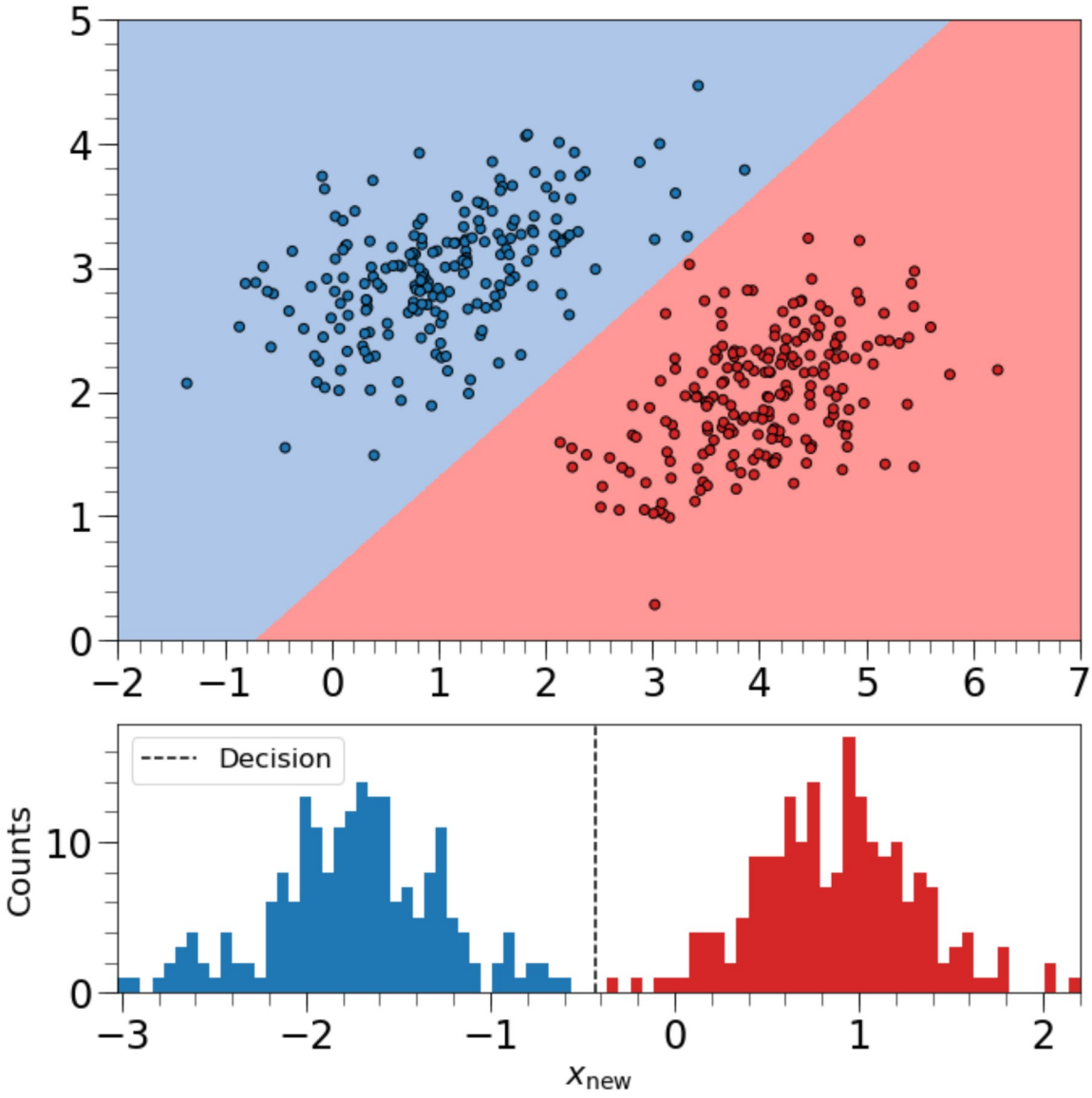
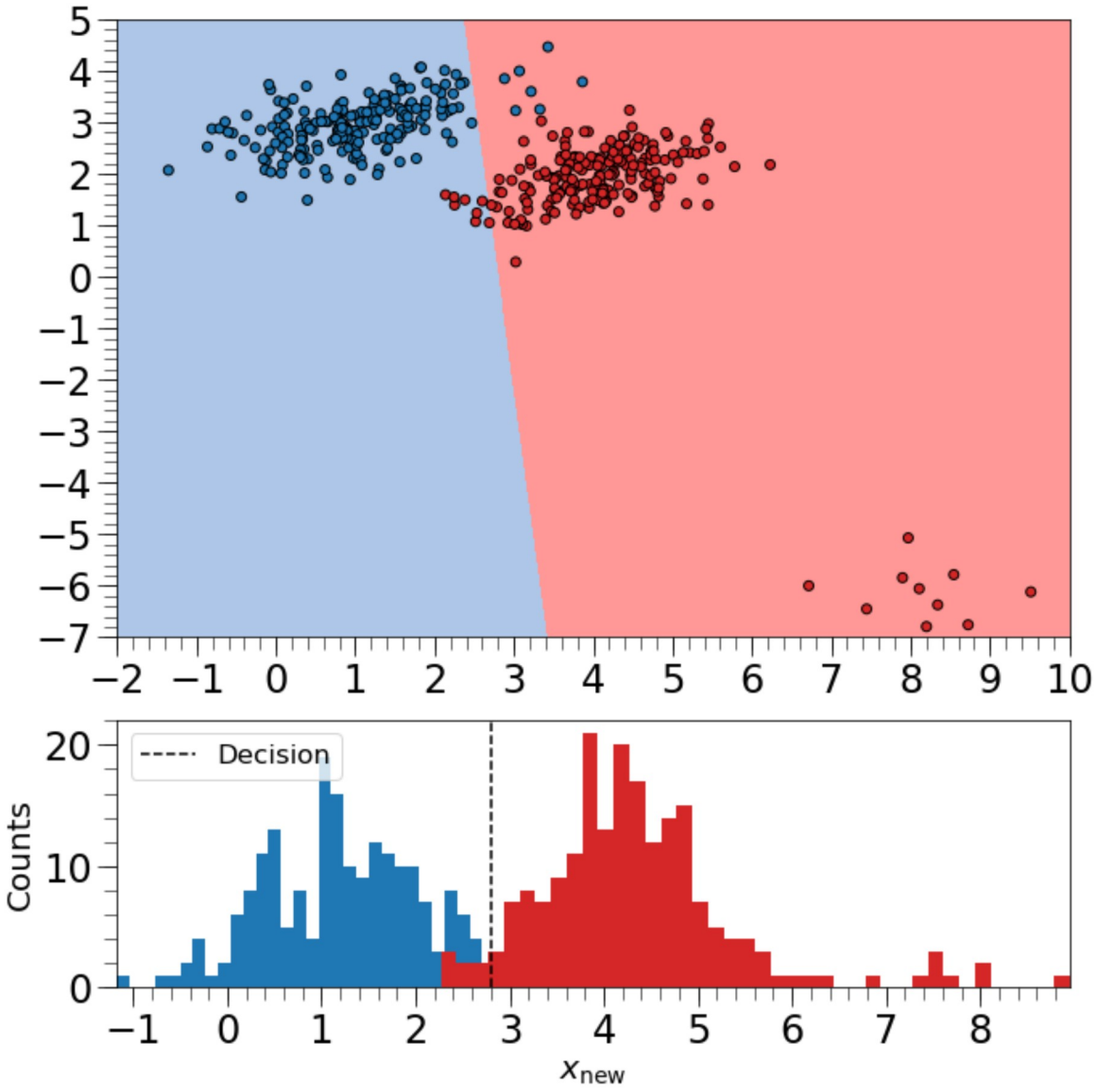
In [11]:
```
# -*- coding: utf-8 -*-
"""
Created on Sun Oct  6 01:22:34 2019

@author: jorge
"""

import numpy as np
from requiredFunctions.fishersDiscriminant import FishersDiscriminant
from requiredFunctions.doubleMoon import doubleMoon
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick


height = 10
width = 10

mpl.rcParams['figure.figsize'] = (width, height)
mpl.rcParams['font.size'] = 20
mpl.rcParams['figure.titlesize'] = 'small'
mpl.rcParams['legend.fontsize'] = 'small'
mpl.rcParams['xtick.major.size'] = 12
mpl.rcParams['xtick.minor.size'] = 8
mpl.rcParams['xtick.labelsize'] = 24
mpl.rcParams['ytick.major.size'] = 12
mpl.rcParams['ytick.minor.size'] = 8
mpl.rcParams['ytick.labelsize'] = 24

data = ['DatasetA', 'DatasetA2', 'double moon']
part = ['a', 'b', 'c']
```

In [12]:
```
# First run is without the w_1 term, second is with it.
for i in range(2):
    for p, d in enumerate(data):
        if d == 'double moon':
            dubmoon = doubleMoon(5000, 0.6, 1, -0.1, seed=0)
            train_x, train_y = dubmoon[:,:2], dubmoon[:,-1]
        else:
            train_x = np.loadtxt('../data/' + d + '_data.csv', delimiter=',')
            train_y = np.loadtxt('../data/' + d + '_labels.csv', delimiter=',')

        fishers = FishersDiscriminant(train_x, train_y)
        fishers.discriminant = fishers.discriminant - i*fishers.weights[0]
        accuracy = fishers.score(train_x, train_y)
        print(d, 'Accuracy:', accuracy)

        x0_min, x0_max = np.floor(train_x[:,0].min()), np.ceil(train_x[:,0].max())
        x1_min, x1_max = np.floor(train_x[:,1].min()), np.ceil(train_x[:,1].max())

        xx0, xx1 = np.meshgrid(np.arange(x0_min, x0_max, 0.01),
                               np.arange(x1_min, x1_max, 0.01))

        cc = fishers.predict(np.c_[xx0.ravel(), xx1.ravel()]).reshape(xx0.shape)

        cmap = plt.get_cmap('tab20')
        cmap_scatter = mpl.colors.ListedColormap(cmap((0, 6)))
        cmap_contour = mpl.colors.ListedColormap(cmap((1, 7)))

        fig = plt.figure()
        ax = [plt.subplot2grid((3,2), (0,0), colspan=2, rowspan = 2, fig=fig),
              plt.subplot2grid((3,2), (2,0), colspan=2, rowspan = 1, fig=fig)]

        ax[0].contourf(xx0, xx1, cc, cmap=cmap_contour)
        ax[0].scatter(train_x[:,0], train_x[:,1], c=train_y,
                      cmap=cmap_scatter, edgecolor='black')
        ax[0].set_xlim(x0_min, x0_max)
        ax[0].set_ylim(x1_min, x1_max)
        ax[0].xaxis.set_major_locator(mtick.MultipleLocator(1))
        ax[0].xaxis.set_minor_locator(mtick.MultipleLocator(0.2))
        ax[0].yaxis.set_major_locator(mtick.MultipleLocator(1))
        ax[0].yaxis.set_minor_locator(mtick.MultipleLocator(0.2))

        #ax.plot(fishers.means[0,:], fishers.means[1,:], color='lightgreen', linewidt
h=5)

        xnew_0 = fishers.project(train_x[train_y == 0])
        xnew_1 = fishers.project(train_x[train_y == 1])
        ax[1].hist(xnew_0, 40, color=cmap_scatter((0)))
        ax[1].hist(xnew_1, 40, color=cmap_scatter((1)))
        ax[1].axvline(fishers.discriminant, color='black', linestyle='--', label='Deci
sion')
        ax[1].set_xlabel(r'$x_{\mathrm{new}}$')
        ax[1].set_ylabel('Counts')
        ax[1].set_xlim(xnew_0.min(), xnew_1.max())
        ax[1].xaxis.set_major_locator(mtick.MultipleLocator(1))
        ax[1].xaxis.set_minor_locator(mtick.MultipleLocator(0.2))
        if d == 'double moon':
            ax[1].yaxis.set_major_locator(mtick.MultipleLocator(100))
            ax[1].yaxis.set_minor_locator(mtick.MultipleLocator(10))
        else:
            ax[1].yaxis.set_major_locator(mtick.MultipleLocator(10))
            ax[1].yaxis.set_minor_locator(mtick.MultipleLocator(2))
        ax[1].legend(loc='upper left')
        fig.tight_layout(h_pad=0)
        plt.savefig('../prob4'+part[p]+str(i)+'.eps', dpi=500)

        plt.show()
```

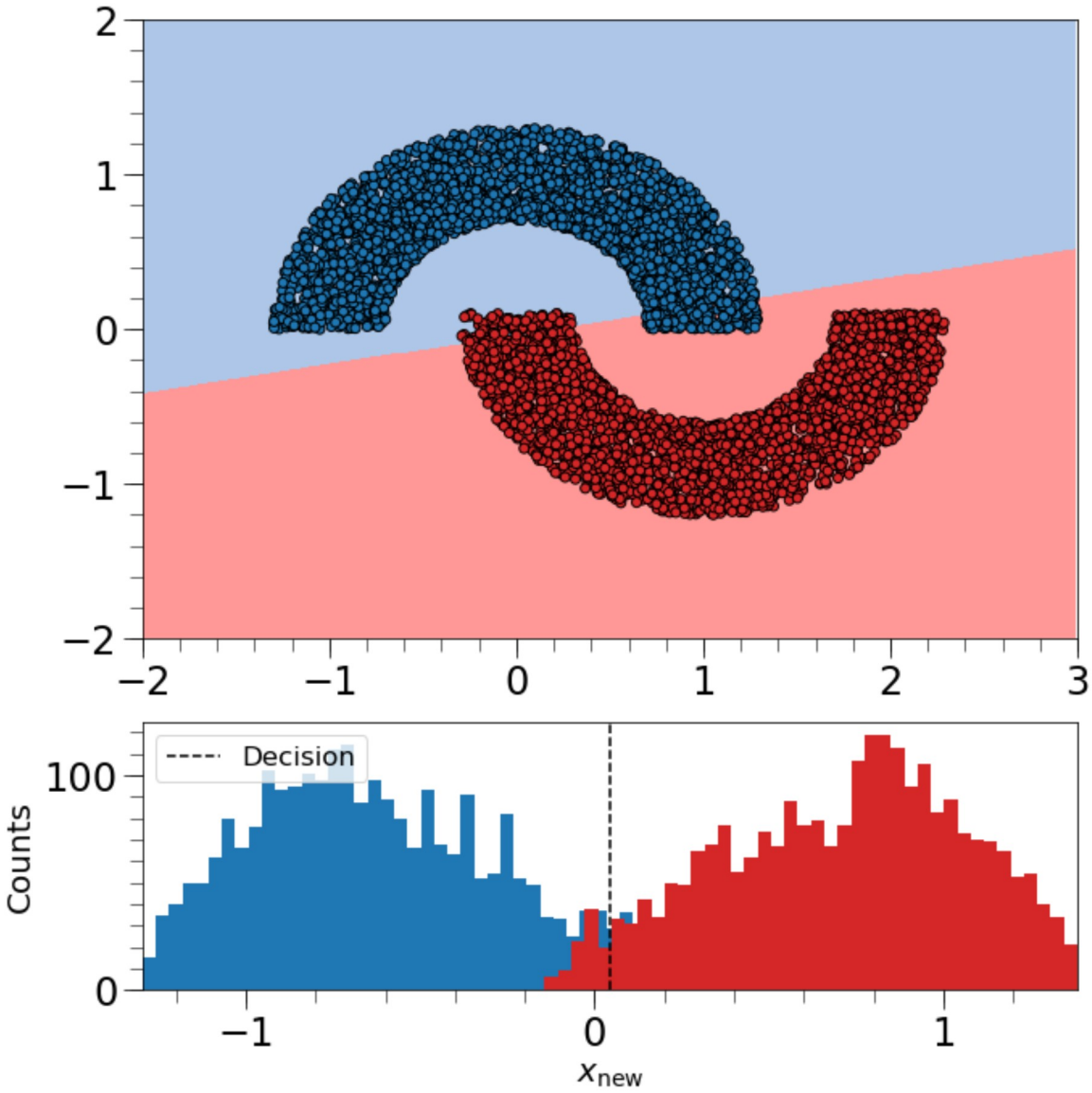
```
DatasetA Accuracy: 1.0
```



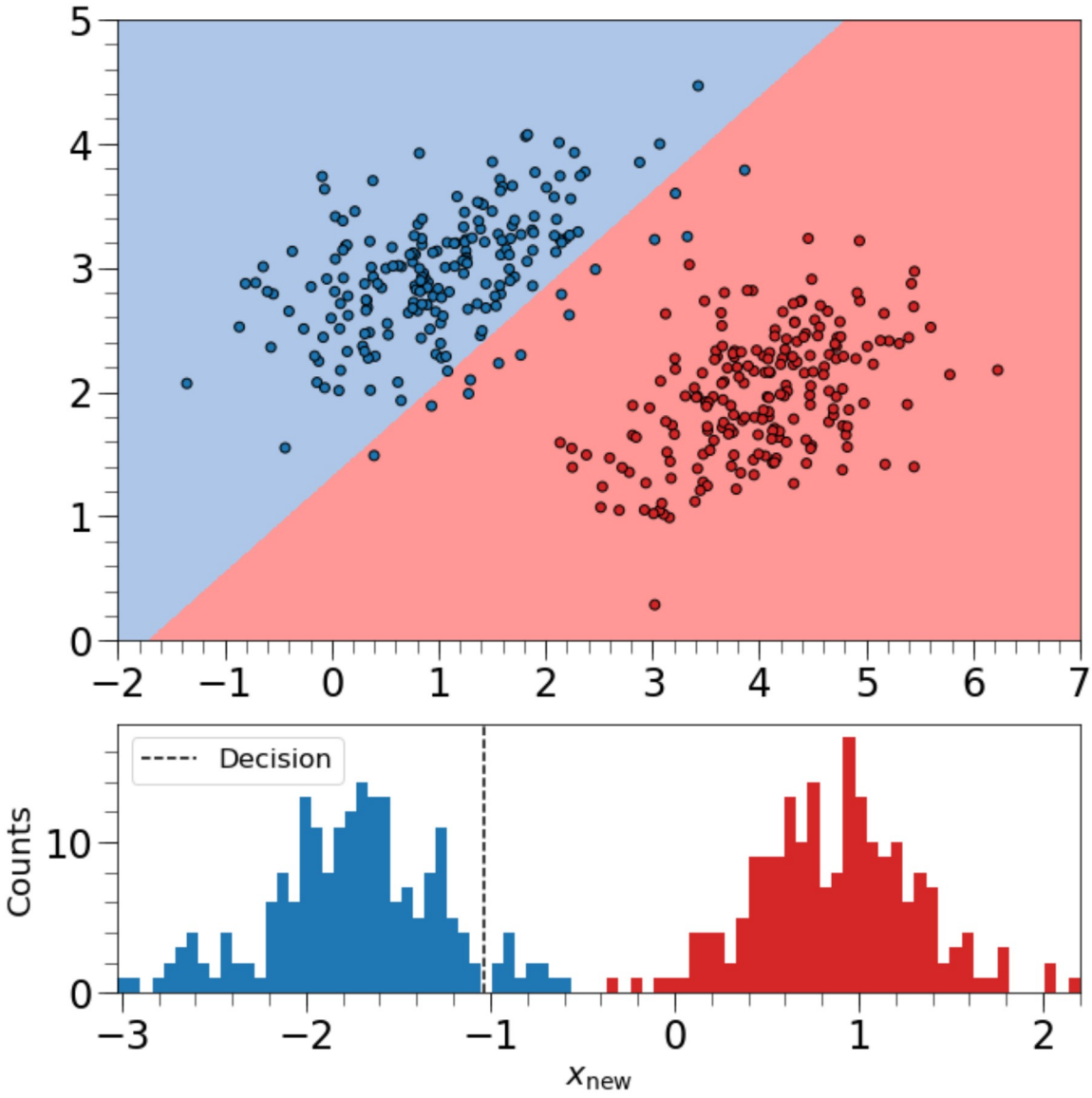

```
DatasetA2 Accuracy: 0.9625
```



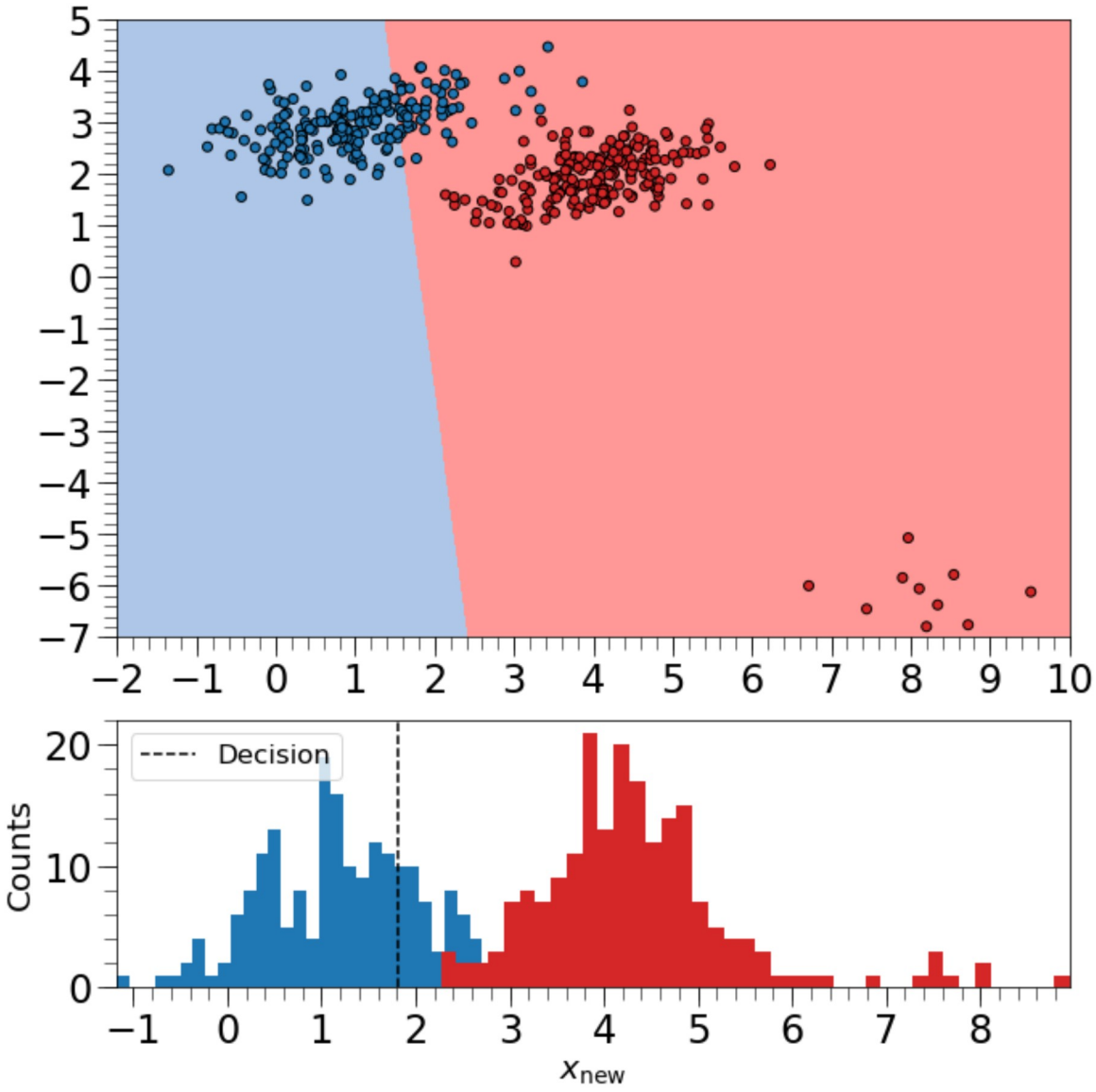

```
double moon Accuracy: 0.9582
```



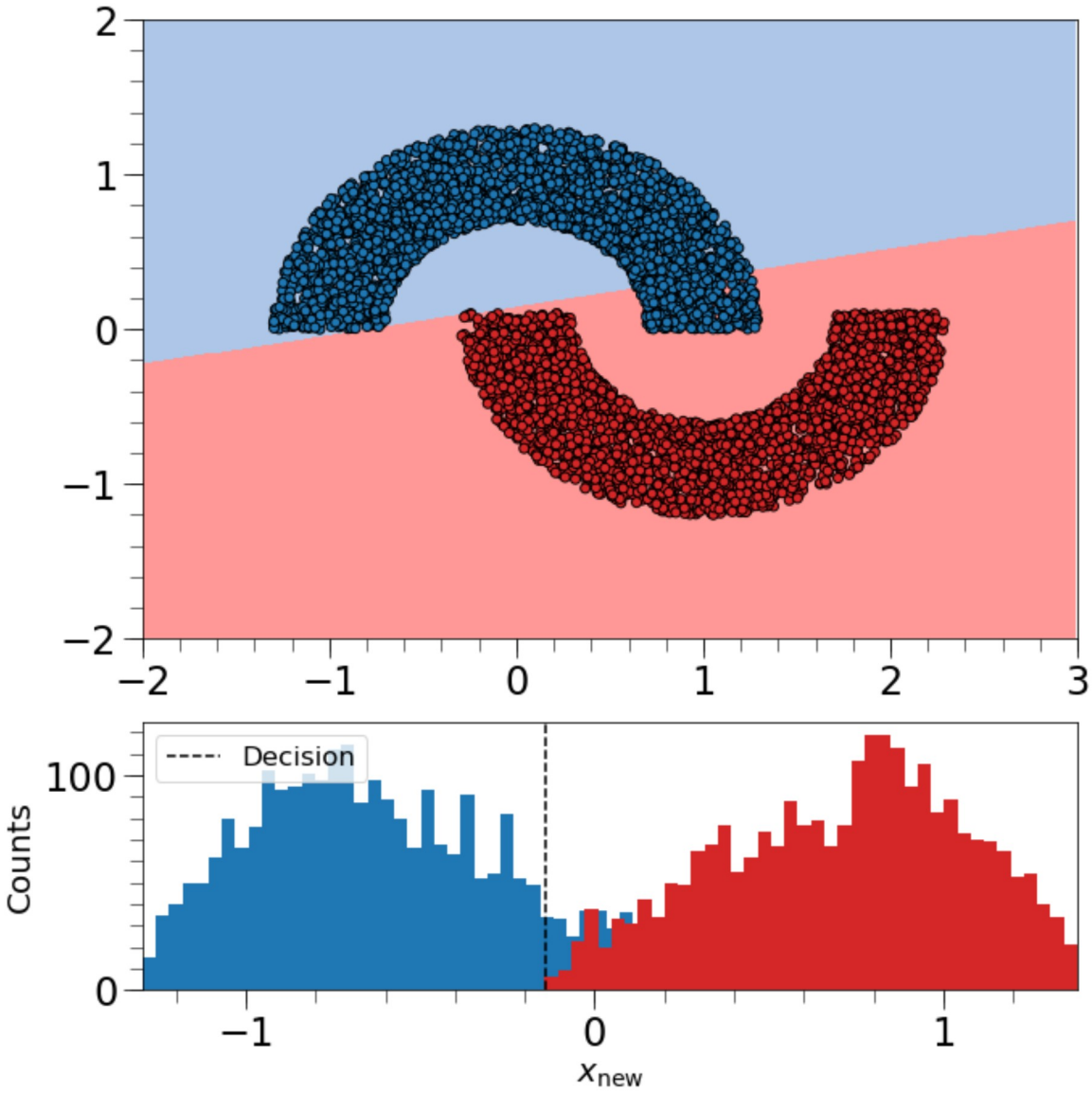

```
DatasetA Accuracy: 0.9675
```




```
DatasetA2 Accuracy: 0.8625
```




```
double moon Accuracy: 0.9462
```



In [ ]: