In [1]:
```
import numpy as np
from sklearn.model_selection import KFold
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import roc_auc_score
import matplotlib.pyplot as plt

from sklearn import datasets, metrics, model_selection, svm


data = pd.read_csv("C:/Users/76380/Desktop/Honda.csv")
```

In [ ]:
```
import numpy as np
from sklearn.model_selection import KFold
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import roc_curve, auc
data = pd.read_csv("C:/Users/76380/Desktop/Honda.csv")


X = data.iloc[:,:4]
Y = data.iloc[:,4:5]
X = np.array(X)
Y = np.array(Y)
data = np.array(data)

clf = GaussianNB()
clf.fit(X, Y)
GaussianNB()
clf.predict(X)
#print(clf.predict([[-0.8, -1,50000,7000]]))
```

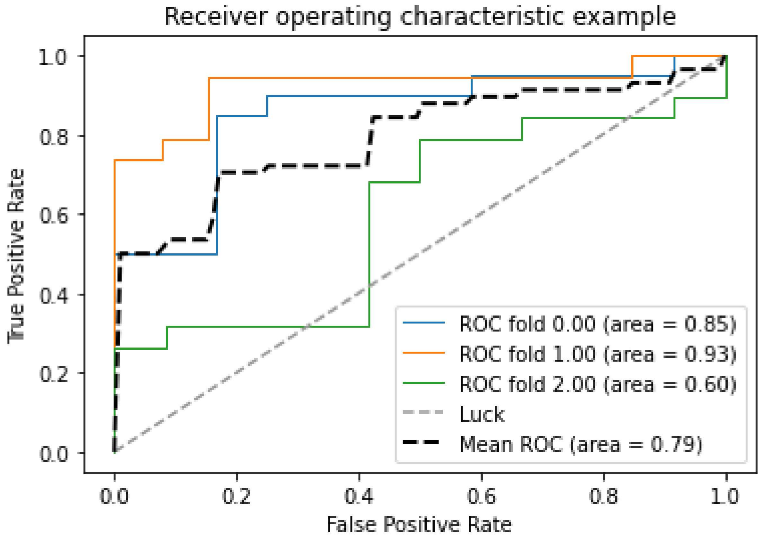
In [ ]:
```
import numpy as np
from sklearn.model_selection import KFold
import pandas as pd

from sklearn.metrics import roc_curve, auc
from sklearn.dummy import DummyClassifier
data = pd.read_csv("C:/Users/76380/Desktop/Honda.csv")


X = data.iloc[:,:4]
Y = data.iloc[:,4:5]
X = np.array(X)
Y = np.array(Y)
data = np.array(data)

dummy_clf = DummyClassifier(strategy="stratified")
dummy_clf.fit(X, Y)
DummyClassifier(strategy='stratified')
dummy_clf.predict(X)
#print(clf.predict([[-0.8, -1,50000,7000]]))
```

In [24]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
```

```
from sklearn.model_selection import StratifiedKFold

data = pd.read_csv("C:/Users/76380/Desktop/Honda.csv")
#data = np.array(data)
X = data.iloc[:,:4]
Y = data.iloc[:,4:5]
X = np.array(X)
y = []



a = np.array(Y)
y =[]
for m in range(0,95):
    for i in a[m]:
        y.append(i)
y = np.array(y)


X, y = X[y != 2], y[y != 2]
n_samples, n_features = X.shape

random_state = np.random.RandomState(0)
X = np.c_[X, random_state.randn(n_samples, 200 * n_features)]

cv = StratifiedKFold(n_splits=3)

classifier = GaussianNB()

mean_tpr = 0.0
mean_fpr = np.linspace(0, 1, 100)
cnt = 0
for i, (train, test) in enumerate(cv.split(X,y)):
    cnt +=1
    probas_ = classifier.fit(X[train], y[train]).predict_proba(X[test])
    y_pred = classifier.fit(X[train], y[train]).predict(X[test])
    print("Number of mislabeled points out of a total %d points : %d" % (X[test].sha
    print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
    fpr, tpr, thresholds = roc_curve(y[test], probas_[:, 1])

    mean_tpr += np.interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=1, label='ROC fold {0:.2f} (area = {1:.2f})'.format(i, roc

plt.plot([0, 1], [0, 1], '--', color=(0.6, 0.6, 0.6), label='Luck')

mean_tpr /= cnt
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)

plt.plot(mean_fpr, mean_tpr, 'k--',label='Mean ROC (area = {0:.2f})'.format(mean_auc

plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

```
Number of mislabeled points out of a total 32 points : 10
Accuracy: 32 percent
Number of mislabeled points out of a total 32 points : 4
Accuracy: 32 percent
Number of mislabeled points out of a total 31 points : 15
Accuracy: 31 percent
```

```
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\1398343822.py:42: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\1398343822.py:42: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\1398343822.py:42: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
```



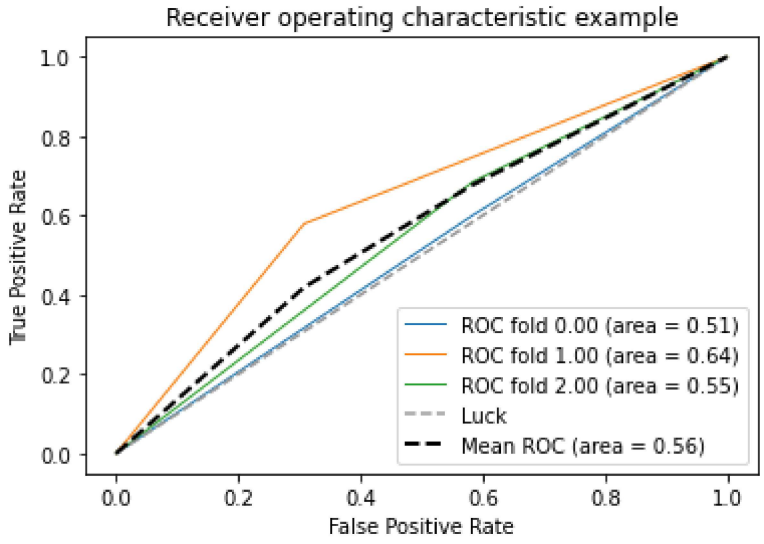
In [23]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import StratifiedKFold

data = pd.read_csv("C:/Users/76380/Desktop/Honda.csv")
#data = np.array(data)
X = data.iloc[:,:4]
Y = data.iloc[:,4:5]
X = np.array(X)
y = []


a = np.array(Y)
y =[]
for m in range(0,95):
    for i in a[m]:
        y.append(i)
y = np.array(y)



X, y = X[y != 2], y[y != 2]
n_samples, n_features = X.shape

random_state = np.random.RandomState(0)
X = np.c_[X, random_state.randn(n_samples, 200 * n_features)]
```

```
cv = StratifiedKFold(n_splits=3)

classifier = DummyClassifier(strategy="stratified")

mean_tpr = 0.0
mean_fpr = np.linspace(0, 1, 100)
cnt = 0
for i, (train, test) in enumerate(cv.split(X,y)):
    cnt +=1
    probas_ = classifier.fit(X[train], y[train]).predict_proba(X[test])
    y_pred = classifier.fit(X[train], y[train]).predict(X[test])
    print("Number of mislabeled points out of a total %d points : %d" % (X[test].sha
    print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
    fpr, tpr, thresholds = roc_curve(y[test], probas_[:, 1])

    mean_tpr += np.interp(mean_fpr, fpr, tpr)
    mean_tpr[0] = 0.0
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=1, label='ROC fold {0:.2f} (area = {1:.2f})'.format(i, roc

plt.plot([0, 1], [0, 1], '--', color=(0.6, 0.6, 0.6), label='Luck')

mean_tpr /= cnt
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)

plt.plot(mean_fpr, mean_tpr, 'k--',label='Mean ROC (area = {0:.2f})'.format(mean_auc

plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

```
Number of mislabeled points out of a total 32 points : 16
Accuracy: 32 percent
Number of mislabeled points out of a total 32 points : 14
Accuracy: 32 percent
Number of mislabeled points out of a total 31 points : 15
Accuracy: 31 percent
```

```
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\3442226410.py:43: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\3442226410.py:43: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
C:\Users\76380\AppData\Local\Temp\ipykernel_2276\3442226410.py:43: RuntimeWarning: d
ivide by zero encountered in divide
  print("Accuracy: %d percent" %(X[test].shape[0]/y[test] != y_pred).sum())
```


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0.00 (area = 0.51)
ROC fold 1.00 (area = 0.64)
ROC fold 2.00 (area = 0.55)
Luck
Mean ROC (area = 0.56)

In [ ]:

In [ ]: