.42984276
Iteration 401, loss = 0.42983165
Iteration 402, loss = 0.42946746
Iteration 403, loss = 0.42966390
Iteration 404, loss = 0.42901980
Iteration 405, loss = 0.43027300
Iteration 406, loss = 0.43070786
Iteration 407, loss = 0.42984516
Iteration 408, loss = 0.42971641
Iteration 409, loss = 0.42926828
Iteration 410, loss = 0.42928338
Iteration 411, loss = 0.42951891
Iteration 412, loss = 0.42872987
Iteration 413, loss = 0.42926577
Iteration 414, loss = 0.42965888
Iteration 415, loss = 0.42945861
Iteration 416, loss = 0.42976015
Iteration 417, loss = 0.42847114
Iteration 418, loss = 0.43014985
Iteration 419, loss = 0.42859646
Iteration 420, loss = 0.42941688
Iteration 421, loss = 0.42862705
Iteration 422, loss = 0.42883114
Iteration 423, loss = 0.42850986
Iteration 424, loss = 0.42827329
Iteration 425, loss = 0.42866771
Iteration 426, loss = 0.42891042
Iteration 427, loss = 0.42839212
```

```
Iteration 428, loss = 0.42835063
Iteration 429, loss = 0.43016282
Iteration 430, loss = 0.42828538
Iteration 431, loss = 0.42805686
Iteration 432, loss = 0.42953473
Iteration 433, loss = 0.42841278
Iteration 434, loss = 0.42796993
Iteration 435, loss = 0.42771917
Iteration 436, loss = 0.42748837
Iteration 437, loss = 0.42785281
Iteration 438, loss = 0.42790719
Iteration 439, loss = 0.42774604
Iteration 440, loss = 0.42740512
Iteration 441, loss = 0.42881988
Iteration 442, loss = 0.42831408
Iteration 443, loss = 0.42802160
Iteration 444, loss = 0.42779131
Iteration 445, loss = 0.42741701
Iteration 446, loss = 0.42702027
Iteration 447, loss = 0.42683362
Iteration 448, loss = 0.42727123
Iteration 449, loss = 0.42769827
Iteration 450, loss = 0.42705711
Iteration 451, loss = 0.42664947
Iteration 452, loss = 0.42678582
Iteration 453, loss = 0.42714266
Iteration 454, loss = 0.42664710
Iteration 455, loss = 0.42685452
Iteration 456, loss = 0.42621223
Iteration 457, loss = 0.42747414
Iteration 458, loss = 0.42743060
Iteration 459, loss = 0.42597071
Iteration 460, loss = 0.42675505
Iteration 461, loss = 0.42637044
Iteration 462, loss = 0.42663071
Iteration 463, loss = 0.42729856
Iteration 464, loss = 0.42616358
Iteration 465, loss = 0.42588530
Iteration 466, loss = 0.42711609
Iteration 467, loss = 0.42738576
Iteration 468, loss = 0.42607820
Iteration 469, loss = 0.42629156
Iteration 470, loss = 0.42627572
Iteration 471, loss = 0.42574781
Iteration 472, loss = 0.42567621
Iteration 473, loss = 0.42555792
Iteration 474, loss = 0.42572445
Iteration 475, loss = 0.42521726
Iteration 476, loss = 0.42612606
Iteration 477, loss = 0.42574192
Iteration 478, loss = 0.42544293
Iteration 479, loss = 0.42562029
Iteration 480, loss = 0.42501433
Iteration 481, loss = 0.42498275
Iteration 482, loss = 0.42487924
Iteration 483, loss = 0.42511551
Iteration 484, loss = 0.42562321
Iteration 485, loss = 0.42468043
Iteration 486, loss = 0.42540962
Iteration 487, loss = 0.42464322
Iteration 488, loss = 0.42528489
```

```
Iteration 489, loss = 0.42426538
Iteration 490, loss = 0.42539358
Iteration 491, loss = 0.42482498
Iteration 492, loss = 0.42495605
Iteration 493, loss = 0.42470411
Iteration 494, loss = 0.42481901
Iteration 495, loss = 0.42468480
Iteration 496, loss = 0.42488192
Iteration 497, loss = 0.42445896
Iteration 498, loss = 0.42466218
Iteration 499, loss = 0.42432414
Iteration 500, loss = 0.42409440

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural_network\multilay
er_perceptron.py:566: ConvergenceWarning: Stochastic Optimizer: Maximum it
erations (500) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

Out[220]:

```
GridSearchCV(cv=3, error_score='raise-deprecating',
             estimator=MLPClassifier(activation='relu', alpha=0.0001,
                                     batch_size='auto', beta_1=0.9,
                                     beta_2=0.999, early_stopping=False,
                                     epsilon=1e-08, hidden_layer_sizes=(10
0,),
                                     learning_rate='constant',
                                     learning_rate_init=0.001, max_iter=20
0,
                                     momentum=0.9, n_iter_no_change=10,
                                     nesterovs_momentum=True, power_t=0.5,
                                     random_state=None, shuffle=True,
                                     solver='adam', tol=0.0001,
                                     validation_fraction=0.1, verbose=Fals
e,
                                     warm_start=False),
             iid='warn', n_jobs=None,
             param_grid={'hidden_layer_sizes': [100, 300],
                         'max_iter': [300, 500], 'solver': ['adam'],
                         'tol': [0.1, 1e-06], 'verbose': ['true']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=Fals
e,
             scoring=None, verbose=0)
```

In [221]:

```
grid_search_ann.best_params_ # Best Parameters.
```

Out[221]:

```
{'hidden_layer_sizes': 100,
 'max_iter': 500,
 'solver': 'adam',
 'tol': 1e-06,
 'verbose': 'true'}
```

In [222]:

```
grid_search_ann.best_estimator_ # Best Model.
```

Out[222]:

```
MLPClassifier(activation='relu', 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=500, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=None, shuffle=True, solver='adam', tol=1e-06,
              validation_fraction=0.1, verbose='true', warm_start=False)
```

In [223]:

```
best_model_ann = grid_search_ann.best_estimator_
```

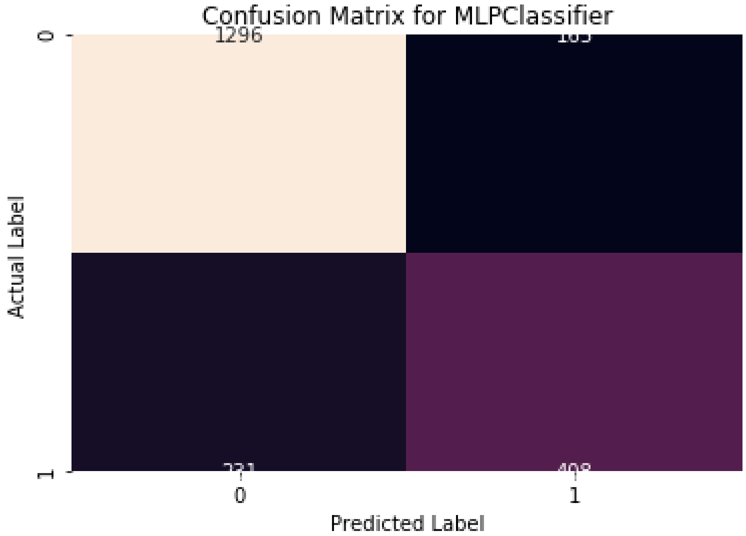
In [224]:

```
y_train_pre_ann = best_model_ann.predict (xtrain)
y_test_pre_ann = best_model_ann.predict (xtest)
```

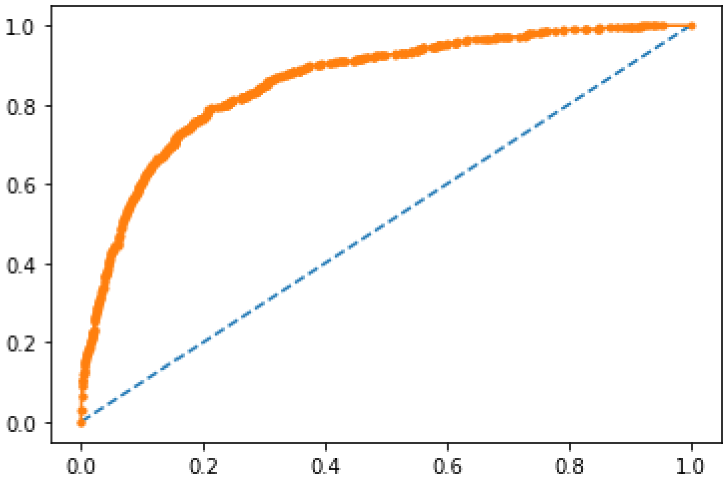
In [225]:

```
# Performance measures on Train set.

models_names={best_model_ann:'MLPClassifier'}

print('Accuracy for {} model is'.format(models_names[best_model_ann]),'\n',accuracy_sco
re(ytrain,y_train_pre_ann))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[best_model_ann]),'\n',roc_auc
_score (ytrain,y_train_pre_ann))

print('\n')

print('Classification report for {} model is'.format(models_names[best_model_ann]),'\n'
,classification_report(ytrain,y_train_pre_ann))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[best_model_ann]),'\n',conf
usion_matrix(ytrain,y_train_pre_ann))

print ('\n')


print('Confusion Matrix for {} model is'.format(models_names[best_model_ann]))

sns.heatmap(confusion_matrix(ytrain,y_train_pre_ann),annot=True,fmt='d',cbar=False)

plt.title('Confusion Matrix for {}'.format(models_names[best_model_ann]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = best_model_ann.predict_proba (xtrain)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytrain, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytrain, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for MLPClassifier model is
 0.8114285714285714


ROC AUC score for MLPClassifier model is
 0.7627806538064802


Classification report for MLPClassifier model is
               precision    recall  f1-score   support

           0       0.85      0.89      0.87      1461
           1       0.71      0.64      0.67       639

    accuracy                           0.81      2100
   macro avg       0.78      0.76      0.77      2100
weighted avg       0.81      0.81      0.81      2100



Confusion Matrix for MLPClassifier model is
 [[1296  165]
 [ 231  408]]


Confusion Matrix for MLPClassifier model is
```



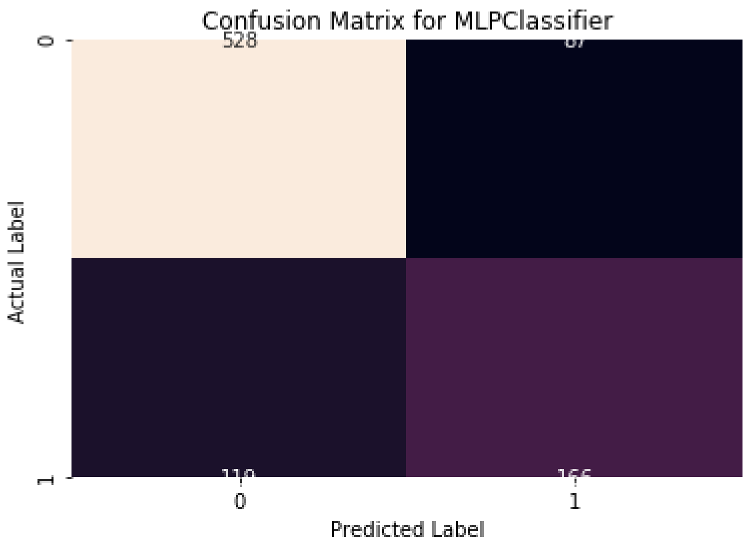
```
AUC: 0.858
```



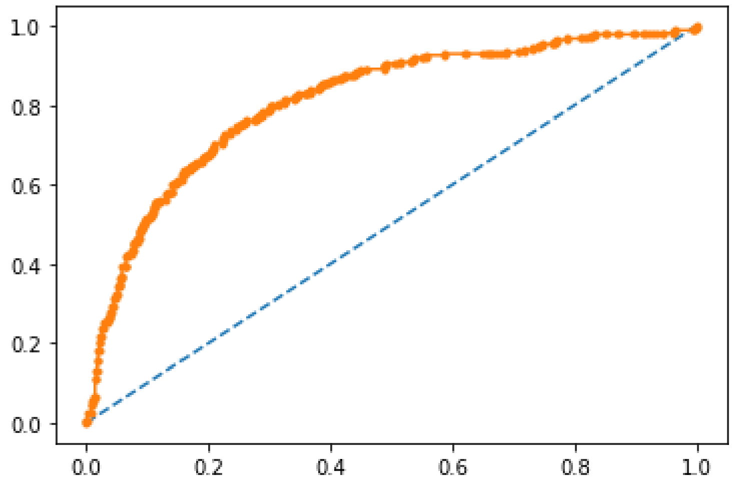
In [226]:

```
# Performance measures on Test set.

models_names={best_model_ann:'MLPClassifier'}

print('Accuracy for {} model is'.format(models_names[best_model_ann]),'\n',accuracy_sco
re(ytest,y_test_pre_ann))

print('\n')

print('ROC AUC score for {} model is'.format(models_names[best_model_ann]),'\n',roc_auc
_score (ytest,y_test_pre_ann))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[best_model_ann]),'\n',conf
usion_matrix(ytest,y_test_pre_ann))

print ('\n')


print('Classification report for {} model is'.format(models_names[best_model_ann]),'\n'
,classification_report(ytest,y_test_pre_ann))

print('\n')

print('Confusion Matrix for {} model is'.format(models_names[best_model_ann]))

sns.heatmap(confusion_matrix(ytest,y_test_pre_ann),annot=True,fmt='d',cbar=False)

plt.title('Confusion Matrix for {}'.format(models_names[best_model_ann]))
plt.xlabel('Predicted Label')
plt.ylabel('Actual Label')
plt.show()

print('\n')


# predict probabilities
probs = best_model_ann.predict_proba (xtest)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
from sklearn.metrics import roc_auc_score
auc = roc_auc_score(ytest, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(ytest, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
# show the plot
plt.show()
```

```
Accuracy for MLPClassifier model is
 0.7711111111111111


ROC AUC score for MLPClassifier model is
 0.7204963628583654


Confusion Matrix for MLPClassifier model is
 [[528  87]
 [119 166]]


Classification report for MLPClassifier model is
               precision    recall  f1-score   support

           0       0.82      0.86      0.84       615
           1       0.66      0.58      0.62       285

    accuracy                           0.77       900
   macro avg       0.74      0.72      0.73       900
weighted avg       0.77      0.77      0.77       900



Confusion Matrix for MLPClassifier model is
```




```
AUC: 0.812
```



Inferences for Artifical Neural Network (ANN):

On Training Data Set Only checking for 1's. Performance Metrics on Training set has given us Accuracy of 81 %. ROC AUC score of 76 %. Recall 64 %. Precision 71 %. f1 score is also 67 %. AUC is 85 %.

False positives (FP) are 231 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1 and TYPE 1 error is very much less than TYPE 2 in this model.

Only checking for 1's. Performance Metrics on Test set has given us Accuracy of 77 %. ROC AUC score of 72 %. Recall 58 %. Precision 66 %. f1 score is also 62 %. AUC is 81 %.

False positives (FP) are 119 means precision is more so Type 1 error is more than Type 2 and, in this problem, we should be more focus on Type 1 and TYPE 1 error is very much less than TYPE 2 in this model.

So, both the Training and Test gives us the nearby similar score for all the Performance measures.

In [238]:

```
# lets also compare the probabilities for all the models:
#:1 Classification Model (DT)
ytest_predict_prob_dt = dt_fine.predict_proba(xtest)
ytest_predict_prob_dt.mean()
```

Out[238]:

```
0.5
```

In [239]:

```
pd.DataFrame (ytest_predict_prob_dt).head()
```

Out[239]:

	0	1
0	0.753012	0.246988
1	0.397727	0.602273
2	0.912821	0.087179
3	0.962162	0.037838
4	0.650000	0.350000

In [233]:

```
# 2: Random Forest (RF)

ytest_predict_prob_rf = best_model.predict_proba(xtest)
ytest_predict_prob_rf.mean()
```

Out[233]:

```
0.5000000000000001
```

In [240]:

```
pd.DataFrame (ytest_predict_prob_rf).head()
```

Out[240]:

	0	1
0	0.790304	0.209696
1	0.468871	0.531129
2	0.789091	0.210909
3	0.867491	0.132509
4	0.725967	0.274033

In [243]:

```
# 3: Artifical Neural Netwrok.

ytest_predict_prob_ann = best_model_ann.predict_proba(xtest)
ytest_predict_prob_ann.mean()
```

Out[243]:

```
0.5
```

In [242]:

```
pd.DataFrame (ytest_predict_prob_ann).head()
```

Out[242]:

	0	1
0	0.709045	0.290955
1	0.378787	0.621213
2	0.551067	0.448933
3	0.948233	0.051767
4	0.722020	0.277980

So as per Probability all the Models have given the Average of 0.50 % as we knew when we checked the accuracy,hence AREA UNDER CURVE will be the identifier for selecting the Best Model.

2.4 Final Model: Compare all the model and write an inference which model is best/optimized.

Comparing all the Models on the basis of Performance Metrics in Structured tabular manner.

Performance Matrix	Models					
	Classification Model (DT)		Random Forest (RF)		Artificial Neural Network	
	Training	Testing	Training	Testing	Training	Testing
Accuracy	79	78	76	76	81	77
AUC	83	81	78	80	85	81
ROC AUC score	76	73	71	72	76	72
Precision						
0	0.85	0.83	0.83	0.82	0.85	0.82
1	0.67	0.66	0.61	0.64	0.71	0.66
Recall						
0	0.86	0.85	0.83	0.84	0.89	0.86
1	0.66	0.63	0.6	0.61	0.64	0.58
f1 score						
0	0.86	0.84	0.83	0.83	0.87	0.84
1	0.67	0.64	0.61	0.62	0.67	0.62
Confusion Matrix	[1253 208 215 424]	[523 92 106 179]	[1218 243 253 386]	[516 99 111 174	[1296 165 231 408]	[528 87 119 166]

Best/Optimized Model is Artificial Neural Network (ANN)

Sensitivity of the Model is directly proportional to the Opposite values in the model, in this case Our main objective is build the model which will PREDICT the customers who will CLAIMED or NOT CLAIMED.

We are mostly focus on prevention of Wrongly Claimed.

Company is more likely to predict those claims which should not be claimed but it has been claimed and then they can evaluate the Inputs.

So, this ANN model is giving us the Precision on Training and Testing Data of 71 % (only considering 1s) and when it compares to other models RF, CART gives maximum of 67 %.

As per the objective we are more focus on PRECISION i.e., on Type 1 error where our model is predicting those customers which should not be claimed but our model has been predicted as they are CLAIMED.

In [ ]:

```

```

In [ ]:

```

```

2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations.

Business Insights:

Objective of this case study was to check why there is a higher Claim frequency. By working on multiple models, we came across that although the given data was Balanced with 69-31 ratio as Not claimed and Claimed, we came across different outputs from all the 3 models after fine tuning where we were able to understand that Models have given average Accuracy of above 75 % which is good in real scenario once deploy this model. Also, all the Models had given the Accuracy for Training and Testing Data is In line with each other.

If in case company had given the Data for more towards CLAIMED people, then we should have a different output and because of that we had figure out the MODEL (ANN) with High Precision so that not claimed will also consider as claimed and based on the INPUT parameters Company can verify the actual behaviour of its INPUTS which they are giving to the Customers and later they can work on those inputs with the same model.

Let’s check the Feature importance for all the above data set, which will help the client to understand the important parameters for their high frequency of Claim.

Lets check the Feature Importance by using Decision Tree and Random Forest.

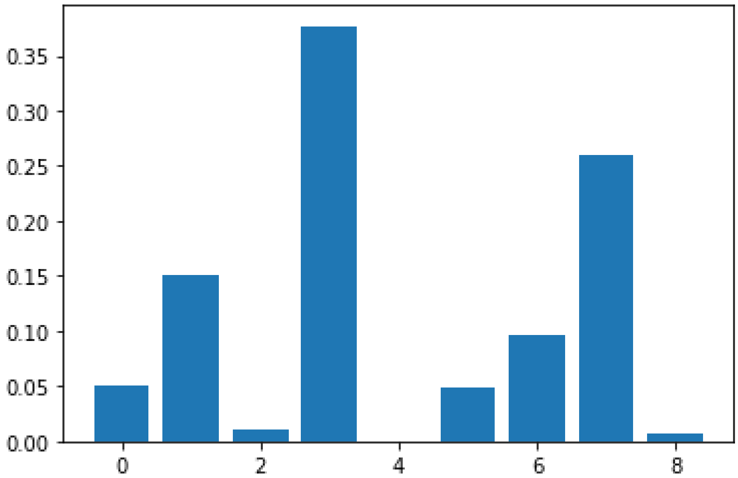
In [210]:

```
imp = dt_fine.feature_importances_
```

In [211]:

```
for i,v in enumerate(imp):
    print('Feature: %0d, Score: %.5f' % (i,v))
plt.bar([x for x in range(len(imp))], imp)
plt.show()
```

```
Feature: 0, Score: 0.04980
Feature: 1, Score: 0.15056
Feature: 2, Score: 0.01095
Feature: 3, Score: 0.37691
Feature: 4, Score: 0.00000
Feature: 5, Score: 0.04795
Feature: 6, Score: 0.09646
Feature: 7, Score: 0.26066
Feature: 8, Score: 0.00671
```



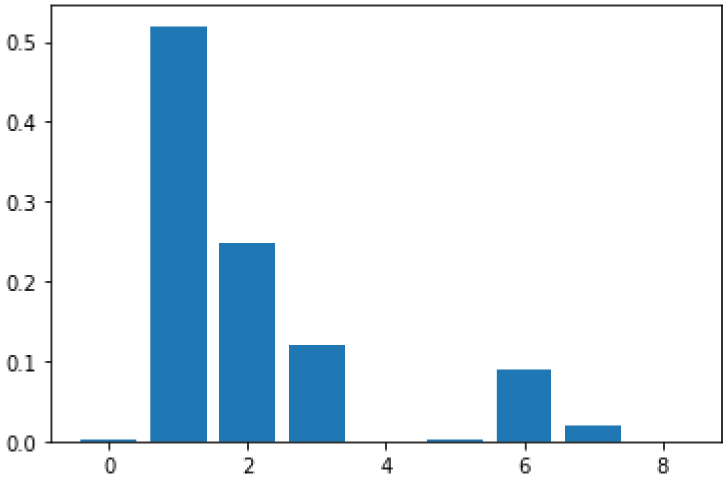
In [214]:

```
fea = best_model.feature_importances_
```

In [215]:

```
for i,v in enumerate(fea):
    print('Feature: %0d, Score: %.5f' % (i,v))
plt.bar([x for x in range(len(imp))], imp)
plt.show()
```

```
Feature: 0, Score: 0.00195
Feature: 1, Score: 0.51969
Feature: 2, Score: 0.24821
Feature: 3, Score: 0.11958
Feature: 4, Score: 0.00000
Feature: 5, Score: 0.00307
Feature: 6, Score: 0.08906
Feature: 7, Score: 0.01845
Feature: 8, Score: 0.00000
```



By checking the Feature importance and comparing for both the Models.

Common Features with high Importance are: 1: Agency Code. 2: Commision. 3: Product Name. 4: Type.

1: Lets check the Highest Agency claiming for YES.

Claimed	Yes	No	ALL
Agency_Code			
C2B	560	364	924
CWT	141	331	472
EPX	193	1172	1365
JZI	30	209	239
ALL	924	2076	3000

So the AGENCY name "C2B" is the highest agency which are claiming YES and with almost 61 % of total claim they receive. Management needs to check whether this agency is following complete SOP's.

2: Agency EPX does not get any commision and C2B is the highest Commission gainer of 924 times which proves that Agency_Code and Commission both are directly dependent.

3: Product Name:

Product Name	Bronze	Cancellation	Customised	Gold	Silver	Total
Agency_Code						
C2B	392	0	60	60	412	924
CWT	19	0	389	49	15	472
EPX	0	678	687	0	0	1365
JZI	239	0	0	0	0	239
ALL	650	678	1136	109	427	3000

So agency C2B is the highest seller of the GOLD plan which gives us confirmation that this GOLD plan is the Highest commision and then also it concludes that Gold, Silver and Bronze are the most important plans which gets highest commission and also same have they higher claim rate.

4: Type:

Type	Airlines	Travel Agency	ALL
Agency_Code			
C2B	924	0	924
CWT	0	472	472
EPX	0	1365	1365
JZI	239	0	239
ALL	1163	1837	3000

So above calculation shows that Type of Booking is also high with 100 % booking rate with Airlines itself for the agency C2B. Which conclude that if booking is done with the Airlines, they are getting the maximum claimed and with Agency there are 0 with the specific agency C2B.

So agency C2B is directly having insurance from Airlines itself hence they are getting the high claimed rate as compared to the other and when we compared the other 2 agencies which are CWT and EPX they both have "0" insurance from Airlines and hence their claimed statues for YES is also very low.

Recommendations:

Management needs to check the Agency C2B whether they are following all the SOP's from the company's book.

Almost 61 % they have the positive claimed status which means in future also they will be having the same rate for the claim status.

And that is the only agency which is the highest seller of the GOLD plan (which I believe is the highest covering plan) and will have the high price for the same hence the same agency is also having the highest Commission.

Also, Management team needs to TRAIN the other Agencies for more focusing on the other plans so that Claimed status will bring down...

And almost 80 % are positive claimed when people booked via Airlines hence management also needs to more focus on how to increase the Agency insurance type rather than Airlines which in future will bring down the Claim's.

```
Thanks,

Romil Totade
```

In [ ]:

```

```