Step 1: Import required packages

In [5]:

```
import pandas as pd
import numpy as np
import time
import seaborn as sns
from numpy import arange
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import cross_val_predict
from itertools import cycle
from catboost import CatBoostClassifier
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.decomposition import PCA
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier as RFC
from sklearn.linear_model import LogisticRegression as LR
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_auc_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.metrics import balanced_accuracy_score
from sklearn.model_selection import RepeatedStratifiedKFold
from xgboost import XGBClassifier
from sklearn.svm import SVC
import warnings
warnings.filterwarnings('ignore')
```

Step 2: Data import and check it

In [6]:

```
train=pd.read_csv(r"C:\Users\wannian\Desktop\5243\train.csv")
test_=pd.read_csv(r"C:\Users\wannian\Desktop\5243\test.csv")
```

In [7]:

```
train.head()
```

Out[7]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt
0	842	0	2.2	0	1	0	7	0.6	188
1	1021	1	0.5	1	0	1	53	0.7	136
2	563	1	0.5	1	2	1	41	0.9	145
3	615	1	2.5	0	0	0	10	0.8	131
4	1821	1	1.2	0	13	1	44	0.6	141

5 rows × 21 columns

In [92]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

In [93]:

```
test_.head()
```

Out[93]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_
0	1	1043	1	1.8	1	14	0	5	0.1	1
1	2	841	1	0.5	1	4	1	61	0.8	1
2	3	1807	1	2.8	0	1	0	27	0.9	1
3	4	1546	0	0.5	1	18	1	25	0.5	
4	5	1434	0	1.4	0	11	1	49	0.5	1

5 rows × 21 columns

In [94]:

```
test_.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             1000 non-null   int64  
 1   battery_power  1000 non-null   int64  
 2   blue           1000 non-null   int64  
 3   clock_speed    1000 non-null   float64
 4   dual_sim       1000 non-null   int64  
 5   fc             1000 non-null   int64  
 6   four_g         1000 non-null   int64  
 7   int_memory     1000 non-null   int64  
 8   m_dep          1000 non-null   float64
 9   mobile_wt      1000 non-null   int64  
 10  n_cores        1000 non-null   int64  
 11  pc             1000 non-null   int64  
 12  px_height      1000 non-null   int64  
 13  px_width       1000 non-null   int64  
 14  ram            1000 non-null   int64  
 15  sc_h           1000 non-null   int64  
 16  sc_w           1000 non-null   int64  
 17  talk_time      1000 non-null   int64  
 18  three_g        1000 non-null   int64  
 19  touch_screen   1000 non-null   int64  
 20  wifi           1000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

Step 3: Data cleaning

1.check the balance of data

In [95]:

```
train.price_range.value_counts()
```

Out[95]:

```
3    500
2    500
1    500
0    500
Name: price_range, dtype: int64
```

2.Drop or fill the Nan, Drop the duplicated data

In [96]:

```
train=train.dropna(axis=0)
train.drop_duplicates(inplace=True)
train.index=range(train.shape[0])
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

In [97]:

```
train.head()
```

Out[97]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt
0	842	0	2.2	0	1	0	7	0.6	188
1	1021	1	0.5	1	0	1	53	0.7	136
2	563	1	0.5	1	2	1	41	0.9	145
3	615	1	2.5	0	0	0	10	0.8	131
4	1821	1	1.2	0	13	1	44	0.6	141

5 rows × 21 columns

Step 4: Exploratory Data Analysis

In [98]:

```
train.describe()
```

Out[98]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_m
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.(
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.(
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.1
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.(
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.(
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.(
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.(
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.(

8 rows × 21 columns

In [178]:

```
train.rename(columns={"blue":"bluetooth", "fc":'front_camera',
            'int_memory':"internal_memory", 'm_dep':"mobile_depth",
            'pc':"primary_camera", 'talk_time':'battery_time'})
```

Out[178]:

	battery_power	bluetooth	clock_speed	dual_sim	front_camera	four_g	internal_memo
0	842	0	2.2	0	1	0	
1	1021	1	0.5	1	0	1	
2	563	1	0.5	1	2	1	
3	615	1	2.5	0	0	0	
4	1821	1	1.2	0	13	1	
...	...	...	...	...	...	...	
1995	794	1	0.5	1	0	1	
1996	1965	1	2.6	1	0	0	
1997	1911	0	0.9	1	1	1	
1998	1512	0	0.9	0	4	1	
1999	510	1	2.0	1	5	1	

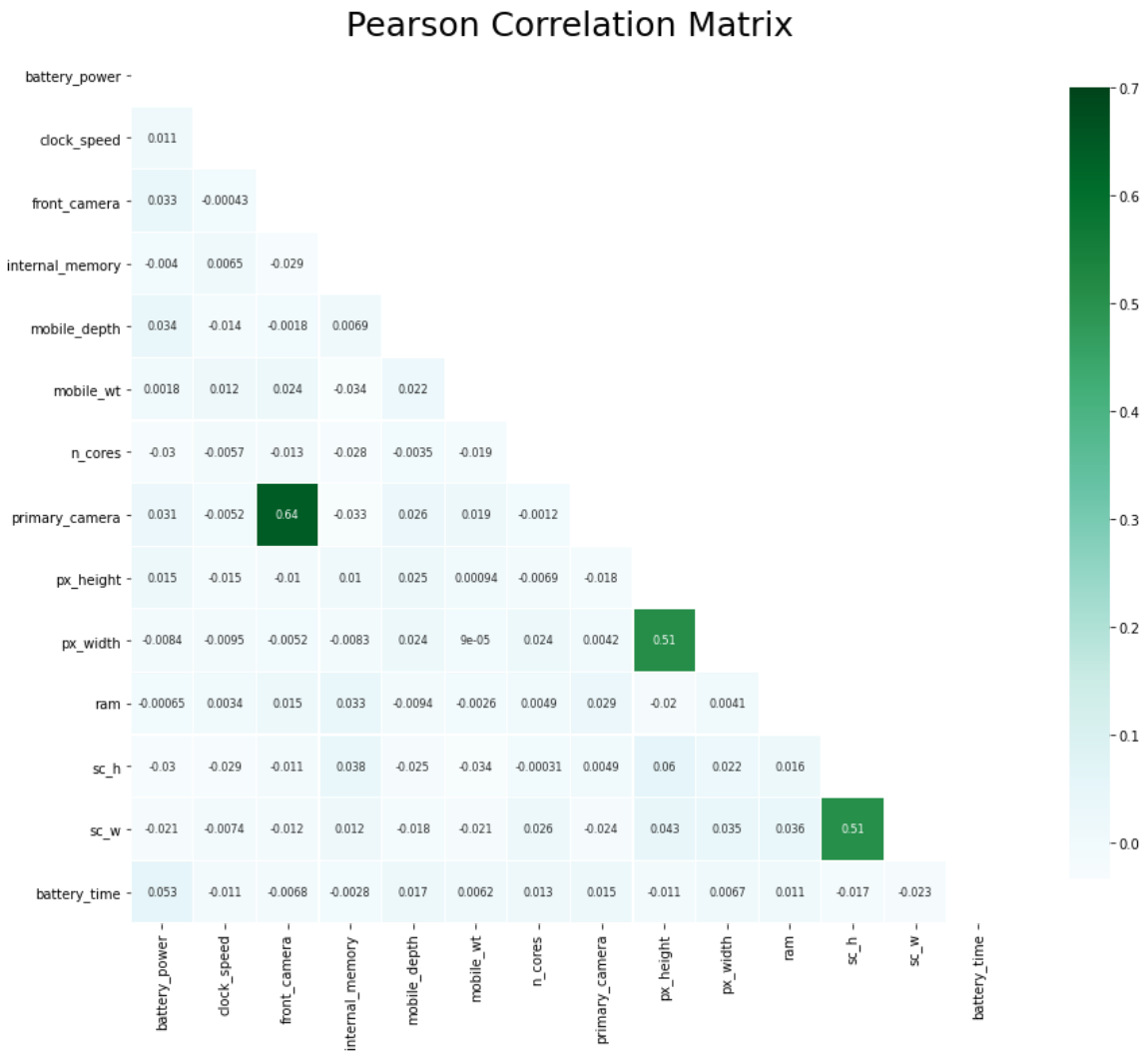
2000 rows × 21 columns

1. Correlation between numerical feature

- The primary camera mega pixels and the front camera mega pixels are highly correlated
- The pixel resolution height and pixel resolution width are highly correlated
- The screen height and screen width are highly correlated
- Battery is uncorrelated with all other numerical variables in general

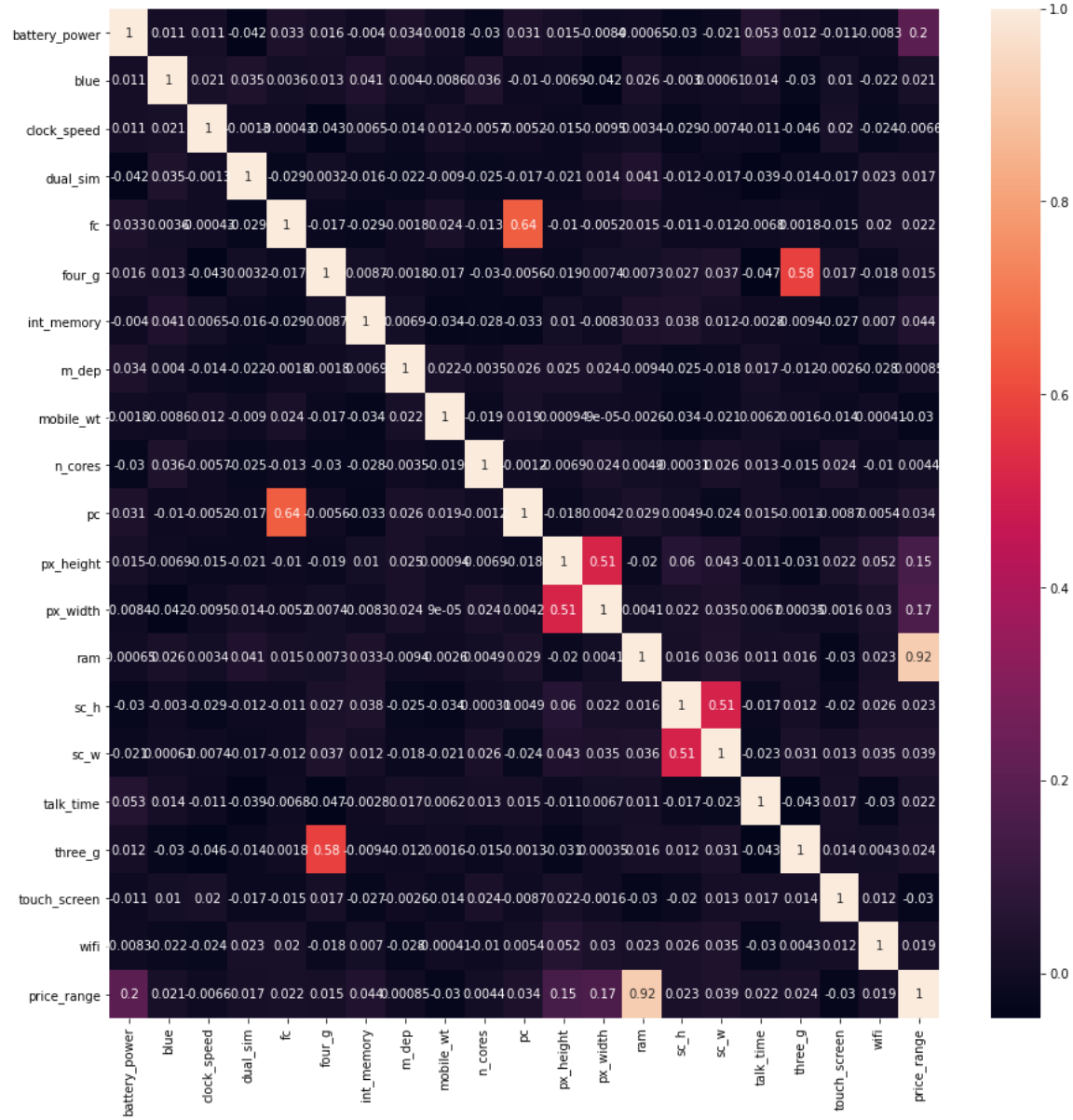
In [163]:

```
features = ['battery_power', 'clock_speed', 'front_camera',
            'internal_memory', 'mobile_depth', 'mobile_wt', 'n_cores', 'primary_camera'
,
            'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'battery_time']

mask = np.zeros_like(train[features].corr(), dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

f, ax = plt.subplots(figsize=(16, 12))
plt.title('Pearson Correlation Matrix',fontsize=25)

sns.heatmap(train[features].corr(),linewidths=0.25,vmax=0.7,square=True,cmap="BuGn",
            linecolor='w',annot=True,annot_kws={"size":8},mask=mask,cbar_kws={"shrink":
.9})
plt.show()
```



In [8]:

```
corrmat = train.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(15,15))
#plot heat map
g=sns.heatmap(train[top_corr_features].corr(),annot=True,cmap="rocket")
```



2. price_range vs. ram

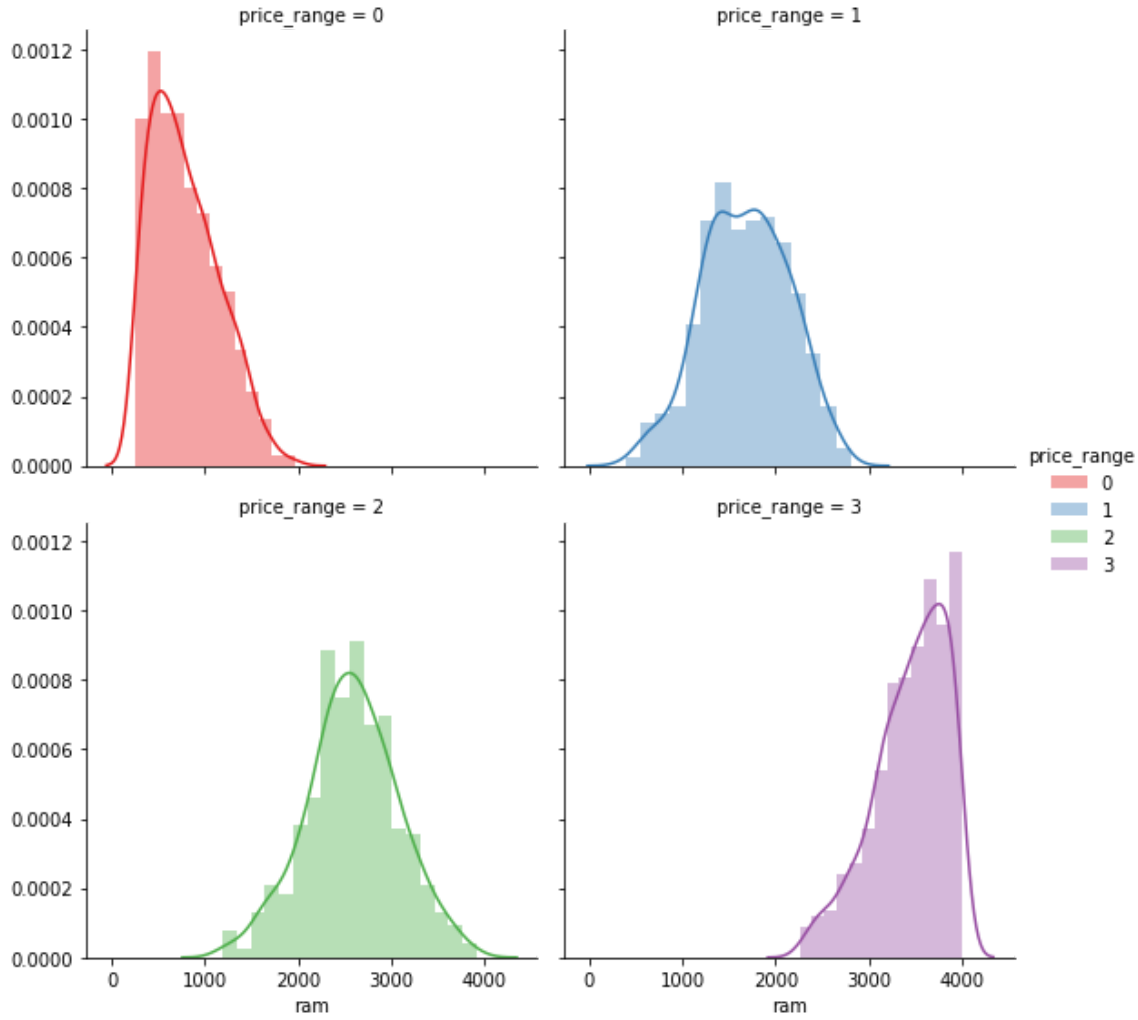
- The distribution curve are changing form right skew to left skew as the price range getting higher, which means higher price range corresponding to higher RAM

In [167]:

```
g = sns.FacetGrid(train, col="price_range", hue="price_range",
                  palette="Set1", col_wrap=2, height=4)
g = (g.map(sns.distplot, "ram").add_legend())
g
```

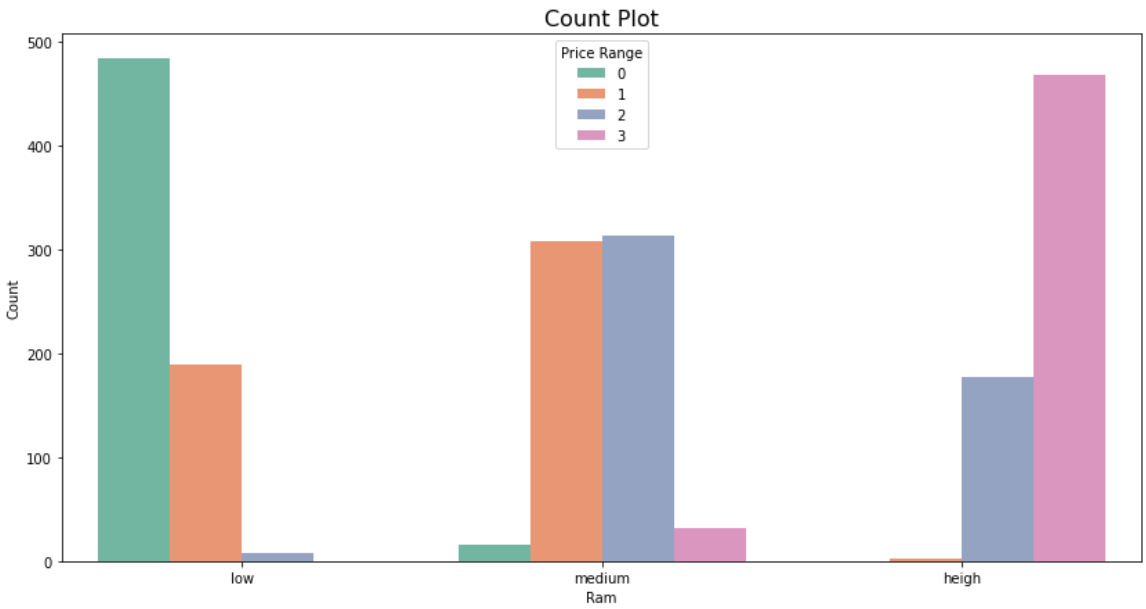
Out[167]:

```
<seaborn.axisgrid.FacetGrid at 0x2adc704bdf0>
```



In [170]:

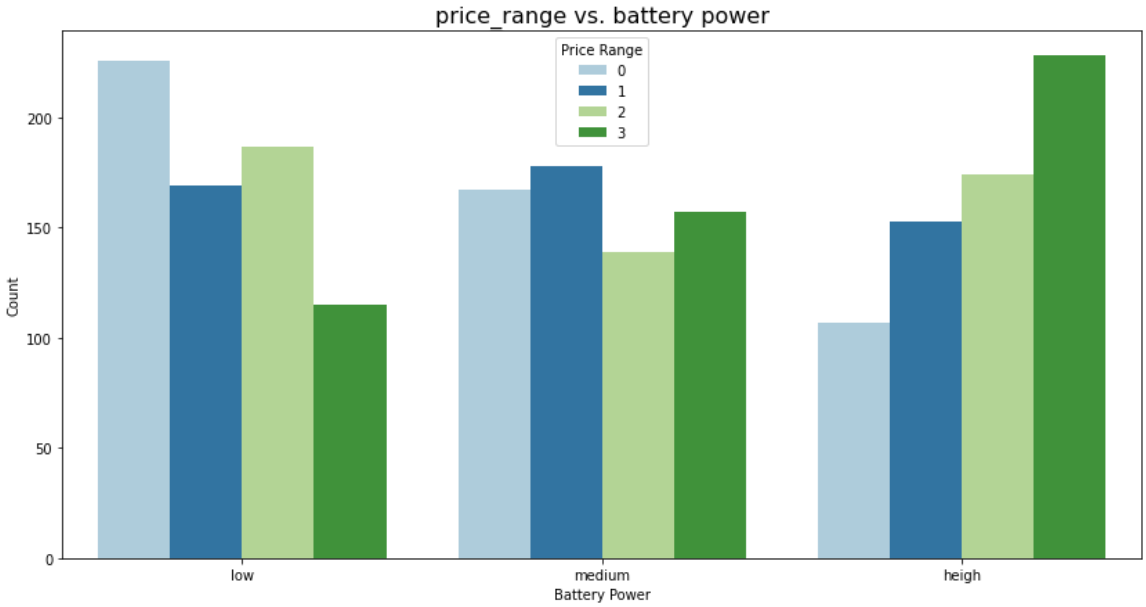
```
df_cut = pd.DataFrame()
df_cut["Ram"] = pd.cut(train["ram"],3,labels=["low","medium","heigh"])
df_cut["Price Range"] = train["price_range"]

plt.figure(figsize=(14,7))
sns.countplot(data=df_cut,x="Ram",hue="Price Range",palette="Set2")
plt.title("Count Plot",fontsize=16)
plt.xlabel("Ram")
plt.ylabel("Count")
plt.show()
```



3. price_range vs. battery_power

In [172]:

```
df_cut["Battery Power"] = pd.cut(train["battery_power"],3,labels=["low","medium","heig
h"])
plt.figure(figsize=(14,7))
sns.countplot(data=df_cut,x="Battery Power",hue="Price Range",palette="Paired")
plt.title("price_range vs. battery power",fontsize=16)
plt.xlabel("Battery Power")
plt.ylabel("Count")
plt.show()
```



4.Scale columns

In [179]:

```
standardScaler = StandardScaler()
columns_to_scale = ['battery_power', 'clock_speed', 'internal_memory', 'mobile_wt', 'px
_height','px_width','ram','sc_h','sc_w','battery_time']
train[columns_to_scale] = standardScaler.fit_transform(train[columns_to_scale])
```

In [180]:

```
train
```

Out[180]:

	battery_power	bluetooth	clock_speed	dual_sim	front_camera	four_g	internal_memo
0	-0.902597	0	0.830779	0	1	0	-1.3806
1	-0.495139	1	-1.253064	1	0	1	1.1550
2	-1.537686	1	-1.253064	1	2	1	0.4935
3	-1.419319	1	1.198517	0	0	0	-1.2152
4	1.325906	1	-0.395011	0	13	1	0.6589
...	...	...	...	...	...	...	
1995	-1.011860	1	-1.253064	1	0	1	-1.6562
1996	1.653694	1	1.321096	1	0	0	0.3832
1997	1.530773	0	-0.762748	1	1	1	0.2179
1998	0.622527	0	-0.762748	0	4	1	0.7691
1999	-1.658331	1	0.585621	1	5	1	0.7140

2000 rows × 21 columns

5.Checking the data distribution

In [190]:

```
train=pd.read_csv(r"C:\Users\wannian\Desktop\5243\train.csv")
```

In [191]:

```
y=train.iloc[:,-1]
y
```

Out[191]:

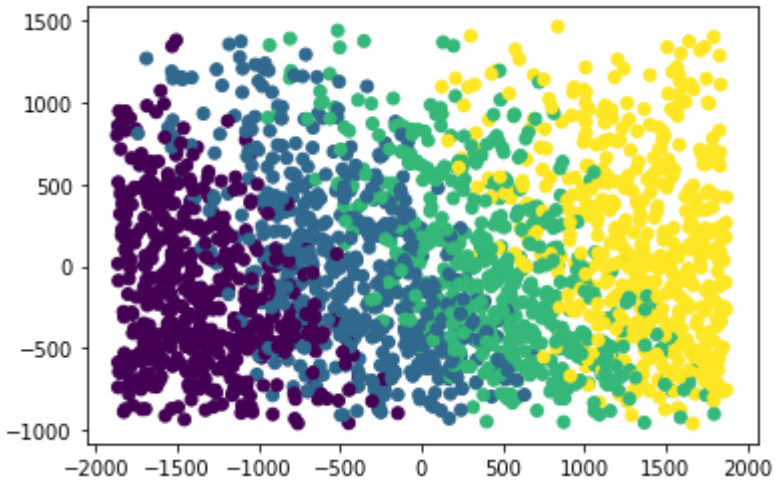
```
0       1
1       2
2       2
3       2
4       1
       ..
1995    0
1996    2
1997    3
1998    0
1999    3
Name: price_range, Length: 2000, dtype: int64
```

In [187]:

```
train=train.drop(columns='price_range')
```

In [189]:

```
x=PCA(2).fit_transform(train)
plt.scatter(x[:,0],x[:,1],c=y)
plt.show()
```



Step 5 future selection and comparasion

In [23]:

```
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier as RFC
```

In [39]:

```
RFC=RFC(n_estimators=100,random_state=0)
train_embedded=SelectFromModel(RFC).fit_transform(train,y)
```

In [42]:

```
train_embedded.shape
```

Out[42]:

```
(2000, 4)
```

Step 6 model building

Decision tree and Random Forest

In [43]:

```
xtrain1,xtest1,ytrain1,ytest1=train_test_split(train_embedded,y,test_size=0.3)
```

In [44]:

```
clf=clf.fit(xtrain1,ytrain1)
RFC=RFC.fit(xtrain1,ytrain1)

score_c1=clf.score(xtest1,ytest1)
score_r1=RFC.score(xtest1,ytest1)

print("single tree:{}".format(score_c1),
     "random forest:{}".format(score_r1))
```

```
single tree:0.8433333333333334 random forest:0.9016666666666666
```

In [30]:

```
xtrain,xtest,ytrain,ytest=train_test_split(train,y,test_size=0.3)
ytrain
```

Out[30]:

```
1263    0
1420    0
1608    1
1068    0
1226    3
       ..
199     1
604     1
873     0
1831    1
1899    0
Name: price_range, Length: 1400, dtype: int64
```

In [27]:

```
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
```

In [34]:

```
clf=DecisionTreeClassifier(random_state=0)

clf=clf.fit(xtrain,ytrain)
RFC=RFC.fit(xtrain,ytrain)

score_c=clf.score(xtest,ytest)
score_r=RFC.score(xtest,ytest)

print("single tree:{}".format(score_c),
     "random forest:{}".format(score_r))
```

```
single tree:0.82 random forest:0.8666666666666667
```

In [35]:

```
rfc_s=cross_val_score(RFC,train,y,cv=10)
```

In [37]:

```
rfc_s.mean()
```

Out[37]:

```
0.8800000000000001
```

logistic Regression

In [25]:

```
from sklearn.linear_model import LogisticRegression as LR
from sklearn.metrics import accuracy_score
```

In [26]:

```
lrl2=LR(penalty="l2",solver="liblinear",C=0.5,max_iter=1000)
```

In [46]:

```
accuracy_score(lrl2.predict(xtest),ytest)
```

Out[46]:

```
0.7533333333333333
```

In [45]:

```
lrl2=lrl2.fit(xtrain,ytrain)
lrl2.coef_
```

Out[45]:

```
array([[-2.30619876e-03,  1.50272300e-01,  6.48241438e-01,
         2.19360973e-01, -3.21805525e-02,  1.76064680e-01,
         1.53093171e-02,  1.19642430e-01,  2.79606741e-02,
         3.04557881e-01,  6.46320787e-02, -2.16604644e-03,
        -5.39148157e-04, -5.88477704e-03,  1.89138251e-01,
        -5.82270406e-03,  8.16318950e-02,  2.45821654e-01,
         2.60737856e-01,  2.20628438e-01],
       [-3.17189382e-05, -1.03588450e-01, -1.11222126e-01,
         6.88406889e-02,  1.11148723e-02, -8.64698342e-04,
         9.05728400e-04,  1.42019334e-01,  2.09112156e-04,
        -6.04191736e-02, -3.47706157e-03,  2.96266753e-04,
        -1.77357576e-04, -5.32056266e-04,  9.02481330e-03,
        -1.61006456e-02,  1.31586602e-02, -4.52857495e-03,
         2.26608701e-01, -2.11360684e-02],
       [-1.96579392e-04, -5.56314251e-02, -4.35109471e-02,
        -1.18521218e-01,  1.74122689e-02, -1.96405659e-01,
        -6.21363524e-03, -2.19133281e-01,  2.40325618e-03,
         1.80300092e-02, -1.10328142e-02,  3.84873249e-05,
        -4.30032338e-05,  5.20291795e-04, -3.30464216e-02,
         2.74834590e-03, -8.07185671e-03,  1.81517311e-01,
        -2.66287536e-01, -8.85159234e-02],
       [ 1.44551541e-03, -1.26066181e-01, -2.48525428e-01,
        -2.41812602e-01, -1.16318384e-02,  7.31570425e-02,
        -3.82559996e-03, -5.30917764e-01, -2.54267505e-02,
        -1.08189541e-01, -1.86153539e-02,  1.27762489e-03,
         5.07172429e-04,  3.36055755e-03, -5.45159502e-02,
         2.31462141e-02, -3.89219756e-02, -5.02826308e-01,
        -6.64536156e-03, -2.31488158e-01]])
```

In [49]:

```
lrl3=lrl2.fit(xtrain1,ytrain1)
lrl3.coef_
```

Out[49]:

```
array([[-1.04431181e-03, -1.42220808e-03,  5.68872641e-06,
        -3.68529420e-03],
       [-9.46264688e-05,  1.54424897e-04, -3.38216077e-05,
        -5.13351374e-04],
       [-3.98380699e-04,  1.87464903e-04, -4.56954778e-04,
         4.26495158e-04],
       [ 8.91652244e-04,  8.01662326e-04,  1.80911551e-04,
         2.41520446e-03]])
```

In [50]:

```
accuracy_score(lrl3.predict(xtest1),ytest1)
```

Out[50]:

```
0.7383333333333333
```

Dense neural network

In [51]:

```
import tensorflow.compat.v1 as tf
tf.disable_eager_execution()
```

In [137]:

```
num_imput=20
num_class=4
```

In [138]:

```
from sklearn.preprocessing import OneHotEncoder

result=OneHotEncoder(categories='auto').fit_transform(ytrain.values.reshape(-1,1))

ytrain1=result.toarray()
ytrain1
```

Out[138]:

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

In [140]:

```
##def generate_batch(self):
##    features_placeholder = tf.placeholder(self.features.dtype, self.features.shape)
##    labels_placeholder = tf.placeholder(self.labels.dtype, self.labels.shape)
##    dataset = tf.data.Dataset.from_tensor_slices((self.features, self.labels))
##    dataset = dataset.repeat(100)
##    batched_dataset = dataset.batch(100)
##    iterator = batched_dataset.make_initializable_iterator()
##    batch_xs, batch_ys = iterator.get_next()
##    return iterator.initializer,batch_xs, batch_ys
```

In [141]:

```
result1=OneHotEncoder(categories='auto').fit_transform(ytest.values.reshape(-1,1))

ytest1=result1.toarray()
ytest1
```

Out[141]:

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

In [142]:

```
x=tf.placeholder(tf.float32,[None,num_imput])
y=tf.placeholder(tf.float32,[None,num_class])
```

In [143]:

```
n_neurons={
    1:512,
    2:256,
    3:256,
    4:128
}
weights={
    'h1':tf.Variable(tf.random_normal([num_imput,n_neurons[1]])),
    'h2':tf.Variable(tf.random_normal([n_neurons[1],n_neurons[2]])),
    'h3':tf.Variable(tf.random_normal([n_neurons[2],n_neurons[3]])),
    'h4':tf.Variable(tf.random_normal([n_neurons[3],n_neurons[4]])),
}
biases={
    'b1':tf.Variable(tf.random_normal([n_neurons[1]])),
    'b2':tf.Variable(tf.random_normal([n_neurons[2]])),
    'b3':tf.Variable(tf.random_normal([n_neurons[3]])),
    'b4':tf.Variable(tf.random_normal([n_neurons[4]])),
    'out':tf.Variable(tf.random_normal([num_class])),
}
```

In [144]:

```
learning_rate=0.01
num_steps=3000
batch_size=50
display_step=100
```

In [148]:

```
def evaluate(logits):
    prediction=tf.nn.softmax(logits)
    cost_function=tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(logits=logits,labels=y))

    optimizer=tf.train.AdamOptimizer(learning_rate=learning_rate)
    train_op=optimizer.minimize(cost_function)


    correct_pred=tf.equal(tf.argmax(prediction,1),tf.argmax(y,1))
    accuracy=tf.reduce_mean(tf.cast(correct_pred,tf.float32))

    with tf.Session() as sess:
        sess.run(tf.global_variables_initializer())
        for step in range(1,num_steps):
            batch_x,batch_y=xtrain,ytrain1
            sess.run(train_op,feed_dict={x:batch_x,y:batch_y})
            if(step%display_step==0 or step==1):
                loss,acc=sess.run([cost_function,accuracy],
                                  feed_dict={x:batch_x,y:batch_y}
                )
                print('step'+str(step)\
                +',loss='+'{:.4f}'.format(loss)\
                +',Training accuracy='+'{:.3f}'.format(acc))
        print("optimization finished!!!!")

        print("testing accuracy:",\
             sess.run(accuracy,feed_dict={x:xtest,
                                          y:ytest1}))
```

In [149]:

```
def neural_net(x,n_layer):
    print("--------vanilla NN with {0} layers".format(n_layer))
    layer_1=tf.add(tf.matmul(x,weights['h1']),biases['b1'])
    layer_2=tf.add(tf.matmul(layer_1,weights['h2']),biases['b2'])
    layer_3=tf.add(tf.matmul(layer_2,weights['h3']),biases['b3'])
    layer_4=tf.add(tf.matmul(layer_3,weights['h4']),biases['b4'])
    hidden_layers={
        1:layer_1,
        2:layer_2,
        3:layer_3,
        4:layer_4
    }
    out_weight=tf.Variable(tf.random_normal([n_neurons[n_layer],num_class]))
    out_layer=tf.matmul(hidden_layers[n_layer],out_weight)+biases['out']
    return out_layer
```

In [150]:

```
evaluate(neural_net(x,4))
```

```
--------vanilla NN with 4 layers
step1,loss=603423488.0000,Training accuracy=0.254
step100,loss=18374170.0000,Training accuracy=0.545
step200,loss=2782928.7500,Training accuracy=0.613
step300,loss=1890129.2500,Training accuracy=0.618
step400,loss=1321193.0000,Training accuracy=0.579
step500,loss=1079764.6250,Training accuracy=0.600
step600,loss=839292.8750,Training accuracy=0.563
step700,loss=845149.8125,Training accuracy=0.583
step800,loss=553842.3125,Training accuracy=0.566
step900,loss=355896.7188,Training accuracy=0.661
step1000,loss=428584.7188,Training accuracy=0.614
step1100,loss=156746.5156,Training accuracy=0.687
step1200,loss=195476.2969,Training accuracy=0.673
step1300,loss=308134.2500,Training accuracy=0.640
step1400,loss=123685.1328,Training accuracy=0.680
step1500,loss=144849.6875,Training accuracy=0.674
step1600,loss=194594.0625,Training accuracy=0.631
step1700,loss=119888.4453,Training accuracy=0.629
step1800,loss=78849.5625,Training accuracy=0.685
step1900,loss=144444.2344,Training accuracy=0.566
step2000,loss=79665.4141,Training accuracy=0.694
step2100,loss=156795.9688,Training accuracy=0.544
step2200,loss=94884.8984,Training accuracy=0.635
step2300,loss=52084.8008,Training accuracy=0.701
step2400,loss=78536.8906,Training accuracy=0.664
step2500,loss=56233.6328,Training accuracy=0.666
step2600,loss=43862.7383,Training accuracy=0.691
step2700,loss=41918.5273,Training accuracy=0.646
step2800,loss=39537.7773,Training accuracy=0.671
step2900,loss=31148.2207,Training accuracy=0.696
optimization finished!!!!
testing accuracy: 0.5683333
```

In [151]:

```
def neural_net_relu(x,n_layer):

    print("--------vanilla NN + RELU with {0} layers".format(n_layer))
    layer_1=tf.add(tf.matmul(x,weights['h1']),biases['b1'])
    relu_1=tf.nn.relu(layer_1)

    layer_2=tf.add(tf.matmul(layer_1,weights['h2']),biases['b2'])
    relu_2=tf.nn.relu(layer_2)

    layer_3=tf.add(tf.matmul(layer_2,weights['h3']),biases['b3'])
    relu_3=tf.nn.relu(layer_3)

    layer_4=tf.add(tf.matmul(layer_3,weights['h4']),biases['b4'])
    hidden_layers={
        1:layer_1,
        2:layer_2,
        3:layer_3,
        4:layer_4
    }
    out_weight=tf.Variable(tf.random_normal([n_neurons[n_layer],num_class]))
    out_layer=tf.matmul(hidden_layers[n_layer],out_weight)+biases['out']
    return out_layer
```

In [152]:

```
evaluate(neural_net_relu(x,3))
```

```
--------vanilla NN + RELU with 3 layers
step1,loss=21570308.0000,Training accuracy=0.249
step100,loss=1053488.2500,Training accuracy=0.551
step200,loss=387603.1562,Training accuracy=0.615
step300,loss=462059.6875,Training accuracy=0.547
step400,loss=931824.6250,Training accuracy=0.389
step500,loss=220626.7500,Training accuracy=0.615
step600,loss=339120.5312,Training accuracy=0.513
step700,loss=211108.0625,Training accuracy=0.608
step800,loss=144494.9531,Training accuracy=0.553
step900,loss=108053.5781,Training accuracy=0.624
step1000,loss=95483.2422,Training accuracy=0.629
step1100,loss=120313.2109,Training accuracy=0.562
step1200,loss=57405.5781,Training accuracy=0.640
step1300,loss=37385.1406,Training accuracy=0.691
step1400,loss=58808.1367,Training accuracy=0.621
step1500,loss=45757.1562,Training accuracy=0.582
step1600,loss=35913.9414,Training accuracy=0.660
step1700,loss=37192.6172,Training accuracy=0.584
step1800,loss=27436.3750,Training accuracy=0.656
step1900,loss=26313.3262,Training accuracy=0.665
step2000,loss=15038.0771,Training accuracy=0.706
step2100,loss=27312.9805,Training accuracy=0.609
step2200,loss=13438.1387,Training accuracy=0.688
step2300,loss=13030.2285,Training accuracy=0.658
step2400,loss=11375.2637,Training accuracy=0.656
step2500,loss=26430.6621,Training accuracy=0.601
step2600,loss=9930.7490,Training accuracy=0.686
step2700,loss=9905.3047,Training accuracy=0.664
step2800,loss=9624.3477,Training accuracy=0.717
step2900,loss=14631.8398,Training accuracy=0.609
optimization finished!!!!
testing accuracy: 0.685
```

In [153]:

```
evaluate(neural_net_relu(x,2))
```

```
--------vanilla NN + RELU with 2 layers
step1,loss=645608.0000,Training accuracy=0.356
step100,loss=54041.4219,Training accuracy=0.536
step200,loss=24549.9707,Training accuracy=0.601
step300,loss=26692.0801,Training accuracy=0.626
step400,loss=168032.0938,Training accuracy=0.558
step500,loss=37863.0703,Training accuracy=0.562
step600,loss=25533.3164,Training accuracy=0.603
step700,loss=58800.8008,Training accuracy=0.507
step800,loss=15036.0342,Training accuracy=0.662
step900,loss=13926.8799,Training accuracy=0.690
step1000,loss=28245.5664,Training accuracy=0.638
step1100,loss=20332.0176,Training accuracy=0.671
step1200,loss=10378.0586,Training accuracy=0.701
step1300,loss=11559.9297,Training accuracy=0.682
step1400,loss=11865.9258,Training accuracy=0.679
step1500,loss=24589.2969,Training accuracy=0.591
step1600,loss=25835.3145,Training accuracy=0.592
step1700,loss=16754.4102,Training accuracy=0.670
step1800,loss=26252.0938,Training accuracy=0.592
step1900,loss=23517.8262,Training accuracy=0.583
step2000,loss=12807.6084,Training accuracy=0.679
step2100,loss=23629.1172,Training accuracy=0.533
step2200,loss=13058.5010,Training accuracy=0.653
step2300,loss=17072.5332,Training accuracy=0.626
step2400,loss=12609.5801,Training accuracy=0.611
step2500,loss=14390.5615,Training accuracy=0.652
step2600,loss=23631.9395,Training accuracy=0.571
step2700,loss=25755.6719,Training accuracy=0.601
step2800,loss=22507.2305,Training accuracy=0.583
step2900,loss=9125.9150,Training accuracy=0.682
optimization finished!!!!
testing accuracy: 0.73833334
```

Naive Bayes

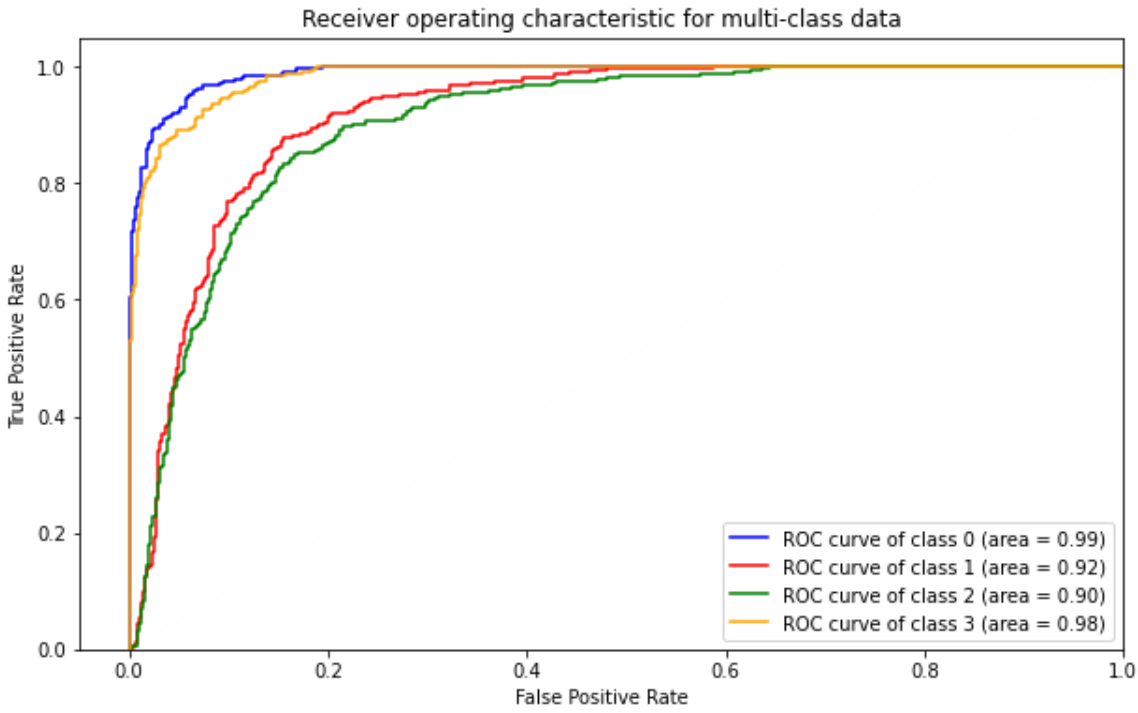
In [193]:

```
params = {}
#gridsearch searches for the best hyperparameters and keeps the classifier with the hig
hest recall score
skf = StratifiedKFold(n_splits=10)

nb = GridSearchCV(GaussianNB(), cv=skf, param_grid=params)

%time nb.fit(xtrain, ytrain)
gnb_2 = nb.fit(xtrain, ytrain)
y_pred_nb2 = gnb_2.predict(xtest)

print(accuracy_score(ytest, y_pred_nb2))
```

```
Wall time: 177 ms
0.8116666666666666
```

In [194]:

```
X1 = xtrain.to_numpy()
y1 = ytrain.to_numpy()

# Binarize the output
y_bin = label_binarize(y1, classes=[0, 1, 2, 3])
n_classes = y_bin.shape[1]

clf = gnb_2
y_score = cross_val_predict(clf, X1, y1, cv=10 ,method='predict_proba')

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_bin[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
colors = cycle(['blue', 'red', 'green','orange'])

plt.figure(figsize=(10,6))

for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color,
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(i, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--',linewidth=0.001)
plt.xlim([-0.05, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic for multi-class data')
plt.legend(loc="lower right")
plt.show()
```



CatBoost

In [196]:

```
CB_Classifier = CatBoostClassifier(iterations=100,verbose=True,
                        learning_rate=0.20,depth=2,l2_leaf_reg=100,bagging_temperature=
0.5)

%time CB_Classifier.fit(xtrain, ytrain,logging_level='Silent')
CB = CB_Classifier.fit(xtrain, ytrain,logging_level='Silent')
```

```
Wall time: 561 ms
```

In [197]:

```
pred_CB = CB.predict(xtest)
print(accuracy_score(ytest, pred_CB))
```

```
0.8716666666666667
```

In [198]:

```
X1 = xtrain.to_numpy()
y1 = ytrain.to_numpy()

# Binarize the output
y_bin = label_binarize(y1, classes=[0, 1, 2, 3])
n_classes = y_bin.shape[1]

y_score = cross_val_predict(CB, X1, y1, cv=10, method='predict_proba')

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_bin[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
colors = cycle(['blue', 'red', 'green','orange'])

plt.figure(figsize=(10,6))

for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color,
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(i, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--',linewidth=0.001)
plt.xlim([-0.05, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic for multi-class data')
plt.legend(loc="lower right")
plt.show()
```

```
0:      learn: 1.2753952        total: 3.5ms    remaining: 346ms
1:      learn: 1.2166303        total: 6.67ms   remaining: 327ms
2:      learn: 1.1376973        total: 10.3ms   remaining: 333ms
3:      learn: 1.0673105        total: 14ms     remaining: 337ms
4:      learn: 1.0077215        total: 17.8ms   remaining: 338ms
5:      learn: 0.9605573        total: 21.8ms   remaining: 342ms
6:      learn: 0.9279784        total: 25.1ms   remaining: 333ms
7:      learn: 0.8885651        total: 28.3ms   remaining: 326ms
8:      learn: 0.8548070        total: 31.5ms   remaining: 319ms
9:      learn: 0.8277770        total: 34.7ms   remaining: 313ms
10:     learn: 0.8013356        total: 37.6ms   remaining: 304ms
11:     learn: 0.7816332        total: 40.8ms   remaining: 299ms
12:     learn: 0.7628017        total: 44.5ms   remaining: 298ms
13:     learn: 0.7433223        total: 48.4ms   remaining: 298ms
14:     learn: 0.7299318        total: 51.6ms   remaining: 293ms
15:     learn: 0.7163519        total: 54.6ms   remaining: 287ms
16:     learn: 0.7027582        total: 57.8ms   remaining: 282ms
17:     learn: 0.6925283        total: 60.7ms   remaining: 277ms
18:     learn: 0.6812710        total: 64.1ms   remaining: 273ms
19:     learn: 0.6733298        total: 67ms     remaining: 268ms
20:     learn: 0.6622096        total: 71ms     remaining: 267ms
21:     learn: 0.6509349        total: 73.8ms   remaining: 261ms
22:     learn: 0.6424750        total: 77ms     remaining: 258ms
23:     learn: 0.6343650        total: 79.8ms   remaining: 253ms
24:     learn: 0.6253367        total: 82.6ms   remaining: 248ms
25:     learn: 0.6161257        total: 86.1ms   remaining: 245ms
26:     learn: 0.6085231        total: 89.1ms   remaining: 241ms
27:     learn: 0.6005571        total: 92.3ms   remaining: 237ms
28:     learn: 0.5925247        total: 95.1ms   remaining: 233ms
29:     learn: 0.5858227        total: 98.3ms   remaining: 229ms
30:     learn: 0.5787345        total: 102ms    remaining: 227ms
31:     learn: 0.5731937        total: 105ms    remaining: 224ms
32:     learn: 0.5686539        total: 109ms    remaining: 220ms
33:     learn: 0.5651651        total: 112ms    remaining: 217ms
34:     learn: 0.5613292        total: 115ms    remaining: 213ms
35:     learn: 0.5568486        total: 117ms    remaining: 209ms
36:     learn: 0.5535523        total: 121ms    remaining: 205ms
37:     learn: 0.5468657        total: 125ms    remaining: 203ms
38:     learn: 0.5428746        total: 127ms    remaining: 199ms
39:     learn: 0.5381051        total: 129ms    remaining: 194ms
40:     learn: 0.5335860        total: 132ms    remaining: 190ms
41:     learn: 0.5284744        total: 134ms    remaining: 186ms
42:     learn: 0.5221637        total: 137ms    remaining: 181ms
43:     learn: 0.5182496        total: 140ms    remaining: 178ms
44:     learn: 0.5132419        total: 142ms    remaining: 174ms
45:     learn: 0.5081537        total: 145ms    remaining: 170ms
46:     learn: 0.5041715        total: 148ms    remaining: 167ms
47:     learn: 0.5009582        total: 151ms    remaining: 163ms
48:     learn: 0.4978343        total: 154ms    remaining: 160ms
49:     learn: 0.4928388        total: 156ms    remaining: 156ms
50:     learn: 0.4897646        total: 158ms    remaining: 152ms
51:     learn: 0.4868937        total: 161ms    remaining: 149ms
52:     learn: 0.4831790        total: 163ms    remaining: 145ms
53:     learn: 0.4813801        total: 166ms    remaining: 141ms
54:     learn: 0.4787171        total: 169ms    remaining: 138ms
55:     learn: 0.4746514        total: 172ms    remaining: 135ms
56:     learn: 0.4715970        total: 174ms    remaining: 131ms
57:     learn: 0.4667925        total: 177ms    remaining: 128ms
58:     learn: 0.4650351        total: 180ms    remaining: 125ms
59:     learn: 0.4616604        total: 182ms    remaining: 122ms
60:     learn: 0.4601590        total: 185ms    remaining: 119ms
```

```
61:     learn: 0.4562544        total: 188ms    remaining: 115ms
62:     learn: 0.4541323        total: 191ms    remaining: 112ms
63:     learn: 0.4504773        total: 193ms    remaining: 109ms
64:     learn: 0.4488448        total: 196ms    remaining: 106ms
65:     learn: 0.4458034        total: 199ms    remaining: 102ms
66:     learn: 0.4438148        total: 201ms    remaining: 99ms
67:     learn: 0.4399860        total: 204ms    remaining: 96.2ms
68:     learn: 0.4366208        total: 207ms    remaining: 93ms
69:     learn: 0.4346316        total: 210ms    remaining: 89.9ms
70:     learn: 0.4334301        total: 213ms    remaining: 86.9ms
71:     learn: 0.4316439        total: 215ms    remaining: 83.7ms
72:     learn: 0.4301232        total: 218ms    remaining: 80.6ms
73:     learn: 0.4286740        total: 220ms    remaining: 77.3ms
74:     learn: 0.4252944        total: 222ms    remaining: 74.2ms
75:     learn: 0.4245711        total: 225ms    remaining: 71.1ms
76:     learn: 0.4216643        total: 228ms    remaining: 68.1ms
77:     learn: 0.4203228        total: 230ms    remaining: 65ms
78:     learn: 0.4173029        total: 233ms    remaining: 62ms
79:     learn: 0.4148108        total: 236ms    remaining: 59ms
80:     learn: 0.4125854        total: 238ms    remaining: 55.9ms
81:     learn: 0.4108614        total: 241ms    remaining: 52.8ms
82:     learn: 0.4084209        total: 243ms    remaining: 49.8ms
83:     learn: 0.4073726        total: 245ms    remaining: 46.7ms
84:     learn: 0.4043530        total: 248ms    remaining: 43.7ms
85:     learn: 0.4021855        total: 250ms    remaining: 40.7ms
86:     learn: 0.4008542        total: 252ms    remaining: 37.7ms
87:     learn: 0.4000093        total: 255ms    remaining: 34.8ms
88:     learn: 0.3985763        total: 257ms    remaining: 31.8ms
89:     learn: 0.3973174        total: 259ms    remaining: 28.8ms
90:     learn: 0.3956914        total: 262ms    remaining: 25.9ms
91:     learn: 0.3946200        total: 265ms    remaining: 23ms
92:     learn: 0.3922520        total: 267ms    remaining: 20.1ms
93:     learn: 0.3911094        total: 270ms    remaining: 17.2ms
94:     learn: 0.3882010        total: 272ms    remaining: 14.3ms
95:     learn: 0.3871040        total: 275ms    remaining: 11.4ms
96:     learn: 0.3867073        total: 277ms    remaining: 8.57ms
97:     learn: 0.3838553        total: 280ms    remaining: 5.71ms
98:     learn: 0.3828718        total: 282ms    remaining: 2.85ms
99:     learn: 0.3810231        total: 284ms    remaining: 0us
0:      learn: 1.2739973        total: 3.75ms   remaining: 371ms
1:      learn: 1.2166861        total: 6.43ms   remaining: 315ms
2:      learn: 1.1374947        total: 10.2ms   remaining: 331ms
3:      learn: 1.0665996        total: 13.7ms   remaining: 329ms
4:      learn: 1.0089090        total: 17ms     remaining: 323ms
5:      learn: 0.9612972        total: 21.3ms   remaining: 334ms
6:      learn: 0.9309506        total: 24.4ms   remaining: 324ms
7:      learn: 0.8917706        total: 26.8ms   remaining: 308ms
8:      learn: 0.8583636        total: 29.9ms   remaining: 302ms
9:      learn: 0.8393388        total: 33.1ms   remaining: 298ms
10:     learn: 0.8161026        total: 36.9ms   remaining: 298ms
11:     learn: 0.7920805        total: 40ms     remaining: 294ms
12:     learn: 0.7704134        total: 43.7ms   remaining: 293ms
13:     learn: 0.7535174        total: 49ms     remaining: 301ms
14:     learn: 0.7362099        total: 52.4ms   remaining: 297ms
15:     learn: 0.7200719        total: 55.9ms   remaining: 294ms
16:     learn: 0.7063741        total: 59.3ms   remaining: 289ms
17:     learn: 0.6952343        total: 62.7ms   remaining: 286ms
18:     learn: 0.6834839        total: 66.1ms   remaining: 282ms
19:     learn: 0.6759346        total: 70ms     remaining: 280ms
20:     learn: 0.6642578        total: 74.2ms   remaining: 279ms
21:     learn: 0.6532281        total: 77.8ms   remaining: 276ms
```

```
22:     learn: 0.6436942        total: 80.9ms   remaining: 271ms
23:     learn: 0.6345967        total: 84.5ms   remaining: 268ms
24:     learn: 0.6252095        total: 87.4ms   remaining: 262ms
25:     learn: 0.6164902        total: 90.6ms   remaining: 258ms
26:     learn: 0.6090140        total: 93.6ms   remaining: 253ms
27:     learn: 0.6007933        total: 96.6ms   remaining: 249ms
28:     learn: 0.5932263        total: 99.6ms   remaining: 244ms
29:     learn: 0.5884919        total: 102ms    remaining: 238ms
30:     learn: 0.5808640        total: 104ms    remaining: 231ms
31:     learn: 0.5740755        total: 106ms    remaining: 226ms
32:     learn: 0.5685084        total: 109ms    remaining: 221ms
33:     learn: 0.5620817        total: 111ms    remaining: 215ms
34:     learn: 0.5590633        total: 113ms    remaining: 210ms
35:     learn: 0.5535282        total: 116ms    remaining: 205ms
36:     learn: 0.5476816        total: 118ms    remaining: 201ms
37:     learn: 0.5416624        total: 121ms    remaining: 197ms
38:     learn: 0.5353077        total: 123ms    remaining: 193ms
39:     learn: 0.5295628        total: 127ms    remaining: 190ms
40:     learn: 0.5259010        total: 129ms    remaining: 185ms
41:     learn: 0.5220766        total: 131ms    remaining: 181ms
42:     learn: 0.5160596        total: 134ms    remaining: 178ms
43:     learn: 0.5117081        total: 137ms    remaining: 174ms
44:     learn: 0.5070295        total: 139ms    remaining: 170ms
45:     learn: 0.5047260        total: 141ms    remaining: 166ms
46:     learn: 0.5026262        total: 144ms    remaining: 162ms
47:     learn: 0.4978408        total: 147ms    remaining: 159ms
48:     learn: 0.4957924        total: 150ms    remaining: 156ms
49:     learn: 0.4925110        total: 153ms    remaining: 153ms
50:     learn: 0.4891969        total: 156ms    remaining: 149ms
51:     learn: 0.4864624        total: 158ms    remaining: 146ms
52:     learn: 0.4823360        total: 161ms    remaining: 143ms
53:     learn: 0.4803527        total: 164ms    remaining: 140ms
54:     learn: 0.4777230        total: 167ms    remaining: 136ms
55:     learn: 0.4738471        total: 171ms    remaining: 134ms
56:     learn: 0.4694232        total: 173ms    remaining: 131ms
57:     learn: 0.4678288        total: 176ms    remaining: 127ms
58:     learn: 0.4651685        total: 179ms    remaining: 125ms
59:     learn: 0.4614232        total: 182ms    remaining: 122ms
60:     learn: 0.4591628        total: 185ms    remaining: 118ms
61:     learn: 0.4571704        total: 187ms    remaining: 114ms
62:     learn: 0.4527313        total: 189ms    remaining: 111ms
63:     learn: 0.4490029        total: 191ms    remaining: 108ms
64:     learn: 0.4470999        total: 194ms    remaining: 105ms
65:     learn: 0.4451107        total: 197ms    remaining: 101ms
66:     learn: 0.4413746        total: 199ms    remaining: 98ms
67:     learn: 0.4395218        total: 202ms    remaining: 94.9ms
68:     learn: 0.4359896        total: 205ms    remaining: 92.1ms
69:     learn: 0.4344898        total: 207ms    remaining: 88.9ms
70:     learn: 0.4313988        total: 210ms    remaining: 85.8ms
71:     learn: 0.4277278        total: 212ms    remaining: 82.5ms
72:     learn: 0.4262451        total: 214ms    remaining: 79.2ms
73:     learn: 0.4246246        total: 216ms    remaining: 75.9ms
74:     learn: 0.4232239        total: 219ms    remaining: 73ms
75:     learn: 0.4205022        total: 221ms    remaining: 69.9ms
76:     learn: 0.4188282        total: 224ms    remaining: 66.9ms
77:     learn: 0.4172578        total: 226ms    remaining: 63.8ms
78:     learn: 0.4144607        total: 229ms    remaining: 60.8ms
79:     learn: 0.4119416        total: 231ms    remaining: 57.8ms
80:     learn: 0.4092955        total: 233ms    remaining: 54.6ms
81:     learn: 0.4080553        total: 235ms    remaining: 51.5ms
82:     learn: 0.4064814        total: 237ms    remaining: 48.4ms
```

```
83:     learn: 0.4046976        total: 239ms    remaining: 45.4ms
84:     learn: 0.4024522        total: 241ms    remaining: 42.5ms
85:     learn: 0.3999078        total: 243ms    remaining: 39.5ms
86:     learn: 0.3990353        total: 245ms    remaining: 36.6ms
87:     learn: 0.3981443        total: 247ms    remaining: 33.7ms
88:     learn: 0.3956926        total: 249ms    remaining: 30.8ms
89:     learn: 0.3943600        total: 251ms    remaining: 27.9ms
90:     learn: 0.3937633        total: 254ms    remaining: 25.1ms
91:     learn: 0.3929888        total: 259ms    remaining: 22.5ms
92:     learn: 0.3903388        total: 262ms    remaining: 19.7ms
93:     learn: 0.3879833        total: 264ms    remaining: 16.8ms
94:     learn: 0.3870272        total: 266ms    remaining: 14ms
95:     learn: 0.3858240        total: 268ms    remaining: 11.1ms
96:     learn: 0.3853358        total: 270ms    remaining: 8.36ms
97:     learn: 0.3829554        total: 273ms    remaining: 5.58ms
98:     learn: 0.3820335        total: 276ms    remaining: 2.78ms
99:     learn: 0.3810022        total: 280ms    remaining: 0us
0:      learn: 1.2734571        total: 3.3ms    remaining: 327ms
1:      learn: 1.1825547        total: 5.6ms    remaining: 274ms
2:      learn: 1.1074134        total: 9.24ms   remaining: 299ms
3:      learn: 1.0432421        total: 12.8ms   remaining: 307ms
4:      learn: 0.9883835        total: 15.4ms   remaining: 293ms
5:      learn: 0.9450527        total: 18.7ms   remaining: 292ms
6:      learn: 0.9114594        total: 21.6ms   remaining: 287ms
7:      learn: 0.8745153        total: 24.2ms   remaining: 279ms
8:      learn: 0.8454975        total: 26.7ms   remaining: 270ms
9:      learn: 0.8206254        total: 31.1ms   remaining: 280ms
10:     learn: 0.7957482        total: 33.6ms   remaining: 272ms
11:     learn: 0.7726662        total: 37.1ms   remaining: 272ms
12:     learn: 0.7548261        total: 39.8ms   remaining: 267ms
13:     learn: 0.7364119        total: 42.8ms   remaining: 263ms
14:     learn: 0.7229205        total: 46.6ms   remaining: 264ms
15:     learn: 0.7139466        total: 49.7ms   remaining: 261ms
16:     learn: 0.6993900        total: 52.6ms   remaining: 257ms
17:     learn: 0.6920901        total: 55.9ms   remaining: 255ms
18:     learn: 0.6802990        total: 57.8ms   remaining: 247ms
19:     learn: 0.6726027        total: 59.7ms   remaining: 239ms
20:     learn: 0.6604347        total: 61.7ms   remaining: 232ms
21:     learn: 0.6491811        total: 64.2ms   remaining: 227ms
22:     learn: 0.6399088        total: 66.5ms   remaining: 223ms
23:     learn: 0.6305521        total: 68.9ms   remaining: 218ms
24:     learn: 0.6204176        total: 71.6ms   remaining: 215ms
25:     learn: 0.6107730        total: 73.7ms   remaining: 210ms
26:     learn: 0.6031537        total: 75.8ms   remaining: 205ms
27:     learn: 0.5942962        total: 78.4ms   remaining: 202ms
28:     learn: 0.5865062        total: 80.6ms   remaining: 197ms
29:     learn: 0.5798560        total: 82.8ms   remaining: 193ms
30:     learn: 0.5724749        total: 85.4ms   remaining: 190ms
31:     learn: 0.5672802        total: 87.5ms   remaining: 186ms
32:     learn: 0.5623949        total: 89.8ms   remaining: 182ms
33:     learn: 0.5564354        total: 92.1ms   remaining: 179ms
34:     learn: 0.5515090        total: 95ms     remaining: 177ms
35:     learn: 0.5452512        total: 98.2ms   remaining: 175ms
36:     learn: 0.5393616        total: 100ms    remaining: 171ms
37:     learn: 0.5330306        total: 102ms    remaining: 167ms
38:     learn: 0.5267077        total: 105ms    remaining: 164ms
39:     learn: 0.5208031        total: 107ms    remaining: 161ms
40:     learn: 0.5165066        total: 110ms    remaining: 158ms
41:     learn: 0.5135979        total: 113ms    remaining: 156ms
42:     learn: 0.5108311        total: 115ms    remaining: 153ms
43:     learn: 0.5055797        total: 118ms    remaining: 150ms
```

```
44:     learn: 0.5004975        total: 122ms    remaining: 150ms
45:     learn: 0.4952886        total: 125ms    remaining: 147ms
46:     learn: 0.4909073        total: 127ms    remaining: 144ms
47:     learn: 0.4876343        total: 132ms    remaining: 143ms
48:     learn: 0.4845572        total: 135ms    remaining: 140ms
49:     learn: 0.4799178        total: 137ms    remaining: 137ms
50:     learn: 0.4777526        total: 139ms    remaining: 133ms
51:     learn: 0.4750375        total: 141ms    remaining: 130ms
52:     learn: 0.4709144        total: 143ms    remaining: 127ms
53:     learn: 0.4690478        total: 145ms    remaining: 123ms
54:     learn: 0.4661119        total: 148ms    remaining: 121ms
55:     learn: 0.4619159        total: 150ms    remaining: 118ms
56:     learn: 0.4591491        total: 152ms    remaining: 115ms
57:     learn: 0.4545879        total: 154ms    remaining: 111ms
58:     learn: 0.4524456        total: 156ms    remaining: 109ms
59:     learn: 0.4487666        total: 159ms    remaining: 106ms
60:     learn: 0.4445857        total: 161ms    remaining: 103ms
61:     learn: 0.4408825        total: 163ms    remaining: 100ms
62:     learn: 0.4387037        total: 165ms    remaining: 97.1ms
63:     learn: 0.4371461        total: 167ms    remaining: 94.2ms
64:     learn: 0.4349878        total: 170ms    remaining: 91.7ms
65:     learn: 0.4315334        total: 173ms    remaining: 89.4ms
66:     learn: 0.4301289        total: 175ms    remaining: 86.4ms
67:     learn: 0.4264857        total: 178ms    remaining: 83.9ms
68:     learn: 0.4231690        total: 181ms    remaining: 81.5ms
69:     learn: 0.4220118        total: 184ms    remaining: 79ms
70:     learn: 0.4188718        total: 187ms    remaining: 76.3ms
71:     learn: 0.4158915        total: 190ms    remaining: 73.7ms
72:     learn: 0.4143470        total: 193ms    remaining: 71.5ms
73:     learn: 0.4116053        total: 196ms    remaining: 69ms
74:     learn: 0.4083495        total: 200ms    remaining: 66.6ms
75:     learn: 0.4053676        total: 202ms    remaining: 63.9ms
76:     learn: 0.4043499        total: 204ms    remaining: 61ms
77:     learn: 0.4025539        total: 206ms    remaining: 58.1ms
78:     learn: 0.3994889        total: 208ms    remaining: 55.4ms
79:     learn: 0.3979664        total: 210ms    remaining: 52.5ms
80:     learn: 0.3963830        total: 212ms    remaining: 49.8ms
81:     learn: 0.3951742        total: 215ms    remaining: 47.2ms
82:     learn: 0.3926445        total: 217ms    remaining: 44.4ms
83:     learn: 0.3917056        total: 219ms    remaining: 41.7ms
84:     learn: 0.3905379        total: 221ms    remaining: 38.9ms
85:     learn: 0.3889954        total: 223ms    remaining: 36.3ms
86:     learn: 0.3883090        total: 225ms    remaining: 33.6ms
87:     learn: 0.3861134        total: 227ms    remaining: 31ms
88:     learn: 0.3835869        total: 229ms    remaining: 28.3ms
89:     learn: 0.3820523        total: 231ms    remaining: 25.7ms
90:     learn: 0.3806172        total: 233ms    remaining: 23.1ms
91:     learn: 0.3786529        total: 235ms    remaining: 20.4ms
92:     learn: 0.3770624        total: 239ms    remaining: 18ms
93:     learn: 0.3759142        total: 241ms    remaining: 15.4ms
94:     learn: 0.3744106        total: 243ms    remaining: 12.8ms
95:     learn: 0.3734959        total: 246ms    remaining: 10.2ms
96:     learn: 0.3714816        total: 248ms    remaining: 7.68ms
97:     learn: 0.3693255        total: 251ms    remaining: 5.13ms
98:     learn: 0.3685149        total: 254ms    remaining: 2.56ms
99:     learn: 0.3668714        total: 256ms    remaining: 0us
0:      learn: 1.2730341        total: 3.79ms   remaining: 376ms
1:      learn: 1.1821454        total: 6.1ms    remaining: 299ms
2:      learn: 1.1082176        total: 8.2ms    remaining: 265ms
3:      learn: 1.0437625        total: 10.1ms   remaining: 242ms
4:      learn: 0.9922814        total: 12.4ms   remaining: 235ms
```

```
5:      learn: 0.9476018        total: 14.4ms   remaining: 226ms
6:      learn: 0.9180266        total: 17.3ms   remaining: 230ms
7:      learn: 0.8793538        total: 20.9ms   remaining: 240ms
8:      learn: 0.8476452        total: 23ms     remaining: 233ms
9:      learn: 0.8240252        total: 25.9ms   remaining: 233ms
10:     learn: 0.8010644        total: 28.1ms   remaining: 227ms
11:     learn: 0.7783951        total: 29.8ms   remaining: 218ms
12:     learn: 0.7608733        total: 31.5ms   remaining: 211ms
13:     learn: 0.7424374        total: 33.6ms   remaining: 206ms
14:     learn: 0.7281112        total: 35.4ms   remaining: 200ms
15:     learn: 0.7184189        total: 37.7ms   remaining: 198ms
16:     learn: 0.7053162        total: 40.2ms   remaining: 196ms
17:     learn: 0.6955800        total: 42.1ms   remaining: 192ms
18:     learn: 0.6843147        total: 43.9ms   remaining: 187ms
19:     learn: 0.6771820        total: 45.8ms   remaining: 183ms
20:     learn: 0.6646665        total: 47.8ms   remaining: 180ms
21:     learn: 0.6523014        total: 49.9ms   remaining: 177ms
22:     learn: 0.6427771        total: 52.4ms   remaining: 176ms
23:     learn: 0.6346290        total: 54.9ms   remaining: 174ms
24:     learn: 0.6252994        total: 56.8ms   remaining: 170ms
25:     learn: 0.6206551        total: 58.8ms   remaining: 167ms
26:     learn: 0.6121720        total: 62.2ms   remaining: 168ms
27:     learn: 0.6025384        total: 66.1ms   remaining: 170ms
28:     learn: 0.5947552        total: 68.4ms   remaining: 167ms
29:     learn: 0.5888564        total: 71.1ms   remaining: 166ms
30:     learn: 0.5810771        total: 73.9ms   remaining: 165ms
31:     learn: 0.5741103        total: 76ms     remaining: 161ms
32:     learn: 0.5687727        total: 78.6ms   remaining: 160ms
33:     learn: 0.5653070        total: 81.5ms   remaining: 158ms
34:     learn: 0.5610922        total: 84.4ms   remaining: 157ms
35:     learn: 0.5558259        total: 86.8ms   remaining: 154ms
36:     learn: 0.5500204        total: 89.2ms   remaining: 152ms
37:     learn: 0.5435205        total: 91.3ms   remaining: 149ms
38:     learn: 0.5398741        total: 94.3ms   remaining: 148ms
39:     learn: 0.5338027        total: 96.8ms   remaining: 145ms
40:     learn: 0.5277522        total: 99.2ms   remaining: 143ms
41:     learn: 0.5249316        total: 102ms    remaining: 140ms
42:     learn: 0.5196455        total: 104ms    remaining: 138ms
43:     learn: 0.5135371        total: 107ms    remaining: 136ms
44:     learn: 0.5099235        total: 109ms    remaining: 134ms
45:     learn: 0.5053703        total: 111ms    remaining: 131ms
46:     learn: 0.5010724        total: 114ms    remaining: 128ms
47:     learn: 0.4981074        total: 116ms    remaining: 126ms
48:     learn: 0.4950427        total: 120ms    remaining: 125ms
49:     learn: 0.4903359        total: 122ms    remaining: 122ms
50:     learn: 0.4880213        total: 125ms    remaining: 120ms
51:     learn: 0.4860777        total: 129ms    remaining: 119ms
52:     learn: 0.4833026        total: 132ms    remaining: 117ms
53:     learn: 0.4813543        total: 135ms    remaining: 115ms
54:     learn: 0.4800892        total: 138ms    remaining: 113ms
55:     learn: 0.4756025        total: 140ms    remaining: 110ms
56:     learn: 0.4727760        total: 144ms    remaining: 109ms
57:     learn: 0.4682468        total: 147ms    remaining: 106ms
58:     learn: 0.4660574        total: 149ms    remaining: 103ms
59:     learn: 0.4622680        total: 151ms    remaining: 101ms
60:     learn: 0.4607101        total: 154ms    remaining: 98.2ms
61:     learn: 0.4583519        total: 156ms    remaining: 95.6ms
62:     learn: 0.4546906        total: 158ms    remaining: 92.8ms
63:     learn: 0.4499452        total: 160ms    remaining: 90ms
64:     learn: 0.4473416        total: 163ms    remaining: 88ms
65:     learn: 0.4441484        total: 166ms    remaining: 85.4ms
```

```
66:     learn: 0.4429304        total: 168ms    remaining: 83ms
67:     learn: 0.4414132        total: 171ms    remaining: 80.5ms
68:     learn: 0.4377387        total: 174ms    remaining: 78ms
69:     learn: 0.4335604        total: 176ms    remaining: 75.3ms
70:     learn: 0.4322205        total: 179ms    remaining: 73.3ms
71:     learn: 0.4285776        total: 182ms    remaining: 70.9ms
72:     learn: 0.4270540        total: 185ms    remaining: 68.3ms
73:     learn: 0.4243449        total: 187ms    remaining: 65.6ms
74:     learn: 0.4211308        total: 189ms    remaining: 62.9ms
75:     learn: 0.4182316        total: 191ms    remaining: 60.2ms
76:     learn: 0.4166012        total: 193ms    remaining: 57.5ms
77:     learn: 0.4151280        total: 195ms    remaining: 54.9ms
78:     learn: 0.4136783        total: 197ms    remaining: 52.4ms
79:     learn: 0.4112297        total: 199ms    remaining: 49.8ms
80:     learn: 0.4098120        total: 201ms    remaining: 47.2ms
81:     learn: 0.4088631        total: 203ms    remaining: 44.6ms
82:     learn: 0.4067142        total: 206ms    remaining: 42.1ms
83:     learn: 0.4055682        total: 208ms    remaining: 39.6ms
84:     learn: 0.4029132        total: 210ms    remaining: 37.1ms
85:     learn: 0.4003752        total: 212ms    remaining: 34.5ms
86:     learn: 0.3993056        total: 214ms    remaining: 31.9ms
87:     learn: 0.3986012        total: 216ms    remaining: 29.4ms
88:     learn: 0.3971960        total: 218ms    remaining: 27ms
89:     learn: 0.3961743        total: 220ms    remaining: 24.5ms
90:     learn: 0.3951629        total: 222ms    remaining: 22ms
91:     learn: 0.3927028        total: 224ms    remaining: 19.5ms
92:     learn: 0.3913428        total: 226ms    remaining: 17ms
93:     learn: 0.3903187        total: 229ms    remaining: 14.6ms
94:     learn: 0.3877488        total: 231ms    remaining: 12.2ms
95:     learn: 0.3869346        total: 233ms    remaining: 9.72ms
96:     learn: 0.3852312        total: 235ms    remaining: 7.27ms
97:     learn: 0.3842379        total: 237ms    remaining: 4.84ms
98:     learn: 0.3832361        total: 240ms    remaining: 2.42ms
99:     learn: 0.3822507        total: 242ms    remaining: 0us
0:      learn: 1.2748110        total: 3.63ms   remaining: 359ms
1:      learn: 1.2160595        total: 6.23ms   remaining: 305ms
2:      learn: 1.1356192        total: 8.46ms   remaining: 273ms
3:      learn: 1.0663006        total: 10.5ms   remaining: 251ms
4:      learn: 1.0091300        total: 12.7ms   remaining: 241ms
5:      learn: 0.9617630        total: 14.9ms   remaining: 234ms
6:      learn: 0.9285921        total: 16.7ms   remaining: 222ms
7:      learn: 0.8900032        total: 19.4ms   remaining: 223ms
8:      learn: 0.8608324        total: 22ms     remaining: 222ms
9:      learn: 0.8318919        total: 24.7ms   remaining: 222ms
10:     learn: 0.8077849        total: 26.5ms   remaining: 214ms
11:     learn: 0.7878018        total: 28.7ms   remaining: 210ms
12:     learn: 0.7654022        total: 30.4ms   remaining: 204ms
13:     learn: 0.7462582        total: 32.4ms   remaining: 199ms
14:     learn: 0.7326158        total: 34.6ms   remaining: 196ms
15:     learn: 0.7167502        total: 36.6ms   remaining: 192ms
16:     learn: 0.7029336        total: 38.5ms   remaining: 188ms
17:     learn: 0.6925769        total: 40.6ms   remaining: 185ms
18:     learn: 0.6812145        total: 43.4ms   remaining: 185ms
19:     learn: 0.6734490        total: 45.8ms   remaining: 183ms
20:     learn: 0.6615812        total: 48.3ms   remaining: 182ms
21:     learn: 0.6507804        total: 50.8ms   remaining: 180ms
22:     learn: 0.6417925        total: 52.9ms   remaining: 177ms
23:     learn: 0.6328479        total: 56.1ms   remaining: 178ms
24:     learn: 0.6227789        total: 58.2ms   remaining: 174ms
25:     learn: 0.6134400        total: 60ms     remaining: 171ms
26:     learn: 0.6059712        total: 61.8ms   remaining: 167ms
```

```
27:     learn: 0.5992634        total: 64.5ms   remaining: 166ms
28:     learn: 0.5896679        total: 66.5ms   remaining: 163ms
29:     learn: 0.5845290        total: 68.4ms   remaining: 160ms
30:     learn: 0.5772459        total: 70.3ms   remaining: 156ms
31:     learn: 0.5707688        total: 72.1ms   remaining: 153ms
32:     learn: 0.5659341        total: 74.3ms   remaining: 151ms
33:     learn: 0.5619819        total: 77.8ms   remaining: 151ms
34:     learn: 0.5587680        total: 79.9ms   remaining: 148ms
35:     learn: 0.5527351        total: 82ms     remaining: 146ms
36:     learn: 0.5465632        total: 83.8ms   remaining: 143ms
37:     learn: 0.5409628        total: 86.2ms   remaining: 141ms
38:     learn: 0.5368085        total: 88.3ms   remaining: 138ms
39:     learn: 0.5316192        total: 90.7ms   remaining: 136ms
40:     learn: 0.5276658        total: 92.4ms   remaining: 133ms
41:     learn: 0.5250389        total: 94.1ms   remaining: 130ms
42:     learn: 0.5220371        total: 95.8ms   remaining: 127ms
43:     learn: 0.5181684        total: 97.7ms   remaining: 124ms
44:     learn: 0.5125511        total: 100ms    remaining: 123ms
45:     learn: 0.5076853        total: 103ms    remaining: 120ms
46:     learn: 0.5027635        total: 104ms    remaining: 118ms
47:     learn: 0.4990960        total: 106ms    remaining: 115ms
48:     learn: 0.4932707        total: 109ms    remaining: 113ms
49:     learn: 0.4888675        total: 111ms    remaining: 111ms
50:     learn: 0.4854530        total: 113ms    remaining: 109ms
51:     learn: 0.4831525        total: 115ms    remaining: 106ms
52:     learn: 0.4797068        total: 117ms    remaining: 103ms
53:     learn: 0.4777064        total: 119ms    remaining: 101ms
54:     learn: 0.4750787        total: 122ms    remaining: 99.9ms
55:     learn: 0.4725369        total: 124ms    remaining: 97.5ms
56:     learn: 0.4687655        total: 126ms    remaining: 95.4ms
57:     learn: 0.4642072        total: 128ms    remaining: 92.9ms
58:     learn: 0.4618020        total: 131ms    remaining: 91.3ms
59:     learn: 0.4573424        total: 135ms    remaining: 90ms
60:     learn: 0.4549616        total: 137ms    remaining: 87.6ms
61:     learn: 0.4524542        total: 139ms    remaining: 85.4ms
62:     learn: 0.4489702        total: 143ms    remaining: 83.8ms
63:     learn: 0.4469760        total: 145ms    remaining: 81.8ms
64:     learn: 0.4451974        total: 148ms    remaining: 79.7ms
65:     learn: 0.4414780        total: 150ms    remaining: 77.3ms
66:     learn: 0.4395333        total: 152ms    remaining: 75.1ms
67:     learn: 0.4355089        total: 155ms    remaining: 73ms
68:     learn: 0.4319363        total: 158ms    remaining: 70.8ms
69:     learn: 0.4307783        total: 160ms    remaining: 68.6ms
70:     learn: 0.4269434        total: 162ms    remaining: 66.3ms
71:     learn: 0.4249432        total: 165ms    remaining: 64.4ms
72:     learn: 0.4234613        total: 169ms    remaining: 62.4ms
73:     learn: 0.4218555        total: 171ms    remaining: 60.1ms
74:     learn: 0.4203872        total: 173ms    remaining: 57.8ms
75:     learn: 0.4189709        total: 176ms    remaining: 55.6ms
76:     learn: 0.4169940        total: 178ms    remaining: 53.3ms
77:     learn: 0.4156836        total: 181ms    remaining: 51ms
78:     learn: 0.4141951        total: 183ms    remaining: 48.6ms
79:     learn: 0.4110638        total: 185ms    remaining: 46.4ms
80:     learn: 0.4095531        total: 188ms    remaining: 44.1ms
81:     learn: 0.4085897        total: 190ms    remaining: 41.6ms
82:     learn: 0.4056237        total: 192ms    remaining: 39.4ms
83:     learn: 0.4037407        total: 195ms    remaining: 37.1ms
84:     learn: 0.4011495        total: 197ms    remaining: 34.8ms
85:     learn: 0.3982042        total: 200ms    remaining: 32.5ms
86:     learn: 0.3974068        total: 202ms    remaining: 30.1ms
87:     learn: 0.3961164        total: 204ms    remaining: 27.8ms
```

```
88:     learn: 0.3933587        total: 205ms    remaining: 25.4ms
89:     learn: 0.3923593        total: 208ms    remaining: 23.1ms
90:     learn: 0.3893516        total: 210ms    remaining: 20.8ms
91:     learn: 0.3868588        total: 213ms    remaining: 18.5ms
92:     learn: 0.3857678        total: 215ms    remaining: 16.2ms
93:     learn: 0.3844606        total: 217ms    remaining: 13.8ms
94:     learn: 0.3833385        total: 219ms    remaining: 11.5ms
95:     learn: 0.3824442        total: 222ms    remaining: 9.24ms
96:     learn: 0.3813170        total: 224ms    remaining: 6.92ms
97:     learn: 0.3801799        total: 226ms    remaining: 4.61ms
98:     learn: 0.3786544        total: 228ms    remaining: 2.31ms
99:     learn: 0.3773045        total: 231ms    remaining: 0us
0:      learn: 1.2755959        total: 3.87ms   remaining: 384ms
1:      learn: 1.1908773        total: 7.03ms   remaining: 344ms
2:      learn: 1.1165634        total: 9.83ms   remaining: 318ms
3:      learn: 1.0497292        total: 12.4ms   remaining: 298ms
4:      learn: 0.9982429        total: 14.7ms   remaining: 279ms
5:      learn: 0.9536789        total: 18.3ms   remaining: 287ms
6:      learn: 0.9204804        total: 21ms     remaining: 279ms
7:      learn: 0.8810426        total: 23.3ms   remaining: 268ms
8:      learn: 0.8562023        total: 25.8ms   remaining: 261ms
9:      learn: 0.8319158        total: 28.5ms   remaining: 256ms
10:     learn: 0.8086451        total: 31.6ms   remaining: 256ms
11:     learn: 0.7848530        total: 33.8ms   remaining: 248ms
12:     learn: 0.7645784        total: 35.7ms   remaining: 239ms
13:     learn: 0.7466474        total: 38.3ms   remaining: 235ms
14:     learn: 0.7324432        total: 40.9ms   remaining: 232ms
15:     learn: 0.7171799        total: 43.8ms   remaining: 230ms
16:     learn: 0.7026465        total: 46.8ms   remaining: 229ms
17:     learn: 0.6908304        total: 49.2ms   remaining: 224ms
18:     learn: 0.6797263        total: 52.9ms   remaining: 225ms
19:     learn: 0.6679585        total: 55.6ms   remaining: 222ms
20:     learn: 0.6638622        total: 57.9ms   remaining: 218ms
21:     learn: 0.6513903        total: 60.7ms   remaining: 215ms
22:     learn: 0.6426002        total: 63.6ms   remaining: 213ms
23:     learn: 0.6343623        total: 67.3ms   remaining: 213ms
24:     learn: 0.6247777        total: 69.7ms   remaining: 209ms
25:     learn: 0.6173067        total: 71.7ms   remaining: 204ms
26:     learn: 0.6094746        total: 74.3ms   remaining: 201ms
27:     learn: 0.5996309        total: 77ms     remaining: 198ms
28:     learn: 0.5917006        total: 79.7ms   remaining: 195ms
29:     learn: 0.5874743        total: 82.1ms   remaining: 192ms
30:     learn: 0.5793928        total: 84.7ms   remaining: 189ms
31:     learn: 0.5725116        total: 88.1ms   remaining: 187ms
32:     learn: 0.5679205        total: 90.8ms   remaining: 184ms
33:     learn: 0.5640220        total: 92.9ms   remaining: 180ms
34:     learn: 0.5599144        total: 94.9ms   remaining: 176ms
35:     learn: 0.5533550        total: 97ms     remaining: 172ms
36:     learn: 0.5474696        total: 99ms     remaining: 169ms
37:     learn: 0.5414353        total: 101ms    remaining: 165ms
38:     learn: 0.5367403        total: 103ms    remaining: 161ms
39:     learn: 0.5306201        total: 105ms    remaining: 158ms
40:     learn: 0.5271537        total: 107ms    remaining: 154ms
41:     learn: 0.5232013        total: 110ms    remaining: 152ms
42:     learn: 0.5171093        total: 112ms    remaining: 148ms
43:     learn: 0.5131573        total: 114ms    remaining: 145ms
44:     learn: 0.5095080        total: 116ms    remaining: 142ms
45:     learn: 0.5041812        total: 118ms    remaining: 139ms
46:     learn: 0.4992955        total: 121ms    remaining: 136ms
47:     learn: 0.4963386        total: 123ms    remaining: 133ms
48:     learn: 0.4916046        total: 125ms    remaining: 130ms
```

```
49:     learn: 0.4886286        total: 127ms    remaining: 127ms
50:     learn: 0.4829413        total: 130ms    remaining: 125ms
51:     learn: 0.4810797        total: 134ms    remaining: 123ms
52:     learn: 0.4786941        total: 136ms    remaining: 121ms
53:     learn: 0.4769168        total: 138ms    remaining: 118ms
54:     learn: 0.4741638        total: 141ms    remaining: 115ms
55:     learn: 0.4698373        total: 144ms    remaining: 113ms
56:     learn: 0.4661090        total: 146ms    remaining: 110ms
57:     learn: 0.4647188        total: 148ms    remaining: 107ms
58:     learn: 0.4623602        total: 150ms    remaining: 104ms
59:     learn: 0.4581716        total: 152ms    remaining: 101ms
60:     learn: 0.4538548        total: 154ms    remaining: 98.7ms
61:     learn: 0.4508766        total: 156ms    remaining: 95.7ms
62:     learn: 0.4462092        total: 158ms    remaining: 92.7ms
63:     learn: 0.4441668        total: 160ms    remaining: 90.3ms
64:     learn: 0.4424473        total: 163ms    remaining: 87.6ms
65:     learn: 0.4397659        total: 165ms    remaining: 85.1ms
66:     learn: 0.4387019        total: 168ms    remaining: 82.6ms
67:     learn: 0.4346189        total: 170ms    remaining: 79.9ms
68:     learn: 0.4325692        total: 172ms    remaining: 77.1ms
69:     learn: 0.4313230        total: 174ms    remaining: 74.7ms
70:     learn: 0.4279555        total: 176ms    remaining: 72ms
71:     learn: 0.4251062        total: 178ms    remaining: 69.3ms
72:     learn: 0.4224219        total: 180ms    remaining: 66.7ms
73:     learn: 0.4190541        total: 182ms    remaining: 64ms
74:     learn: 0.4174267        total: 184ms    remaining: 61.2ms
75:     learn: 0.4164491        total: 188ms    remaining: 59.2ms
76:     learn: 0.4151799        total: 189ms    remaining: 56.6ms
77:     learn: 0.4137364        total: 191ms    remaining: 54ms
78:     learn: 0.4105232        total: 193ms    remaining: 51.4ms
79:     learn: 0.4082452        total: 195ms    remaining: 48.8ms
80:     learn: 0.4067397        total: 198ms    remaining: 46.4ms
81:     learn: 0.4049240        total: 200ms    remaining: 43.9ms
82:     learn: 0.4029647        total: 202ms    remaining: 41.3ms
83:     learn: 0.4013745        total: 204ms    remaining: 38.8ms
84:     learn: 0.3986819        total: 206ms    remaining: 36.3ms
85:     learn: 0.3962460        total: 208ms    remaining: 33.8ms
86:     learn: 0.3950892        total: 209ms    remaining: 31.3ms
87:     learn: 0.3939476        total: 211ms    remaining: 28.8ms
88:     learn: 0.3917176        total: 213ms    remaining: 26.3ms
89:     learn: 0.3909361        total: 215ms    remaining: 23.9ms
90:     learn: 0.3893448        total: 217ms    remaining: 21.4ms
91:     learn: 0.3873474        total: 218ms    remaining: 19ms
92:     learn: 0.3858557        total: 220ms    remaining: 16.6ms
93:     learn: 0.3845773        total: 222ms    remaining: 14.2ms
94:     learn: 0.3835120        total: 225ms    remaining: 11.8ms
95:     learn: 0.3825805        total: 227ms    remaining: 9.46ms
96:     learn: 0.3811228        total: 229ms    remaining: 7.08ms
97:     learn: 0.3791872        total: 231ms    remaining: 4.71ms
98:     learn: 0.3777426        total: 233ms    remaining: 2.35ms
99:     learn: 0.3767658        total: 235ms    remaining: 0us
0:      learn: 1.2747998        total: 3.39ms   remaining: 335ms
1:      learn: 1.2172413        total: 6.93ms   remaining: 340ms
2:      learn: 1.1381695        total: 10.3ms   remaining: 334ms
3:      learn: 1.0677310        total: 13.2ms   remaining: 317ms
4:      learn: 1.0102880        total: 16.3ms   remaining: 311ms
5:      learn: 0.9617603        total: 19.7ms   remaining: 309ms
6:      learn: 0.9300508        total: 23.4ms   remaining: 311ms
7:      learn: 0.8915524        total: 26.1ms   remaining: 300ms
8:      learn: 0.8622218        total: 28.6ms   remaining: 289ms
9:      learn: 0.8335029        total: 31.4ms   remaining: 282ms
```

```
10:     learn: 0.8148632        total: 34ms     remaining: 275ms
11:     learn: 0.7898943        total: 37.3ms   remaining: 273ms
12:     learn: 0.7692301        total: 40ms     remaining: 268ms
13:     learn: 0.7522622        total: 42.8ms   remaining: 263ms
14:     learn: 0.7358949        total: 45.7ms   remaining: 259ms
15:     learn: 0.7204714        total: 48.8ms   remaining: 256ms
16:     learn: 0.7057014        total: 51.8ms   remaining: 253ms
17:     learn: 0.6952028        total: 54.2ms   remaining: 247ms
18:     learn: 0.6839958        total: 56.9ms   remaining: 242ms
19:     learn: 0.6728585        total: 60.3ms   remaining: 241ms
20:     learn: 0.6614812        total: 62.9ms   remaining: 237ms
21:     learn: 0.6513471        total: 65.5ms   remaining: 232ms
22:     learn: 0.6423604        total: 69.1ms   remaining: 231ms
23:     learn: 0.6338626        total: 72.1ms   remaining: 228ms
24:     learn: 0.6243426        total: 75.2ms   remaining: 226ms
25:     learn: 0.6195071        total: 77.8ms   remaining: 221ms
26:     learn: 0.6119768        total: 80.4ms   remaining: 217ms
27:     learn: 0.6044604        total: 83.5ms   remaining: 215ms
28:     learn: 0.5965408        total: 86.2ms   remaining: 211ms
29:     learn: 0.5902864        total: 88.8ms   remaining: 207ms
30:     learn: 0.5825791        total: 91.7ms   remaining: 204ms
31:     learn: 0.5759140        total: 94.4ms   remaining: 201ms
32:     learn: 0.5706780        total: 97.1ms   remaining: 197ms
33:     learn: 0.5666452        total: 99ms     remaining: 192ms
34:     learn: 0.5600935        total: 101ms    remaining: 188ms
35:     learn: 0.5542554        total: 103ms    remaining: 184ms
36:     learn: 0.5483048        total: 106ms    remaining: 180ms
37:     learn: 0.5434714        total: 108ms    remaining: 177ms
38:     learn: 0.5387406        total: 112ms    remaining: 174ms
39:     learn: 0.5340992        total: 114ms    remaining: 170ms
40:     learn: 0.5306830        total: 116ms    remaining: 167ms
41:     learn: 0.5263043        total: 118ms    remaining: 163ms
42:     learn: 0.5214161        total: 120ms    remaining: 160ms
43:     learn: 0.5163988        total: 123ms    remaining: 157ms
44:     learn: 0.5127763        total: 125ms    remaining: 153ms
45:     learn: 0.5075948        total: 128ms    remaining: 150ms
46:     learn: 0.5035384        total: 131ms    remaining: 148ms
47:     learn: 0.5011518        total: 134ms    remaining: 145ms
48:     learn: 0.4956189        total: 137ms    remaining: 143ms
49:     learn: 0.4911091        total: 139ms    remaining: 139ms
50:     learn: 0.4882946        total: 142ms    remaining: 136ms
51:     learn: 0.4858175        total: 144ms    remaining: 133ms
52:     learn: 0.4837512        total: 147ms    remaining: 130ms
53:     learn: 0.4817344        total: 149ms    remaining: 127ms
54:     learn: 0.4798949        total: 151ms    remaining: 124ms
55:     learn: 0.4754026        total: 154ms    remaining: 121ms
56:     learn: 0.4705384        total: 156ms    remaining: 118ms
57:     learn: 0.4692427        total: 158ms    remaining: 115ms
58:     learn: 0.4666872        total: 160ms    remaining: 111ms
59:     learn: 0.4624697        total: 162ms    remaining: 108ms
60:     learn: 0.4600988        total: 164ms    remaining: 105ms
61:     learn: 0.4585958        total: 167ms    remaining: 102ms
62:     learn: 0.4543766        total: 168ms    remaining: 98.9ms
63:     learn: 0.4524496        total: 171ms    remaining: 96.2ms
64:     learn: 0.4509457        total: 173ms    remaining: 93.3ms
65:     learn: 0.4469401        total: 176ms    remaining: 90.4ms
66:     learn: 0.4451373        total: 178ms    remaining: 87.4ms
67:     learn: 0.4412409        total: 179ms    remaining: 84.4ms
68:     learn: 0.4383815        total: 181ms    remaining: 81.4ms
69:     learn: 0.4348269        total: 183ms    remaining: 78.6ms
70:     learn: 0.4313773        total: 186ms    remaining: 75.8ms
```

```
71:     learn: 0.4286500        total: 187ms    remaining: 72.8ms
72:     learn: 0.4257717        total: 189ms    remaining: 69.9ms
73:     learn: 0.4242274        total: 191ms    remaining: 67ms
74:     learn: 0.4231216        total: 192ms    remaining: 64.1ms
75:     learn: 0.4216411        total: 194ms    remaining: 61.3ms
76:     learn: 0.4200788        total: 196ms    remaining: 58.6ms
77:     learn: 0.4179978        total: 198ms    remaining: 55.9ms
78:     learn: 0.4164353        total: 200ms    remaining: 53.1ms
79:     learn: 0.4142536        total: 202ms    remaining: 50.5ms
80:     learn: 0.4114078        total: 204ms    remaining: 47.8ms
81:     learn: 0.4101479        total: 206ms    remaining: 45.2ms
82:     learn: 0.4080861        total: 208ms    remaining: 42.7ms
83:     learn: 0.4067932        total: 210ms    remaining: 40ms
84:     learn: 0.4043287        total: 212ms    remaining: 37.3ms
85:     learn: 0.4016966        total: 213ms    remaining: 34.7ms
86:     learn: 0.4002564        total: 215ms    remaining: 32.2ms
87:     learn: 0.3978463        total: 218ms    remaining: 29.7ms
88:     learn: 0.3964029        total: 220ms    remaining: 27.1ms
89:     learn: 0.3954603        total: 221ms    remaining: 24.6ms
90:     learn: 0.3938256        total: 223ms    remaining: 22.1ms
91:     learn: 0.3923623        total: 227ms    remaining: 19.7ms
92:     learn: 0.3912258        total: 228ms    remaining: 17.2ms
93:     learn: 0.3900926        total: 230ms    remaining: 14.7ms
94:     learn: 0.3875896        total: 232ms    remaining: 12.2ms
95:     learn: 0.3852526        total: 233ms    remaining: 9.72ms
96:     learn: 0.3839876        total: 235ms    remaining: 7.26ms
97:     learn: 0.3814387        total: 237ms    remaining: 4.83ms
98:     learn: 0.3803839        total: 239ms    remaining: 2.41ms
99:     learn: 0.3787239        total: 240ms    remaining: 0us
0:      learn: 1.2735701        total: 4.09ms   remaining: 405ms
1:      learn: 1.2153876        total: 8.19ms   remaining: 401ms
2:      learn: 1.1356124        total: 10.8ms   remaining: 350ms
3:      learn: 1.0641008        total: 13.7ms   remaining: 328ms
4:      learn: 1.0056898        total: 16.3ms   remaining: 310ms
5:      learn: 0.9590838        total: 18.9ms   remaining: 297ms
6:      learn: 0.9263456        total: 22.9ms   remaining: 304ms
7:      learn: 0.8879549        total: 25.6ms   remaining: 295ms
8:      learn: 0.8542090        total: 28.7ms   remaining: 290ms
9:      learn: 0.8310565        total: 31.5ms   remaining: 283ms
10:     learn: 0.8082540        total: 34ms     remaining: 275ms
11:     learn: 0.7879774        total: 36.6ms   remaining: 269ms
12:     learn: 0.7692434        total: 40ms     remaining: 268ms
13:     learn: 0.7501197        total: 43.1ms   remaining: 265ms
14:     learn: 0.7314532        total: 46.3ms   remaining: 263ms
15:     learn: 0.7167868        total: 49ms     remaining: 257ms
16:     learn: 0.7035641        total: 52.1ms   remaining: 254ms
17:     learn: 0.6941544        total: 55.5ms   remaining: 253ms
18:     learn: 0.6828769        total: 58.5ms   remaining: 249ms
19:     learn: 0.6717598        total: 61.5ms   remaining: 246ms
20:     learn: 0.6610451        total: 64.2ms   remaining: 241ms
21:     learn: 0.6513309        total: 66.9ms   remaining: 237ms
22:     learn: 0.6420732        total: 69.6ms   remaining: 233ms
23:     learn: 0.6329806        total: 72.3ms   remaining: 229ms
24:     learn: 0.6234502        total: 75.7ms   remaining: 227ms
25:     learn: 0.6189404        total: 78.4ms   remaining: 223ms
26:     learn: 0.6115004        total: 80.9ms   remaining: 219ms
27:     learn: 0.6033167        total: 83.8ms   remaining: 215ms
28:     learn: 0.5943136        total: 86.4ms   remaining: 212ms
29:     learn: 0.5876793        total: 89ms     remaining: 208ms
30:     learn: 0.5800801        total: 91.7ms   remaining: 204ms
31:     learn: 0.5735889        total: 94.1ms   remaining: 200ms
```

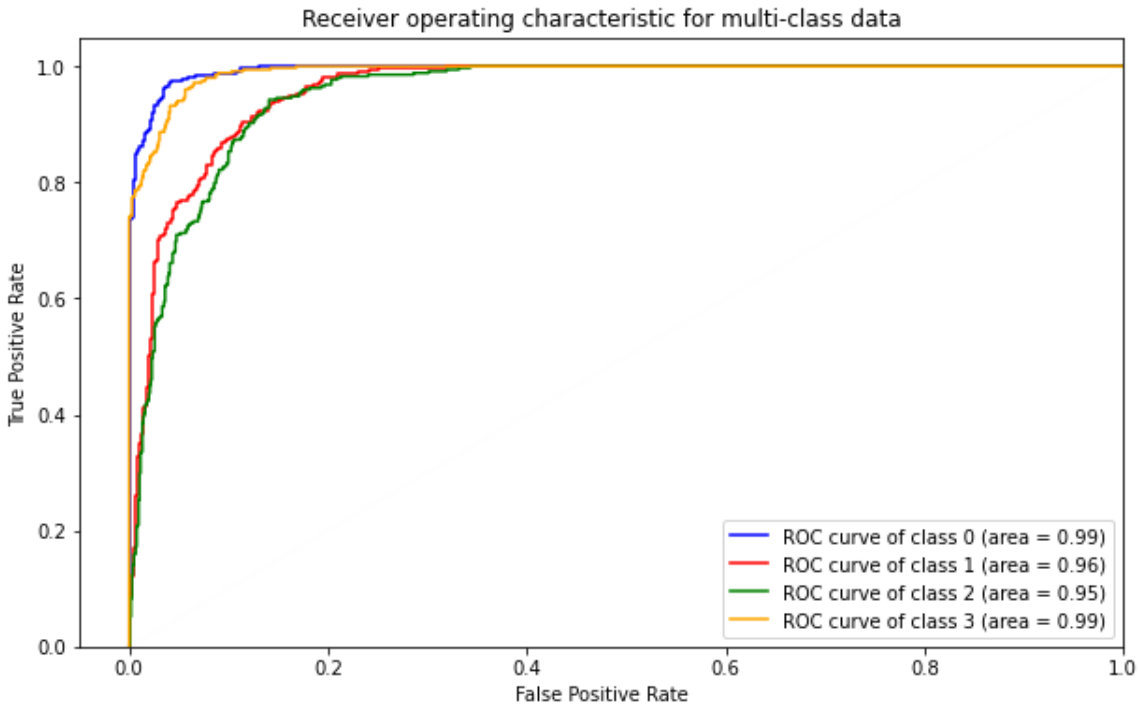
```
32:     learn: 0.5688617        total: 96.7ms   remaining: 196ms
33:     learn: 0.5654787        total: 99.4ms   remaining: 193ms
34:     learn: 0.5582101        total: 102ms    remaining: 190ms
35:     learn: 0.5517983        total: 105ms    remaining: 187ms
36:     learn: 0.5462140        total: 108ms    remaining: 183ms
37:     learn: 0.5407971        total: 110ms    remaining: 180ms
38:     learn: 0.5364395        total: 113ms    remaining: 177ms
39:     learn: 0.5320764        total: 116ms    remaining: 174ms
40:     learn: 0.5280794        total: 119ms    remaining: 171ms
41:     learn: 0.5254834        total: 122ms    remaining: 168ms
42:     learn: 0.5194652        total: 125ms    remaining: 165ms
43:     learn: 0.5154503        total: 128ms    remaining: 162ms
44:     learn: 0.5115025        total: 130ms    remaining: 159ms
45:     learn: 0.5069749        total: 133ms    remaining: 156ms
46:     learn: 0.5022412        total: 136ms    remaining: 153ms
47:     learn: 0.4994969        total: 138ms    remaining: 150ms
48:     learn: 0.4960511        total: 141ms    remaining: 147ms
49:     learn: 0.4913701        total: 144ms    remaining: 144ms
50:     learn: 0.4871163        total: 146ms    remaining: 141ms
51:     learn: 0.4843727        total: 149ms    remaining: 138ms
52:     learn: 0.4794486        total: 153ms    remaining: 136ms
53:     learn: 0.4768706        total: 156ms    remaining: 133ms
54:     learn: 0.4742422        total: 159ms    remaining: 130ms
55:     learn: 0.4699489        total: 162ms    remaining: 127ms
56:     learn: 0.4663356        total: 165ms    remaining: 124ms
57:     learn: 0.4644426        total: 167ms    remaining: 121ms
58:     learn: 0.4618373        total: 170ms    remaining: 118ms
59:     learn: 0.4581741        total: 173ms    remaining: 115ms
60:     learn: 0.4567053        total: 176ms    remaining: 112ms
61:     learn: 0.4545093        total: 179ms    remaining: 110ms
62:     learn: 0.4522191        total: 182ms    remaining: 107ms
63:     learn: 0.4508318        total: 185ms    remaining: 104ms
64:     learn: 0.4493061        total: 188ms    remaining: 101ms
65:     learn: 0.4458368        total: 191ms    remaining: 98.3ms
66:     learn: 0.4448092        total: 194ms    remaining: 95.3ms
67:     learn: 0.4431584        total: 197ms    remaining: 92.5ms
68:     learn: 0.4399117        total: 199ms    remaining: 89.6ms
69:     learn: 0.4363538        total: 203ms    remaining: 86.8ms
70:     learn: 0.4329708        total: 207ms    remaining: 84.5ms
71:     learn: 0.4294181        total: 210ms    remaining: 81.5ms
72:     learn: 0.4261615        total: 213ms    remaining: 78.7ms
73:     learn: 0.4231900        total: 216ms    remaining: 75.9ms
74:     learn: 0.4219890        total: 219ms    remaining: 73.1ms
75:     learn: 0.4200449        total: 222ms    remaining: 70.1ms
76:     learn: 0.4183613        total: 224ms    remaining: 67ms
77:     learn: 0.4178982        total: 227ms    remaining: 64.1ms
78:     learn: 0.4150389        total: 230ms    remaining: 61ms
79:     learn: 0.4127137        total: 233ms    remaining: 58.4ms
80:     learn: 0.4098998        total: 236ms    remaining: 55.4ms
81:     learn: 0.4083726        total: 239ms    remaining: 52.5ms
82:     learn: 0.4070757        total: 242ms    remaining: 49.5ms
83:     learn: 0.4054920        total: 244ms    remaining: 46.5ms
84:     learn: 0.4031838        total: 246ms    remaining: 43.4ms
85:     learn: 0.4018090        total: 248ms    remaining: 40.5ms
86:     learn: 0.4006799        total: 252ms    remaining: 37.6ms
87:     learn: 0.3992127        total: 254ms    remaining: 34.6ms
88:     learn: 0.3968596        total: 256ms    remaining: 31.7ms
89:     learn: 0.3959583        total: 259ms    remaining: 28.7ms
90:     learn: 0.3930842        total: 261ms    remaining: 25.8ms
91:     learn: 0.3909100        total: 263ms    remaining: 22.9ms
92:     learn: 0.3898211        total: 266ms    remaining: 20ms
```

```
93:     learn: 0.3884551        total: 268ms    remaining: 17.1ms
94:     learn: 0.3861331        total: 270ms    remaining: 14.2ms
95:     learn: 0.3836530        total: 273ms    remaining: 11.4ms
96:     learn: 0.3826122        total: 275ms    remaining: 8.52ms
97:     learn: 0.3799528        total: 278ms    remaining: 5.67ms
98:     learn: 0.3778438        total: 281ms    remaining: 2.83ms
99:     learn: 0.3757384        total: 283ms    remaining: 0us
0:      learn: 1.2723152        total: 5.29ms   remaining: 524ms
1:      learn: 1.1863494        total: 8.68ms   remaining: 425ms
2:      learn: 1.1118378        total: 12.1ms   remaining: 392ms
3:      learn: 1.0462871        total: 15.6ms   remaining: 374ms
4:      learn: 0.9918557        total: 19.5ms   remaining: 370ms
5:      learn: 0.9460111        total: 22.2ms   remaining: 348ms
6:      learn: 0.9140214        total: 25.3ms   remaining: 336ms
7:      learn: 0.8777098        total: 28.6ms   remaining: 328ms
8:      learn: 0.8495920        total: 31.5ms   remaining: 319ms
9:      learn: 0.8210475        total: 34.8ms   remaining: 313ms
10:     learn: 0.8027171        total: 38.4ms   remaining: 311ms
11:     learn: 0.7790725        total: 41.2ms   remaining: 302ms
12:     learn: 0.7589382        total: 44.1ms   remaining: 295ms
13:     learn: 0.7407975        total: 46.9ms   remaining: 288ms
14:     learn: 0.7275980        total: 50.5ms   remaining: 286ms
15:     learn: 0.7120808        total: 53.2ms   remaining: 279ms
16:     learn: 0.6982566        total: 55.9ms   remaining: 273ms
17:     learn: 0.6879209        total: 58.7ms   remaining: 267ms
18:     learn: 0.6771587        total: 61.5ms   remaining: 262ms
19:     learn: 0.6663201        total: 64.7ms   remaining: 259ms
20:     learn: 0.6546062        total: 67.4ms   remaining: 254ms
21:     learn: 0.6442036        total: 70.1ms   remaining: 248ms
22:     learn: 0.6356420        total: 73.2ms   remaining: 245ms
23:     learn: 0.6279372        total: 76.5ms   remaining: 242ms
24:     learn: 0.6190433        total: 79.1ms   remaining: 237ms
25:     learn: 0.6108978        total: 81.7ms   remaining: 233ms
26:     learn: 0.6017141        total: 84.3ms   remaining: 228ms
27:     learn: 0.5955901        total: 87.9ms   remaining: 226ms
28:     learn: 0.5880651        total: 90.6ms   remaining: 222ms
29:     learn: 0.5825260        total: 93.2ms   remaining: 217ms
30:     learn: 0.5752025        total: 95.7ms   remaining: 213ms
31:     learn: 0.5687629        total: 98.2ms   remaining: 209ms
32:     learn: 0.5625164        total: 101ms    remaining: 205ms
33:     learn: 0.5592776        total: 104ms    remaining: 201ms
34:     learn: 0.5525014        total: 106ms    remaining: 197ms
35:     learn: 0.5455043        total: 109ms    remaining: 193ms
36:     learn: 0.5403243        total: 111ms    remaining: 189ms
37:     learn: 0.5359331        total: 114ms    remaining: 186ms
38:     learn: 0.5324086        total: 116ms    remaining: 182ms
39:     learn: 0.5271389        total: 119ms    remaining: 178ms
40:     learn: 0.5240054        total: 121ms    remaining: 175ms
41:     learn: 0.5193238        total: 124ms    remaining: 172ms
42:     learn: 0.5136314        total: 127ms    remaining: 169ms
43:     learn: 0.5085698        total: 130ms    remaining: 165ms
44:     learn: 0.5050685        total: 133ms    remaining: 162ms
45:     learn: 0.5003555        total: 135ms    remaining: 159ms
46:     learn: 0.4959733        total: 138ms    remaining: 156ms
47:     learn: 0.4917321        total: 141ms    remaining: 152ms
48:     learn: 0.4885225        total: 143ms    remaining: 149ms
49:     learn: 0.4843745        total: 146ms    remaining: 146ms
50:     learn: 0.4813822        total: 149ms    remaining: 144ms
51:     learn: 0.4791163        total: 152ms    remaining: 140ms
52:     learn: 0.4770977        total: 155ms    remaining: 138ms
53:     learn: 0.4731452        total: 158ms    remaining: 135ms
```

```
54:     learn: 0.4712638        total: 161ms    remaining: 132ms
55:     learn: 0.4678018        total: 164ms    remaining: 128ms
56:     learn: 0.4643050        total: 166ms    remaining: 125ms
57:     learn: 0.4624788        total: 169ms    remaining: 122ms
58:     learn: 0.4600881        total: 172ms    remaining: 119ms
59:     learn: 0.4582824        total: 175ms    remaining: 116ms
60:     learn: 0.4569280        total: 178ms    remaining: 114ms
61:     learn: 0.4555909        total: 181ms    remaining: 111ms
62:     learn: 0.4521197        total: 183ms    remaining: 108ms
63:     learn: 0.4499343        total: 186ms    remaining: 104ms
64:     learn: 0.4485921        total: 189ms    remaining: 102ms
65:     learn: 0.4456566        total: 192ms    remaining: 98.7ms
66:     learn: 0.4446278        total: 194ms    remaining: 95.7ms
67:     learn: 0.4402586        total: 197ms    remaining: 92.6ms
68:     learn: 0.4383439        total: 200ms    remaining: 89.8ms
69:     learn: 0.4348304        total: 203ms    remaining: 86.9ms
70:     learn: 0.4310599        total: 206ms    remaining: 84.1ms
71:     learn: 0.4278742        total: 209ms    remaining: 81.3ms
72:     learn: 0.4249936        total: 212ms    remaining: 78.4ms
73:     learn: 0.4223415        total: 215ms    remaining: 75.7ms
74:     learn: 0.4213885        total: 218ms    remaining: 72.7ms
75:     learn: 0.4199991        total: 221ms    remaining: 69.7ms
76:     learn: 0.4172399        total: 224ms    remaining: 66.8ms
77:     learn: 0.4162112        total: 226ms    remaining: 63.8ms
78:     learn: 0.4136123        total: 229ms    remaining: 60.9ms
79:     learn: 0.4119034        total: 232ms    remaining: 58ms
80:     learn: 0.4093375        total: 235ms    remaining: 55.2ms
81:     learn: 0.4078059        total: 239ms    remaining: 52.4ms
82:     learn: 0.4059058        total: 242ms    remaining: 49.6ms
83:     learn: 0.4047079        total: 246ms    remaining: 46.9ms
84:     learn: 0.4023359        total: 249ms    remaining: 44ms
85:     learn: 0.3997476        total: 252ms    remaining: 41.1ms
86:     learn: 0.3983462        total: 255ms    remaining: 38.1ms
87:     learn: 0.3970695        total: 258ms    remaining: 35.2ms
88:     learn: 0.3955394        total: 261ms    remaining: 32.3ms
89:     learn: 0.3944622        total: 264ms    remaining: 29.3ms
90:     learn: 0.3930360        total: 266ms    remaining: 26.3ms
91:     learn: 0.3908554        total: 269ms    remaining: 23.4ms
92:     learn: 0.3896321        total: 271ms    remaining: 20.4ms
93:     learn: 0.3882895        total: 274ms    remaining: 17.5ms
94:     learn: 0.3860400        total: 276ms    remaining: 14.5ms
95:     learn: 0.3847615        total: 278ms    remaining: 11.6ms
96:     learn: 0.3839134        total: 281ms    remaining: 8.68ms
97:     learn: 0.3829297        total: 283ms    remaining: 5.78ms
98:     learn: 0.3802806        total: 286ms    remaining: 2.89ms
99:     learn: 0.3783345        total: 290ms    remaining: 0us
0:      learn: 1.2744446        total: 3.34ms   remaining: 331ms
1:      learn: 1.2160922        total: 6.42ms   remaining: 315ms
2:      learn: 1.1372016        total: 9.43ms   remaining: 305ms
3:      learn: 1.0666464        total: 12.7ms   remaining: 306ms
4:      learn: 1.0092365        total: 15.6ms   remaining: 296ms
5:      learn: 0.9636505        total: 18.3ms   remaining: 286ms
6:      learn: 0.9295366        total: 22.4ms   remaining: 298ms
7:      learn: 0.8902721        total: 25.3ms   remaining: 291ms
8:      learn: 0.8584915        total: 28.3ms   remaining: 286ms
9:      learn: 0.8323570        total: 31.3ms   remaining: 281ms
10:     learn: 0.8118794        total: 34.3ms   remaining: 277ms
11:     learn: 0.7872915        total: 37.1ms   remaining: 272ms
12:     learn: 0.7687710        total: 40.2ms   remaining: 269ms
13:     learn: 0.7503957        total: 42.8ms   remaining: 263ms
14:     learn: 0.7336293        total: 45.8ms   remaining: 260ms
```

```
15:     learn: 0.7180932        total: 48.6ms   remaining: 255ms
16:     learn: 0.7036947        total: 51.3ms   remaining: 250ms
17:     learn: 0.6962835        total: 54.2ms   remaining: 247ms
18:     learn: 0.6853672        total: 57.8ms   remaining: 246ms
19:     learn: 0.6780137        total: 61.7ms   remaining: 247ms
20:     learn: 0.6664308        total: 64.6ms   remaining: 243ms
21:     learn: 0.6552146        total: 67.6ms   remaining: 240ms
22:     learn: 0.6462329        total: 70.5ms   remaining: 236ms
23:     learn: 0.6372409        total: 73.9ms   remaining: 234ms
24:     learn: 0.6278507        total: 78.7ms   remaining: 236ms
25:     learn: 0.6184935        total: 81.5ms   remaining: 232ms
26:     learn: 0.6099269        total: 84.9ms   remaining: 230ms
27:     learn: 0.6026534        total: 87.6ms   remaining: 225ms
28:     learn: 0.5939284        total: 90.9ms   remaining: 223ms
29:     learn: 0.5868452        total: 93.8ms   remaining: 219ms
30:     learn: 0.5795519        total: 96.8ms   remaining: 215ms
31:     learn: 0.5737730        total: 99.8ms   remaining: 212ms
32:     learn: 0.5677996        total: 103ms    remaining: 209ms
33:     learn: 0.5639703        total: 108ms    remaining: 209ms
34:     learn: 0.5573413        total: 110ms    remaining: 205ms
35:     learn: 0.5524005        total: 113ms    remaining: 201ms
36:     learn: 0.5468409        total: 117ms    remaining: 199ms
37:     learn: 0.5419281        total: 120ms    remaining: 195ms
38:     learn: 0.5380596        total: 123ms    remaining: 192ms
39:     learn: 0.5319581        total: 126ms    remaining: 189ms
40:     learn: 0.5283898        total: 130ms    remaining: 187ms
41:     learn: 0.5255842        total: 133ms    remaining: 183ms
42:     learn: 0.5197602        total: 137ms    remaining: 181ms
43:     learn: 0.5155139        total: 140ms    remaining: 179ms
44:     learn: 0.5120505        total: 144ms    remaining: 176ms
45:     learn: 0.5073448        total: 147ms    remaining: 172ms
46:     learn: 0.5027644        total: 150ms    remaining: 169ms
47:     learn: 0.5002999        total: 153ms    remaining: 166ms
48:     learn: 0.4966630        total: 157ms    remaining: 163ms
49:     learn: 0.4927376        total: 160ms    remaining: 160ms
50:     learn: 0.4883528        total: 163ms    remaining: 157ms
51:     learn: 0.4862898        total: 166ms    remaining: 153ms
52:     learn: 0.4835977        total: 169ms    remaining: 150ms
53:     learn: 0.4792803        total: 173ms    remaining: 147ms
54:     learn: 0.4769451        total: 176ms    remaining: 144ms
55:     learn: 0.4746798        total: 180ms    remaining: 141ms
56:     learn: 0.4695351        total: 183ms    remaining: 138ms
57:     learn: 0.4674329        total: 186ms    remaining: 135ms
58:     learn: 0.4654562        total: 188ms    remaining: 131ms
59:     learn: 0.4616406        total: 192ms    remaining: 128ms
60:     learn: 0.4574566        total: 195ms    remaining: 125ms
61:     learn: 0.4552547        total: 198ms    remaining: 122ms
62:     learn: 0.4534912        total: 201ms    remaining: 118ms
63:     learn: 0.4494684        total: 205ms    remaining: 115ms
64:     learn: 0.4475396        total: 208ms    remaining: 112ms
65:     learn: 0.4458013        total: 213ms    remaining: 110ms
66:     learn: 0.4446913        total: 216ms    remaining: 106ms
67:     learn: 0.4431341        total: 220ms    remaining: 103ms
68:     learn: 0.4395648        total: 223ms    remaining: 100ms
69:     learn: 0.4370425        total: 227ms    remaining: 97.2ms
70:     learn: 0.4334834        total: 230ms    remaining: 94.1ms
71:     learn: 0.4310739        total: 233ms    remaining: 90.8ms
72:     learn: 0.4282001        total: 236ms    remaining: 87.3ms
73:     learn: 0.4268663        total: 239ms    remaining: 83.8ms
74:     learn: 0.4254689        total: 241ms    remaining: 80.5ms
75:     learn: 0.4232891        total: 244ms    remaining: 77ms
```

```
76:     learn: 0.4221248        total: 246ms    remaining: 73.6ms
77:     learn: 0.4191207        total: 249ms    remaining: 70.3ms
78:     learn: 0.4162089        total: 252ms    remaining: 66.9ms
79:     learn: 0.4132921        total: 254ms    remaining: 63.6ms
80:     learn: 0.4121373        total: 257ms    remaining: 60.2ms
81:     learn: 0.4103290        total: 259ms    remaining: 56.9ms
82:     learn: 0.4077934        total: 262ms    remaining: 53.6ms
83:     learn: 0.4061003        total: 264ms    remaining: 50.3ms
84:     learn: 0.4050745        total: 267ms    remaining: 47.2ms
85:     learn: 0.4033309        total: 270ms    remaining: 44ms
86:     learn: 0.4019581        total: 273ms    remaining: 40.7ms
87:     learn: 0.4008587        total: 275ms    remaining: 37.5ms
88:     learn: 0.3992354        total: 278ms    remaining: 34.3ms
89:     learn: 0.3967304        total: 280ms    remaining: 31.2ms
90:     learn: 0.3937468        total: 283ms    remaining: 28ms
91:     learn: 0.3928905        total: 285ms    remaining: 24.8ms
92:     learn: 0.3917519        total: 288ms    remaining: 21.7ms
93:     learn: 0.3903306        total: 291ms    remaining: 18.6ms
94:     learn: 0.3883280        total: 294ms    remaining: 15.5ms
95:     learn: 0.3861319        total: 296ms    remaining: 12.4ms
96:     learn: 0.3842445        total: 299ms    remaining: 9.24ms
97:     learn: 0.3828038        total: 302ms    remaining: 6.15ms
98:     learn: 0.3803170        total: 305ms    remaining: 3.08ms
99:     learn: 0.3780856        total: 308ms    remaining: 0us
```



QDA

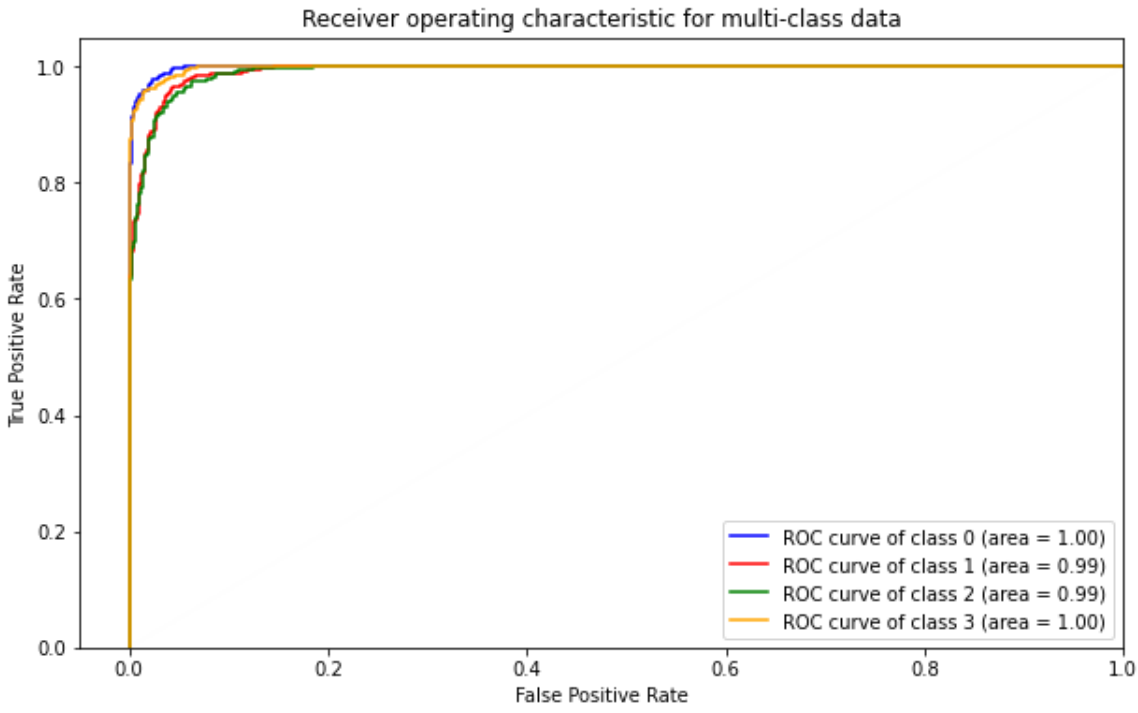
In [199]:

```
estimator_3 = QuadraticDiscriminantAnalysis()
parameters_3 = {
    'reg_param': (0.00001, 0.0001, 0.001,0.01, 0.1),
    'store_covariance': (True, False),
    'tol': (0.0001, 0.001,0.01, 0.1),
                  }
# with GridSearch
grid_search_qda = GridSearchCV(
    estimator=estimator_3,
    param_grid=parameters_3,
    scoring = 'accuracy',
    n_jobs = -1,
    cv = 5
)
%time grid_search_qda.fit(xtrain, ytrain)
qda = grid_search_qda.fit(xtrain, ytrain)
pred_qda =qda.predict(xtest)

print(accuracy_score(ytest, pred_qda))
```

```
Wall time: 8.45 s
0.9316666666666666
```

In [200]:

```
X1 = xtrain.to_numpy()
y1 = ytrain.to_numpy()

# Binarize the output
y_bin = label_binarize(y1, classes=[0, 1, 2, 3])
n_classes = y_bin.shape[1]

clf = qda
y_score = cross_val_predict(clf, X1, y1, cv=10 ,method='predict_proba')

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_bin[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
colors = cycle(['blue', 'red', 'green','orange'])

plt.figure(figsize=(10,6))

for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color,
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(i, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--',linewidth=0.001)
plt.xlim([-0.05, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic for multi-class data')
plt.legend(loc="lower right")
plt.show()
```



KNN

In [201]:

```
def train_knn(feature_train,label_train,n_neighbors):
    # Input: feature data frame, label series, model parameters
    # Output: time to train model, trained model
    start = time.time()
    knn = KNeighborsClassifier(n_neighbors=n_neighbors).fit(feature_train,label_train)
    end = time.time()
    train_time = end-start
    return [train_time,knn]
def compute_metrics(feature_test,label_test,test_preds, model):
    classification_error = np.mean(np.array(test_preds) != np.array(label_test))
    accuracy = 1-classification_error
    test_probs = model.predict_proba(feature_test)[:,1]
    return [accuracy]

def test_model(model, feature_test):
    # Input: test features, a trained model
    # Output: prediction time, test predictions
    start = time.time()
    test_preds = model.predict(feature_test)
    end = time.time()
    prediction_time = end-start
    return [prediction_time,test_preds]
```

In [202]:

```
[train_time, knn] = train_knn(xtrain,ytrain,n_neighbors=25)
print('\nTraining time: {:4f} seconds'.format(train_time))

[prediction_time,test_preds] = test_model(knn,xtest)
print('Prediction time: {:4f} seconds'.format(prediction_time))

[accuracy] = compute_metrics(xtest,ytest,test_preds,knn)
print('\nAccuracy: {:4f}'.format(accuracy))
#print('AUC: {:4f}'.format(auc))

row = pd.Series({#'Feature Extraction Train Time':tm_feature_train_improved,
        #'Feature Extraction Test Time':tm_feature_test_improved,
         'Train Time':train_time,
         'Prediction Time':prediction_time,
        'Accuracy':accuracy},name='KNN')
```

```
Training time: 0.034039 seconds
Prediction time: 0.095580 seconds

Accuracy: 0.950000
```

LDA

In [203]:

```
model = LDA(solver='eigen')
# define model evaluation method
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
# define grid
grid = dict()
grid['shrinkage'] = arange(0, 1, 0.01)
# define search
search = GridSearchCV(model, grid, scoring='accuracy', cv=cv, n_jobs=-1)
# perform the search
results = search.fit(xtrain, ytrain)
y_pred=search.predict(xtest)
# summarize
print(accuracy_score(ytest, y_pred))
print('Mean Accuracy: %.3f' % results.best_score_)
print('Config: %s' % results.best_params_)
```

```
0.9566666666666667
Mean Accuracy: 0.948
Config: {'shrinkage': 0.02}
```

XGBOOST

In [205]:

```
model_xgb = XGBClassifier()

model_xgb.fit(xtrain, ytrain)
pred = model_xgb.predict(xtest)

accuracy = float(np.sum(pred==ytest))/ytest.shape[0]
print(accuracy_score(ytest, pred))
```

```
0.91
```

In [207]:

```
# Incorporating the results from Gridsearch to xgb model
model_xgb_tuned = XGBClassifier(colsample_bytree=0.7,
                                learning_rate=0.1,
                                max_depth=3,
                                min_child_weight=5,
                                n_estimators=500,
                                objective='binary:logistic',
                                subsample=0.7,
                                seed=123,
                                silent=True)

results = model_xgb_tuned.fit(xtrain, ytrain)
y_pred_tuned=model_xgb_tuned.predict(xtest)
print(accuracy_score(ytest, y_pred_tuned))
```

```
[08:33:31] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release
_1.2.0\src\learner.cc:516:
Parameters: { silent } might not be used.

  This may not be accurate due to some parameters are only used in languag
e bindings but
  passed down to XGBoost core.  Or some parameters are not used but slip t
hrough this
  verification. Please open an issue if you find above cases.


0.9216666666666666
```

SVM

In [209]:

```
# basic SVM model
svm_model = SVC()
svm_model.fit(xtrain, ytrain)
test_preds = svm_model.predict(xtest)
accuracy = accuracy_score(ytest, test_preds)
print('\nAccuracy: {:4f}'.format(accuracy))
```

```
Accuracy: 0.955000
```

In [211]:

```
# grid search
# 1. Kernel: transform the dataset into the required form,
# choose 'linear', 'poly' and 'rbf' since teh datset is not complex
# 2. C: regularization, smaller C is less error bearable
# 3. degree: degree for poly
params = {'C': [0.001,0.01,1,10,15,20],
        'kernel':['linear', 'rbf', 'poly'],
        'degree':[2,3,4]}
gscv = GridSearchCV(SVC(random_state = 2020), params, cv=5, return_train_score=True)
gscv.fit(xtrain, ytrain)
gscv.best_params_
# #output: {'C': 15, 'degree': 2, 'kernel': 'linear'}
#improved svm using parameters from grid search
# basic SVM model
svm_model = SVC(C=15,
                kernel='linear',
                degree=2)
svm_model.fit(xtrain, ytrain)
test_preds = svm_model.predict(xtest)
accuracy = accuracy_score(ytest, test_preds)
print('\nAccuracy: {:4f}'.format(accuracy))
```

```
Accuracy: 0.978333
```