CS/ECE/ME532 Activity 8

Estimated Time: 15 min for P1, 15 min for P2, 20 min for P3, 5 min for P4, 15 min for P5

1. **Binary linear classifiers.** Assume there are two possible labels, $y = 1$ or $y = -1$ associated with two features x_1 and x_2. We consider several different linear classifiers $\hat{y} = \text{sign}\{\boldsymbol{x}^T\boldsymbol{w}\}$ where $\boldsymbol{x}$ is derived from x_1 and x_2 and $\boldsymbol{w}$ are the classifier weights. Define the decision boundary of the classifier as the set $\{x_1, x_2\}$ for which $\boldsymbol{x}^T\boldsymbol{w} = 0$. Let x_2 be the vertical axis in your sketches and depict the interval $0 \leq x_1, x_2 \leq 1$.

   a) *Classifier 1.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 5 \\ -2 \end{bmatrix}$.

      i. Sketch the decision boundary in the x_1-x_2 plane.

      ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

   b) *Classifier 2.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix}$.

      i. Sketch the decision boundary in the x_1-x_2 plane.

      ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

   c) *Classifier 3.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1^2 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$.

      i. Sketch the decision boundary in the x_1-x_2 plane.

      ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

2. *Linear Classifier.* Download the script and the data file `classifier_data.mat`. This code trains linear classifiers using least squares. The scripts provided steps through the problems below.

   Make sure to 'publish' your results and store them as a PDF file for submission.

   a) *Classifier 1.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$. Briefly comment on the fit of the classifier to the decision boundary apparent in the evaluation data. Also identify the percent error based on the ratio of misclassified evaluation data points to the total number of evaluation data points.

b) *Classifier 2.* Consider squaring the original features, and also using them for classification, so that: $\boldsymbol{x}^T = \begin{bmatrix} x_1^2 & x_2^2 & x_1 & x_2 & 1 \end{bmatrix}$. This will allow for a curved decision boundary. Briefly comment on the fit of the classifier to the decision boundary apparent in the evaluation data. Also identify the percent error based on the ratio of misclassified evaluation data points to the total number of evaluation data points.

c) *Shortcoming of training using least squares as a loss function.* Training a classifier using the squared error as a loss function can fail when correctly labeled data points lie far from the decison boundary. A new dataset consisting of the first dataset, plus 1000 (correctly labeled) datapoints at $x_1 = 0, x_2 = 3$ is created. What happens to the decision boundary when these new data points are included in training? What happens to the error rate if you move the 1000 data points to $x_1 = 0, x_2 = 10$? Why does this happen?

3. *Overfitting.* Download the dataset `overfitting_data.mat`. You may find it helpful to adapt the code from the previous problem. The dataset has 50 data points for training, and 10,000 data points to be used for evaluation of the classifier. Each data point consists a two-dimensional feature vector $\boldsymbol{x}$ and a label $\boldsymbol{y} \in \{-1, 1\}$. The feature vector is a "noisy" version of the true underlying feature, which blurs the boundary between classes.

a) Plot the training data using a scatter plot. Indicate the points $y = -1$ using one color, and the points with $y = 1$ with another.

b) Plot the evaluation data using a scatter plot. Indicate the points labeled -1 using one color, and the points labeled $+1$ with another.

c) Classifier 1. As before, $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, and $y = \text{sign}(\boldsymbol{x}^T \boldsymbol{w})$.

   i. Train the classifier using least squares to find the classifier weights $\boldsymbol{w}$. Apply the classifier to the evaluation data, and plot the data points using a scatter plot with different colors for different predicted labels.

   ii. Plot the correctly predicted evaluation data points using one color, and incorrectly predicted points using a second color. How many errors are there?

d) Classifier 2. Let $\boldsymbol{x} = \begin{bmatrix} x_1^2 & x_2^2 & x_1 & x_2 & 1 \end{bmatrix}^T$, and $y = \text{sign}(\boldsymbol{x}^T \boldsymbol{w})$.

   i. Train the classifier using least squares to find the classifier weights $\boldsymbol{w}$. Apply the classifier to the evaluation data, and plot the data points using a scatter plot with different colors for different predicted labels.

   ii. Plot the correctly predicted evaluation data points using one color, and incorrectly predicted points using a second color. How many errors are there?

e) Classifier 3. Let $\boldsymbol{x} = \begin{bmatrix} x_1^6 & x_2^6 & x_1^5 & x_2^5 & \dots & x_1 & x_2 & 1 \end{bmatrix}^T$, and $y = \text{sign}(\boldsymbol{x}^T\boldsymbol{w})$.

i. Train the classifier using least squares to find the classifier weights $\boldsymbol{w}$. Apply the classifier to the evaluation data, and plot the data points using a scatter plot with different colors for different predicted labels.

ii. Plot the correctly predicted evaluation data points using one color, and incorrectly predicted points using a second color. How many errors are there?

f) Of the three classifiers, which one performs worse, and why?

4. A binary linear classifier based on three features x_1, x_2, and x_3 is $\hat{y} = \text{sign}\{\boldsymbol{x}^T\boldsymbol{w}\}$ where $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$. Hence the decision boundary is the set $\{x_1, x_2, x_3\}$ for which $\boldsymbol{x}^T\boldsymbol{w} = 0$.

The decision boundary for a two-dimensional classifier is a line. What type of geometric object is the decision boundary in three dimensions?

5. A decision boundary for a classification problem involving features x_1, x_2, and x_3 is defined as $\boldsymbol{x}^T\boldsymbol{w} = 0$ where $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 & 1 \end{bmatrix}$. Find $\boldsymbol{w}$ so that the decision boundary is parallel to the x_1-x_2 plane and includes the point $(x_1, x_2, x_3) = (0, 0, 1)$.

532 Activity 8 - DEVIN BRESSER

1. Binary linear classifiers. Assume there are two possible labels, $y = 1$ or $y = -1$ associated with two features x_1 and x_2. We consider several different linear classifiers $\hat{y} = \text{sign}\{\boldsymbol{x}^T\boldsymbol{w}\}$ where $\boldsymbol{x}$ is derived from x_1 and x_2 and $\boldsymbol{w}$ are the classifier weights. Define the decision boundary of the classifier as the set $\{x_1, x_2\}$ for which $\boldsymbol{x}^T\boldsymbol{w} = 0$. Let x_2 be the vertical axis in your sketches and depict the interval $0 \le x_1, x_2 \le 1$.

a) *Classifier 1.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 5 \\ -2 \end{bmatrix}$.

i. Sketch the decision boundary in the x_1-x_2 plane.

ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

b) *Classifier 2.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix}$.

i. Sketch the decision boundary in the x_1-x_2 plane.

ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

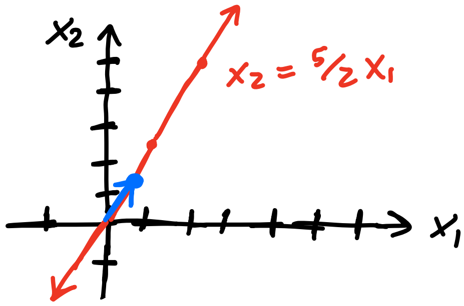
c) *Classifier 3.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1^2 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$.

i. Sketch the decision boundary in the x_1-x_2 plane.

ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

a.) $\underline{x}^T \underline{w} = [x_1 \;\; x_2] \begin{bmatrix} 5 \\ -2 \end{bmatrix} \Rightarrow 5x_1 - 2x_2 = 0$ decision boundary.

$\rightarrow 2x_2 = 5x_1 \rightarrow x_2 = 5/2\, x_1 \rightarrow$



b.) Yes, represents a 1-D subspace in $\mathbb{R}^2$ because goes through $\underline{0}$.

$x_2 = 5/2\, x_1 \quad (1)$

$\sqrt{x_1^2 + x_2^2} = 1 \quad (2)$

$\rightarrow \sqrt{x_1^2 + (5/2\, x_1)^2} = 1 \quad \rightarrow x_2 = 5/2 \cdot \sqrt{4/29}$

$\rightarrow x_1^2 + 25/4\, x_1^2 = 1 \quad \rightarrow x_2 = 5/\sqrt{29}$

$\rightarrow 29/4\, x_1^2 = 1$

$\rightarrow x_1 = \sqrt{4/29}$

$\rightarrow x_1 = 2/\sqrt{29}$

$\rightarrow \begin{bmatrix} 2/\sqrt{29} \\ 5/\sqrt{29} \end{bmatrix}$ is orthonormal basis

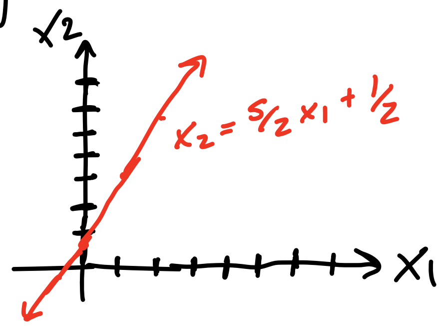
b) *Classifier 2.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix}$.

i. Sketch the decision boundary in the x_1-x_2 plane.

ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

b.) $\underline{x}^T \underline{w} = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix} \Rightarrow 5x_1 - 2x_2 + 1 = 0$ decision boundary

$x_2 = \dfrac{5x_1 + 1}{2} = \frac{5}{2} x_1 + \frac{1}{2}$



Not a subspace because 1-D line does not pass through the origin.

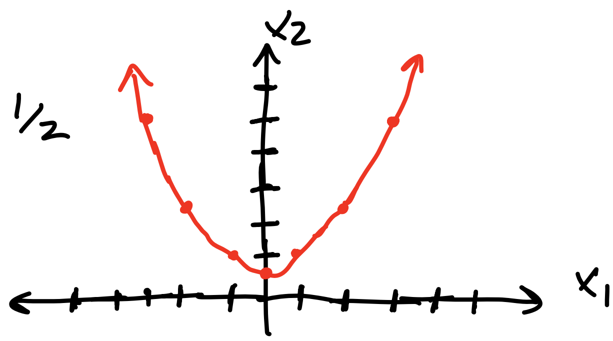
c) *Classifier 3.* Let $\boldsymbol{x}^T = \begin{bmatrix} x_1^2 & x_2 & 1 \end{bmatrix}$ and assume $\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$.

i. Sketch the decision boundary in the x_1-x_2 plane.

ii. Does the decision boundary represent a subspace in $\mathbb{R}^2$? Why or why not? If it represents a subspace, then find an orthonormal basis for the subspace.

c.) $\underline{x}^T \underline{w} = \begin{bmatrix} x_1^2 & x_2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \Rightarrow x_1^2 - 2x_2 + 1 = 0$ decision boundary.

$\rightarrow \dfrac{x_1^2 + 1}{2} = x_2 \rightarrow x_2 = \frac{1}{2} x_1^2 + \frac{1}{2}$

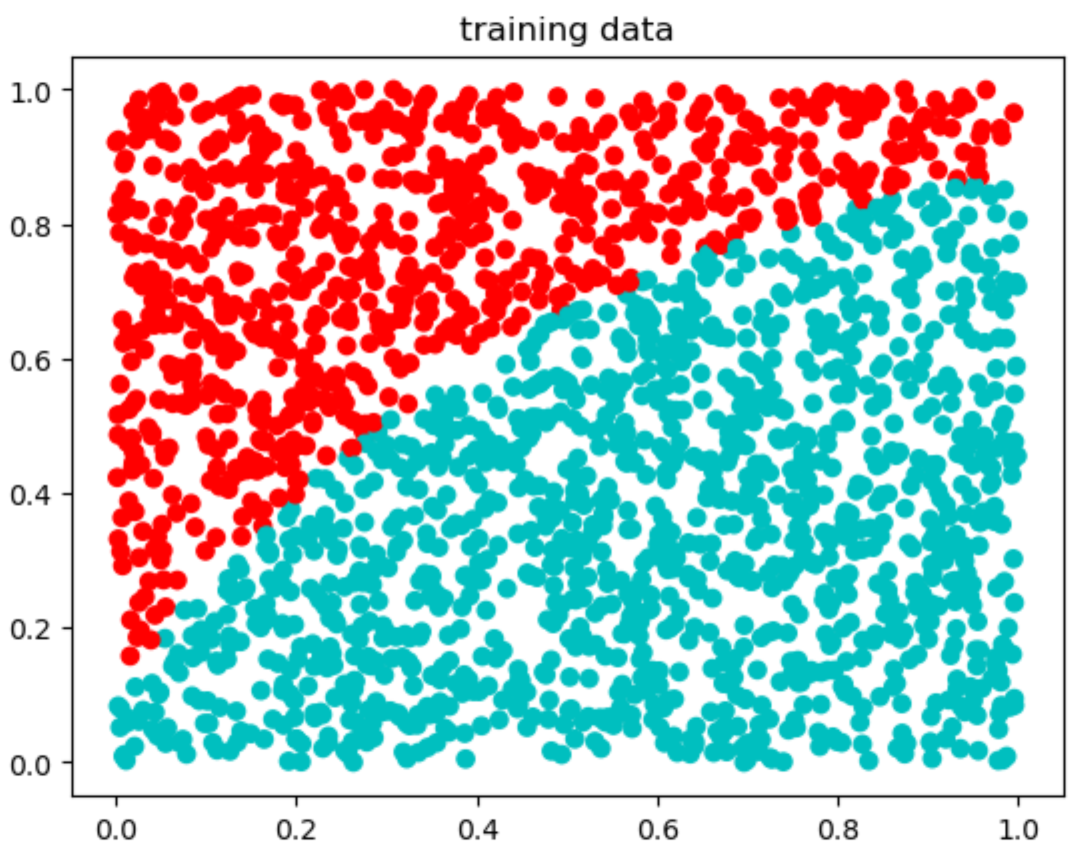


Not a subspace because decision boundary does not pass through $\underline{0}$.

2a)

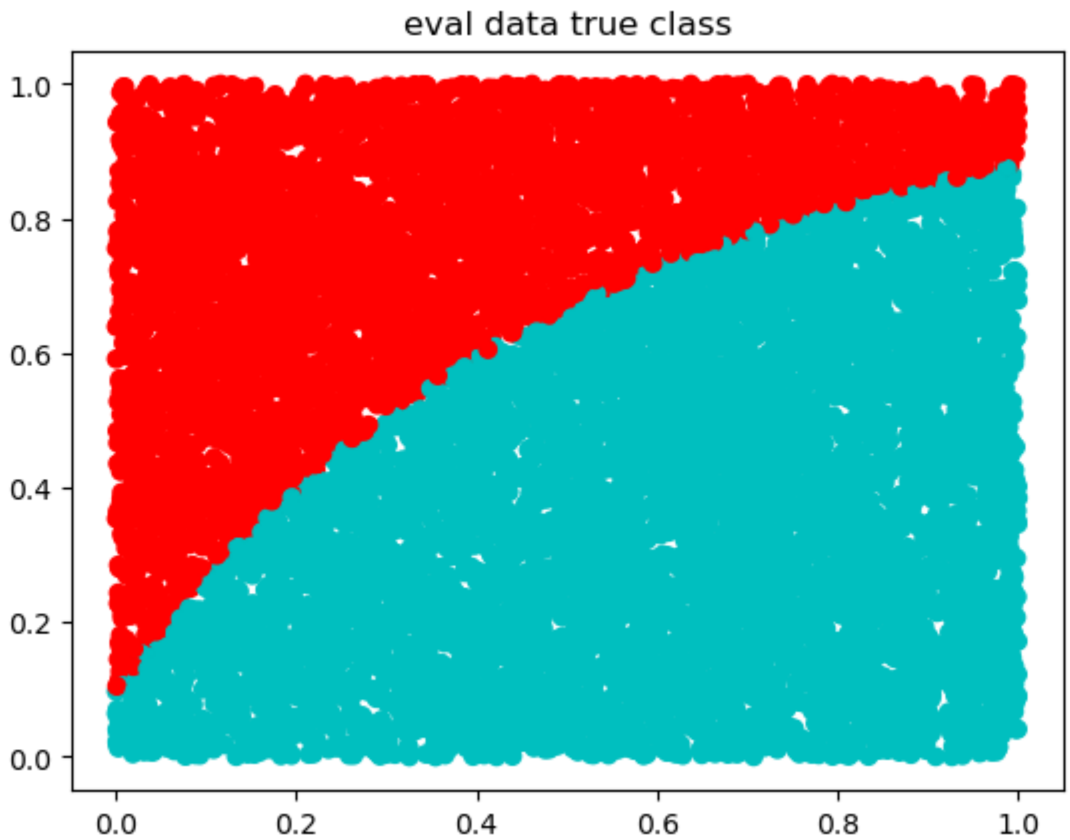
In [1]:

```
import numpy as np
from scipy.io import loadmat
import matplotlib.pyplot as plt

in_data = loadmat('classifier_data.mat')
#print([key for key in in_data]) # -- use this line to see the keys in the dictionary da

x_train = in_data['x_train']
x_eval = in_data['x_eval']
y_train = in_data['y_train']
y_eval = in_data['y_eval']

n_eval = np.size(y_eval)
n_train = np.size(y_train)

plt.scatter(x_train[:,0],x_train[:,1], color=['c' if i==-1 else 'r' for i in y_train[:,0
plt.title('training data')
plt.show()
```

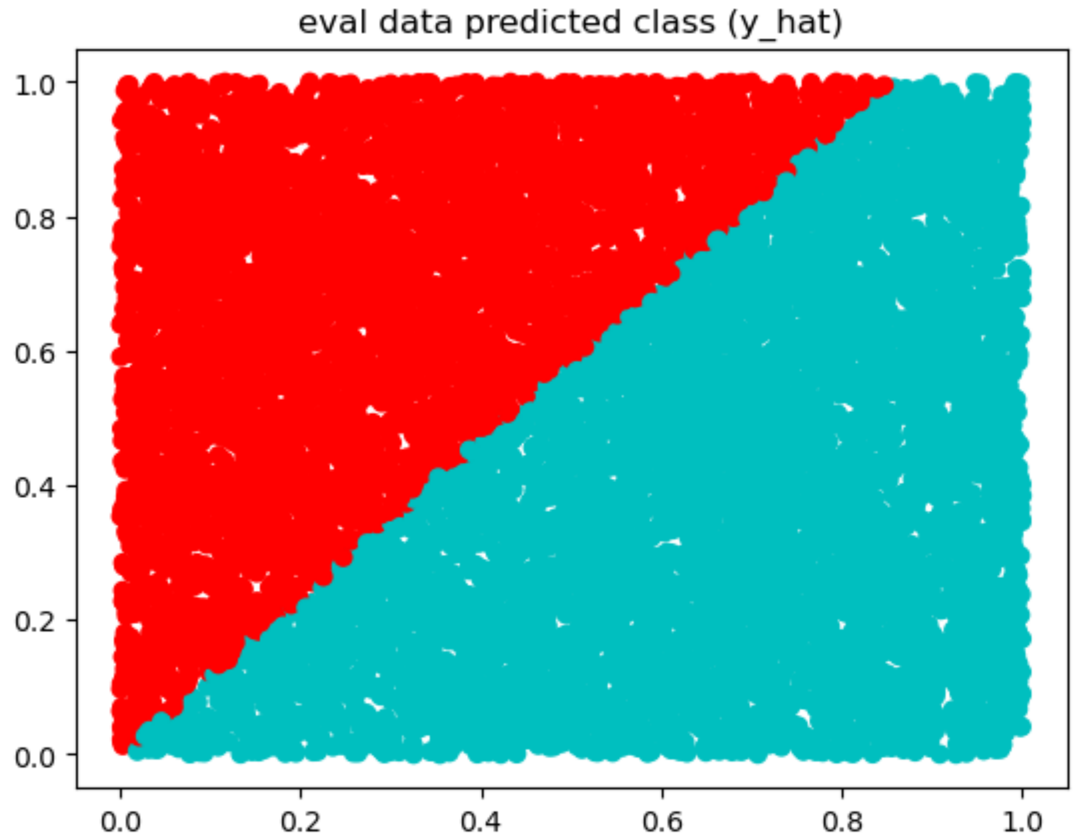


In [2]:

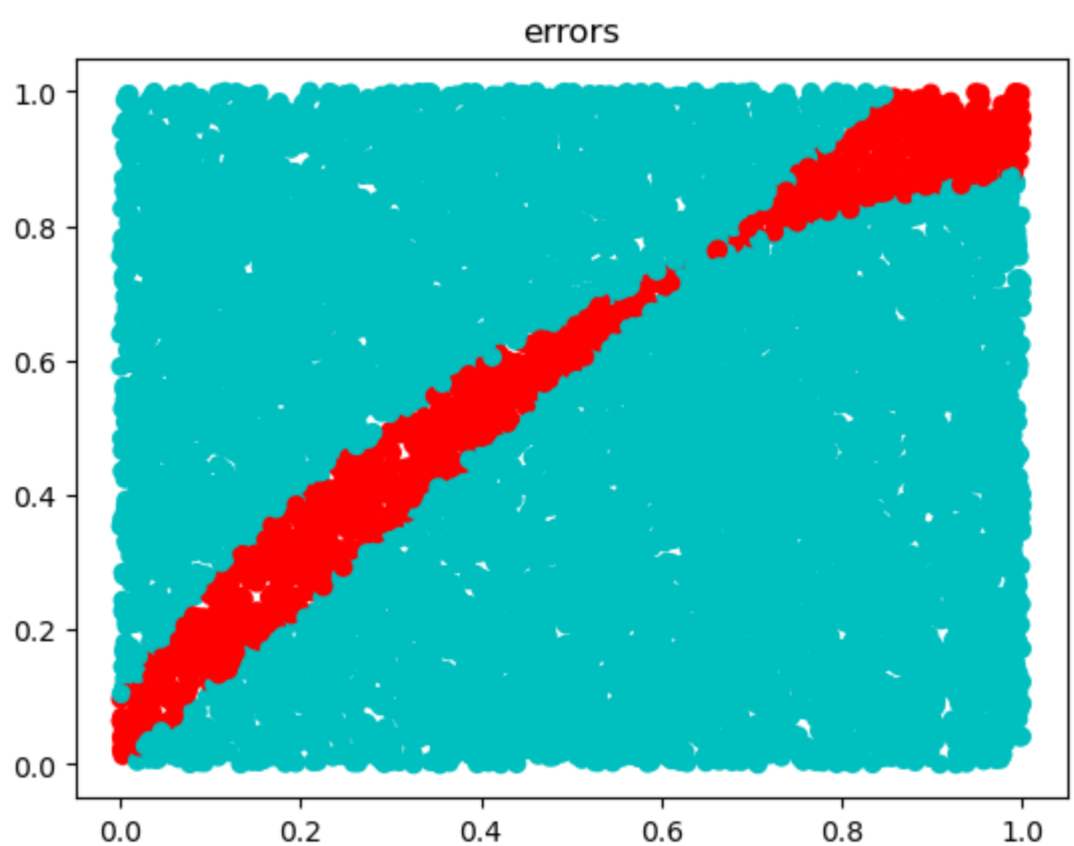
```
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_eval[:,0]])
plt.title('eval data true class')
plt.show()
```



In [3]:
```
## Classifier 1

# w = (X^T X)^(-1)X^T y
w_opt = np.linalg.inv(x_train.transpose()@x_train)@x_train.transpose()@y_train
y_hat = np.sign(x_eval@w_opt)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat[:,0]])
plt.title('eval data predicted class (y_hat)')
plt.show()

# Problem 2a comment:
# The training data appears to be split along a curved decision boundary
# so the case where x^T is [x1 x2], which makes a linear decision boundary,
# has significant errors.

# % error = 1102/10000 = 0.1102 = 11.02%
```



In [4]:
```
error_vec = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat, y_eval))]
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec])
plt.title('errors')
plt.show()

print('Errors: '+ str(sum(error_vec)))
```

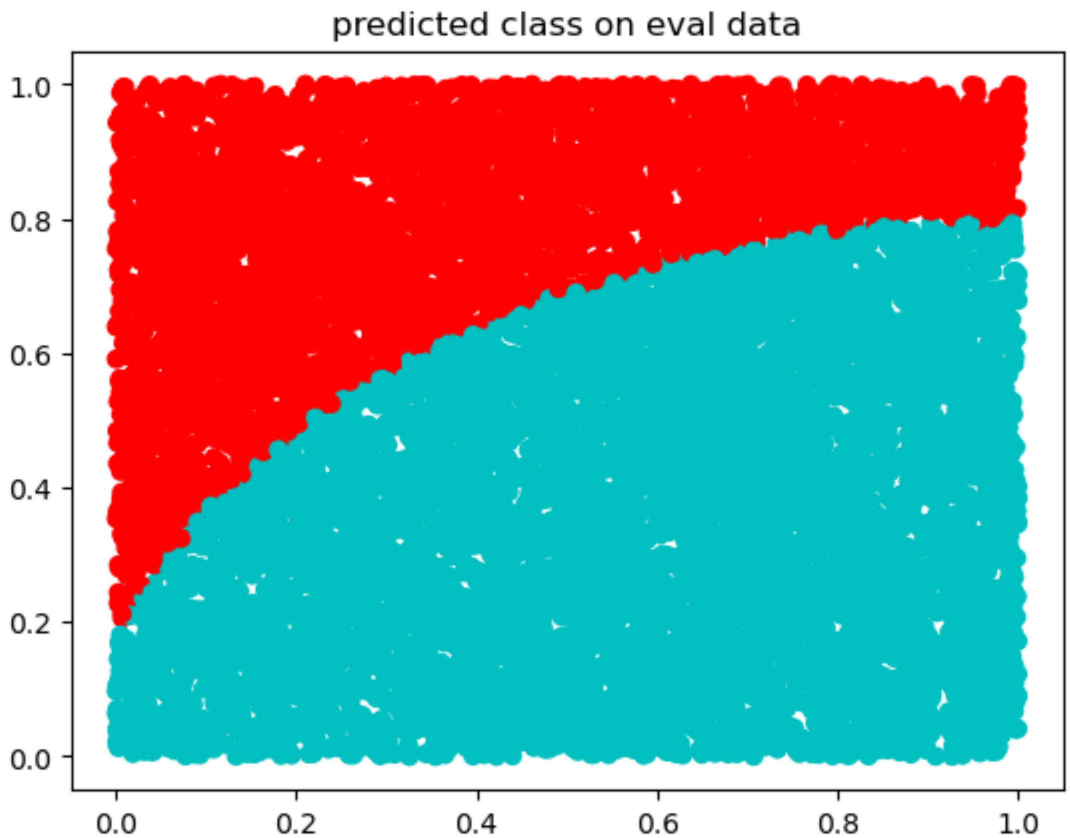



```
Errors: 1102
```

2b)

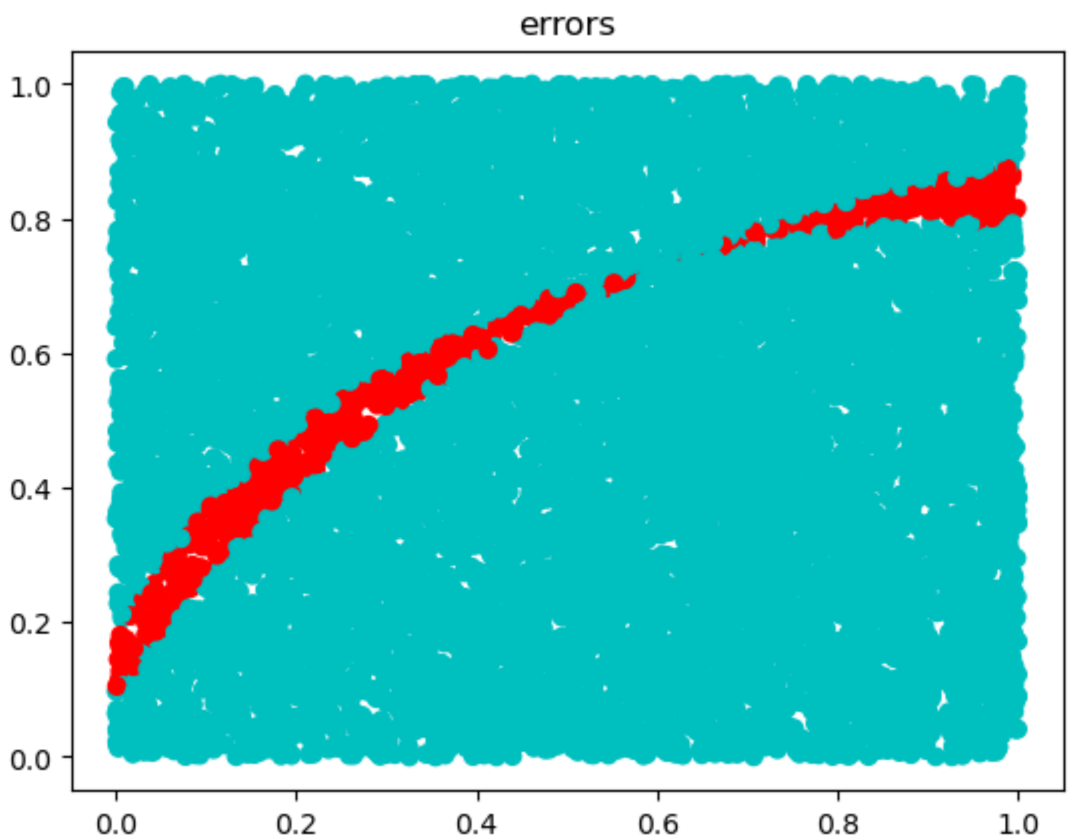
In [5]:

```
## Classifier 2
x_train_2 = np.hstack((x_train**2, x_train, np.ones((n_train,1)) ))
x_eval_2 = np.hstack((x_eval**2, x_eval, np.ones((n_eval,1)) ))

w_opt_2 = np.linalg.inv(x_train_2.transpose()@x_train_2)@x_train_2.transpose()@y_train
y_hat_2 = np.sign(x_eval_2@w_opt_2)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat_2[:,0]]
plt.title('predicted class on eval data')
plt.show()

# Problem 2b comment:
# The curved decision boundary fits the training data much better as
# the training data appears to be split along a curve

# % error = 542/10000 = 0.0542 = 5.42%
```



In [6]:

```
error_vec_2 = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat_2, y_eval))]
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec_2])
plt.title('errors')
plt.show()

print('Error: '+ str(sum(error_vec_2)))
```

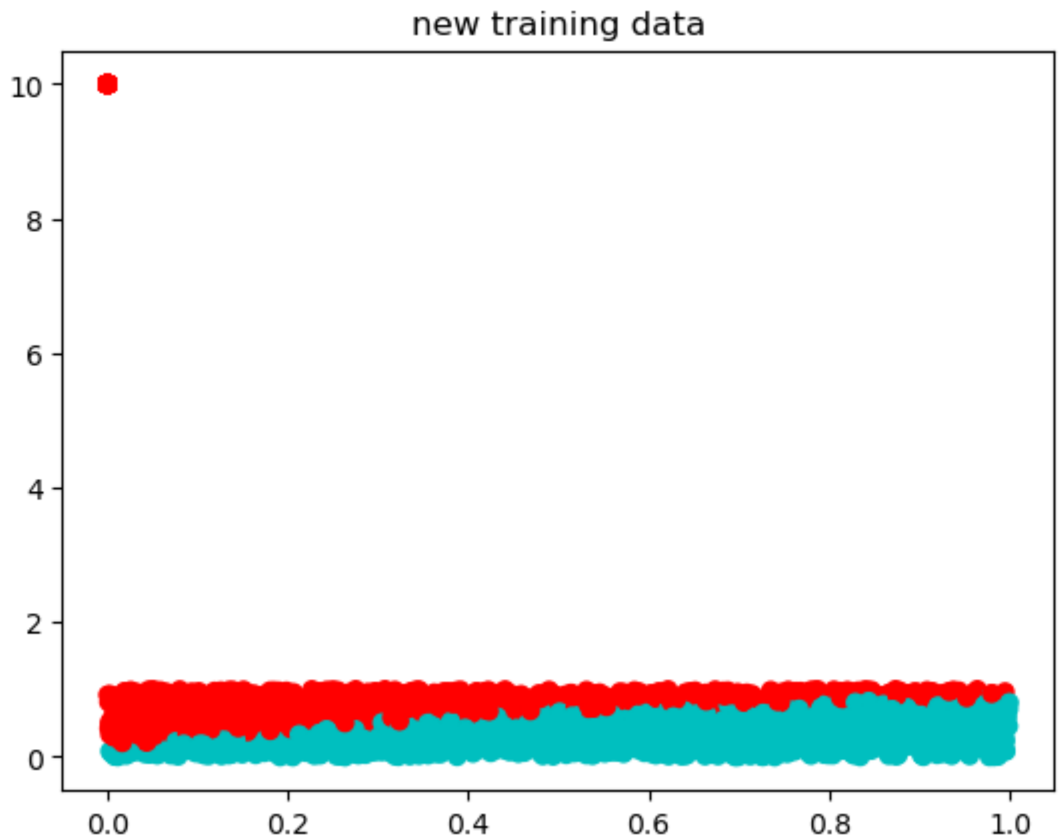



```
Error: 542
```

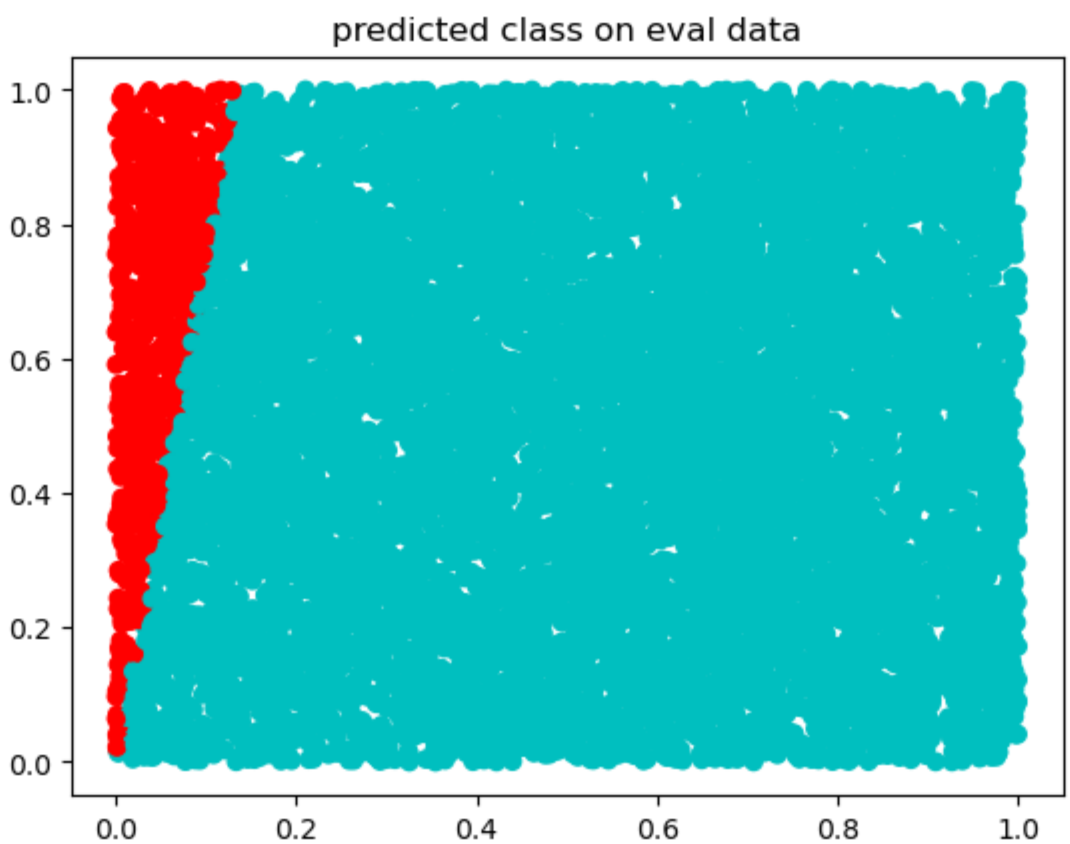
2c)

In [7]:

```
## create new, correctly labeled points
n_new = 1000 #number of new datapoints
x_train_new = np.hstack((np.zeros((n_new,1)), 10*np.ones((n_new,1))))
y_train_new = np.ones((n_new,1))

## add these to the training data
x_train_outlier = np.vstack((x_train,x_train_new))
y_train_outlier = np.vstack((y_train,y_train_new))
plt.scatter(x_train_outlier[:,0],x_train_outlier[:,1], color=['c' if i==-1 else 'r' for
plt.title('new training data')
plt.show()

# Problem 2c comment:
# The decision boundary shifts towards the outliers
# the further away the outliers are, the more it tilts towards them
# and the # of errors increases more and more
#
```

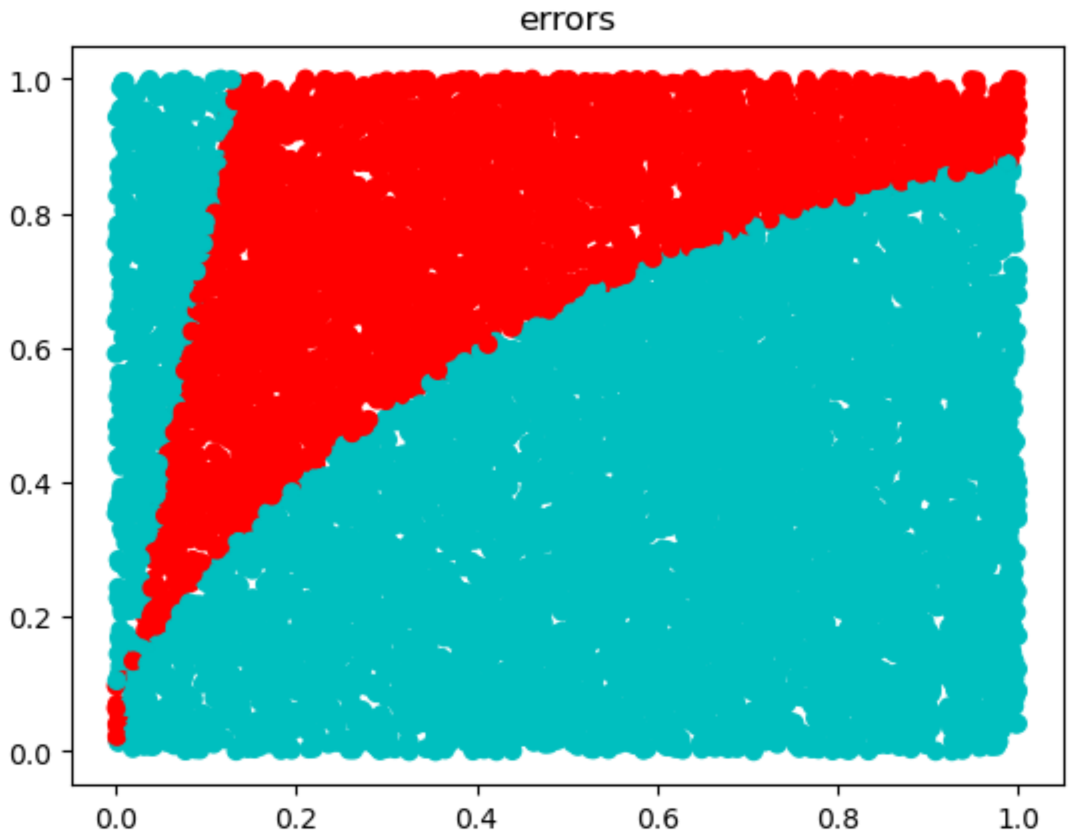


In [8]:
```
#train with new data
w_opt_outlier = np.linalg.inv(x_train_outlier.transpose()@x_train_outlier)@x_train_outli
y_hat_outlier = np.sign(x_eval@w_opt_outlier)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat_outlier
plt.title('predicted class on eval data')
plt.show()
```



In [9]:
```
error_vec = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat_outlier, y_eval))]
```

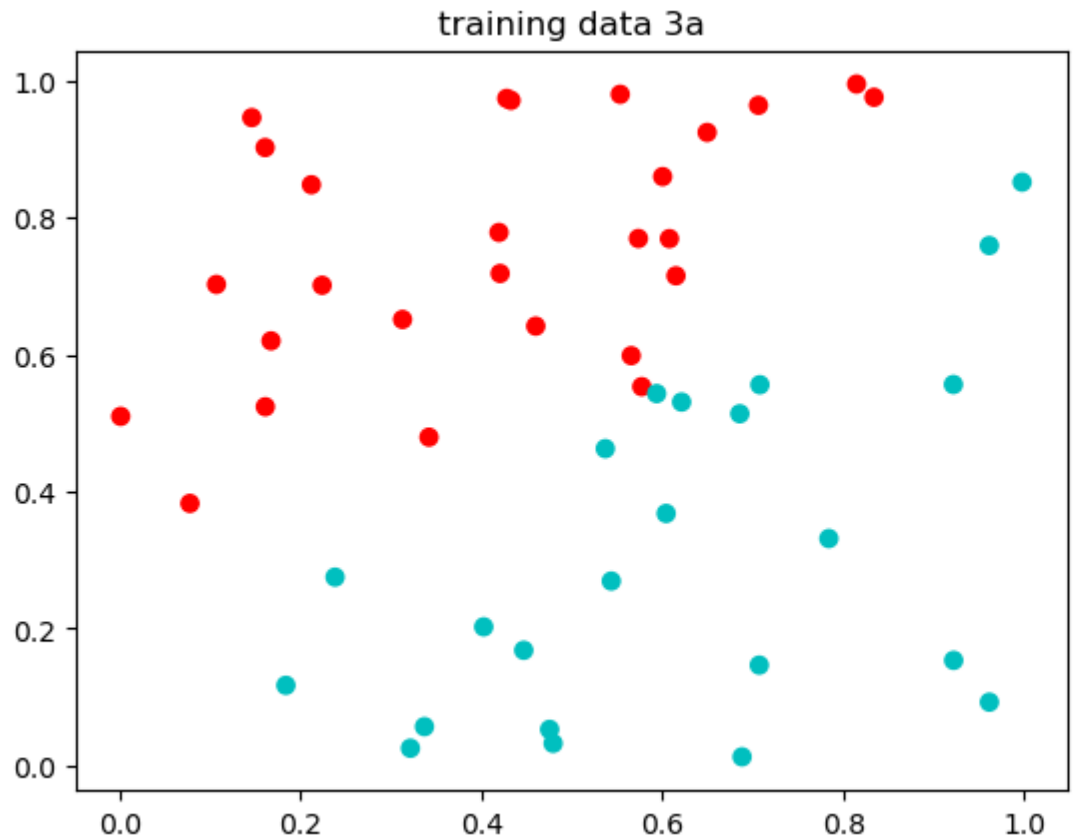
```
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec])
plt.title('errors')
plt.show()

print('Errors: '+ str(sum(error_vec)))
```



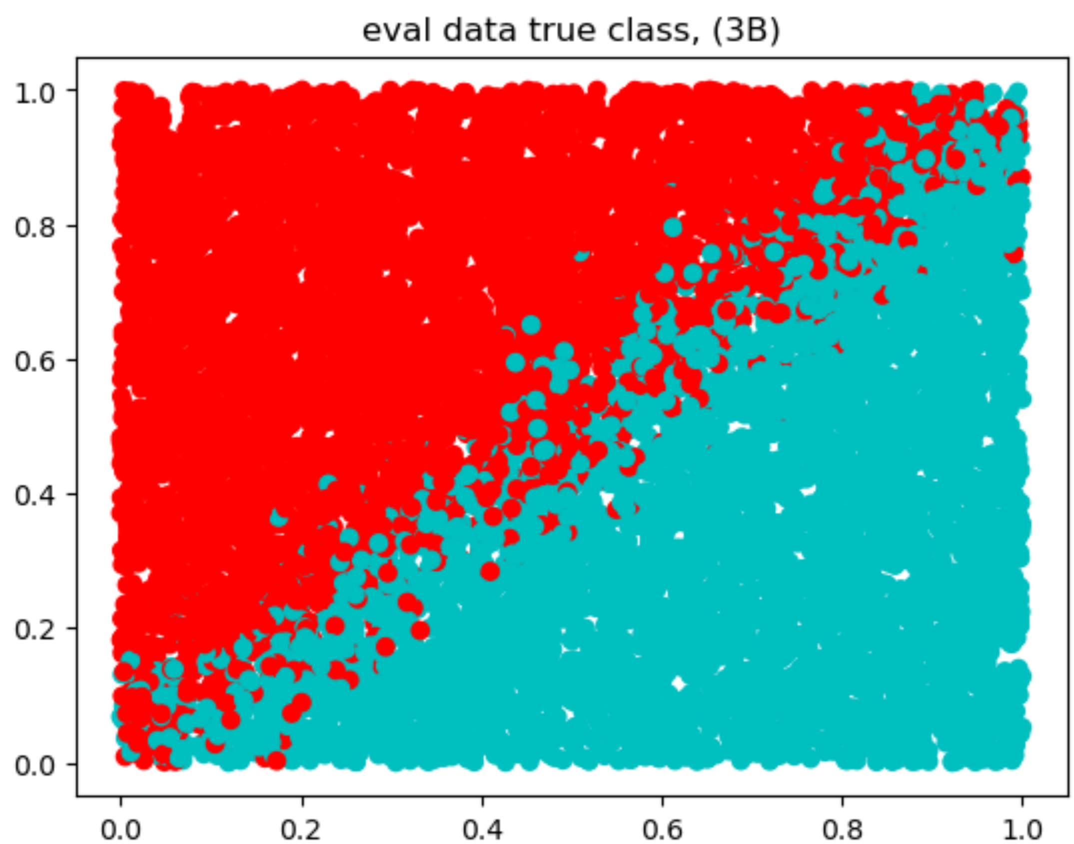

```
Errors: 3277
```

In [10]:

```
### 3a ###

import numpy as np
from scipy.io import loadmat
import matplotlib.pyplot as plt

in_data = loadmat('./overfitting_data.mat')
#print([key for key in in_data]) # -- use this line to see the keys in the dictionary da

x_train = in_data['x_train']
x_eval = in_data['x_eval']
y_train = in_data['y_train']
y_eval = in_data['y_eval']

n_eval = np.size(y_eval)
n_train = np.size(y_train)

plt.scatter(x_train[:,0],x_train[:,1], color=['c' if i==-1 else 'r' for i in y_train[:,0
plt.title('training data 3a')
plt.show()
```

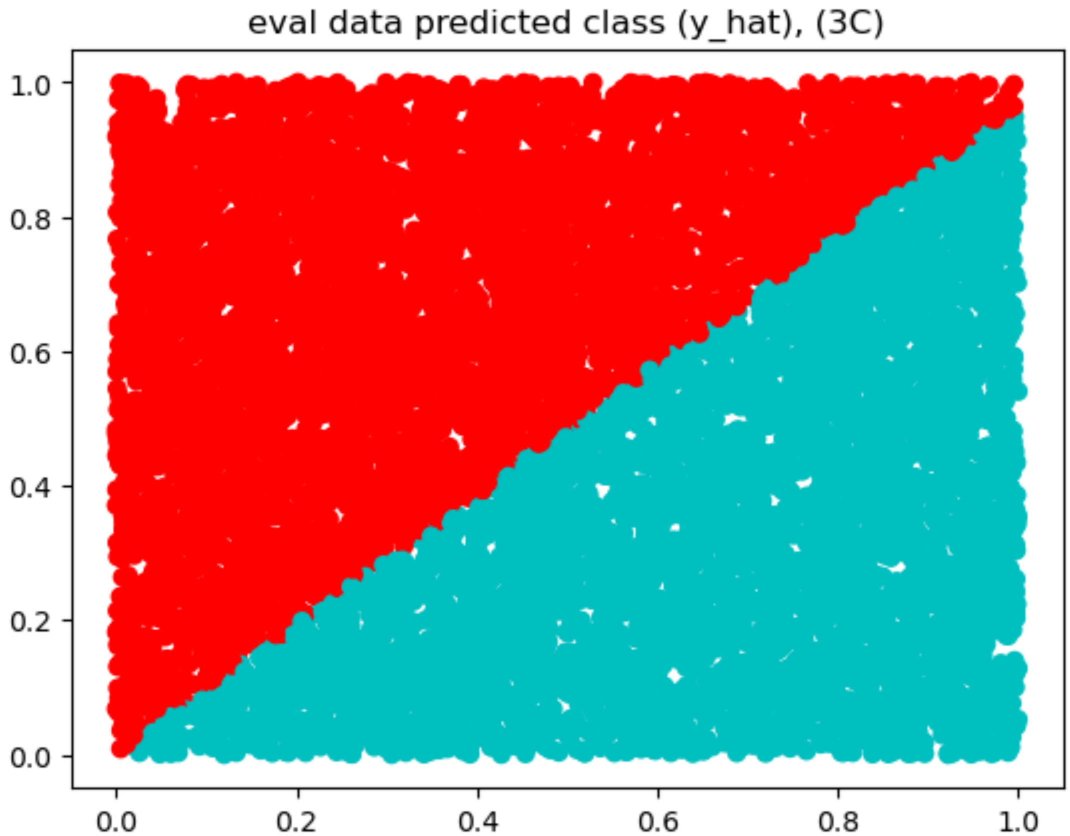


In [11]:
```
### 3b ###
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_eval[:,0]])
plt.title('eval data true class, (3B)')
plt.show()
```

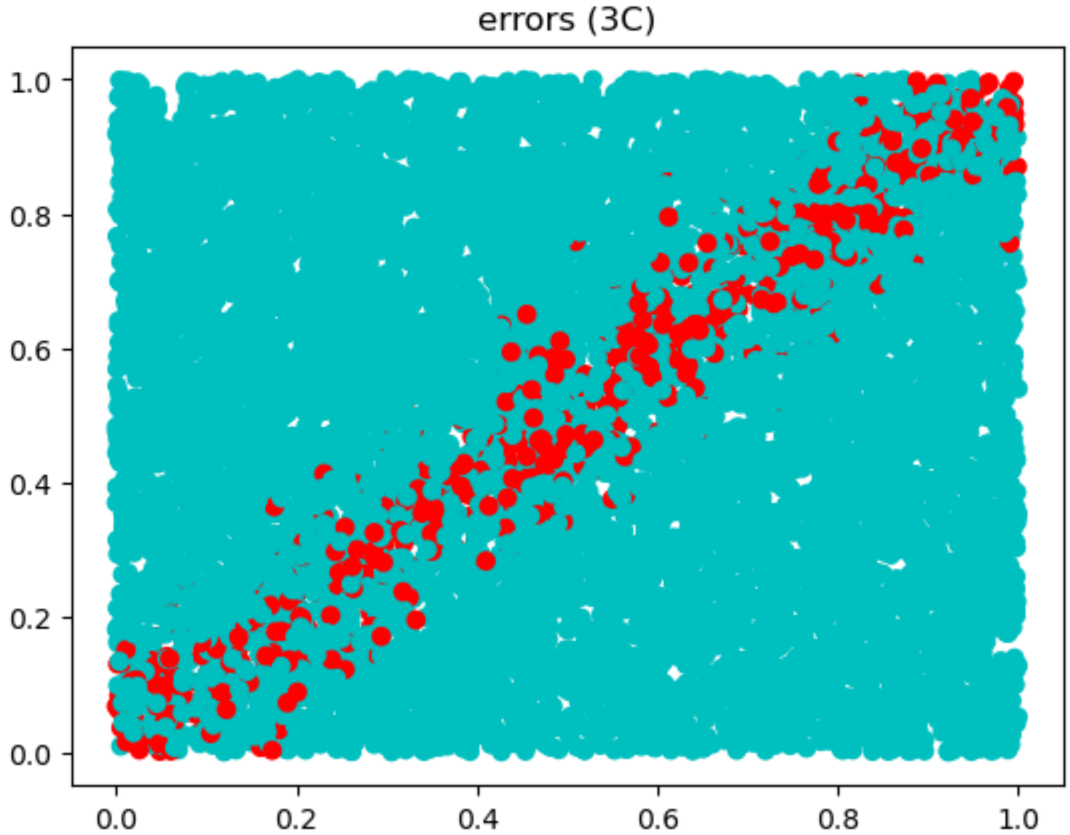


In [12]:
```
### 3c ###
## Classifier 1

# w = (X^T X)^(-1)X^T y
```

```
w_opt = np.linalg.inv(x_train.transpose()@x_train)@x_train.transpose()@y_train
y_hat = np.sign(x_eval@w_opt)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat[:,0]])
plt.title('eval data predicted class (y_hat), (3C)')
plt.show()
```

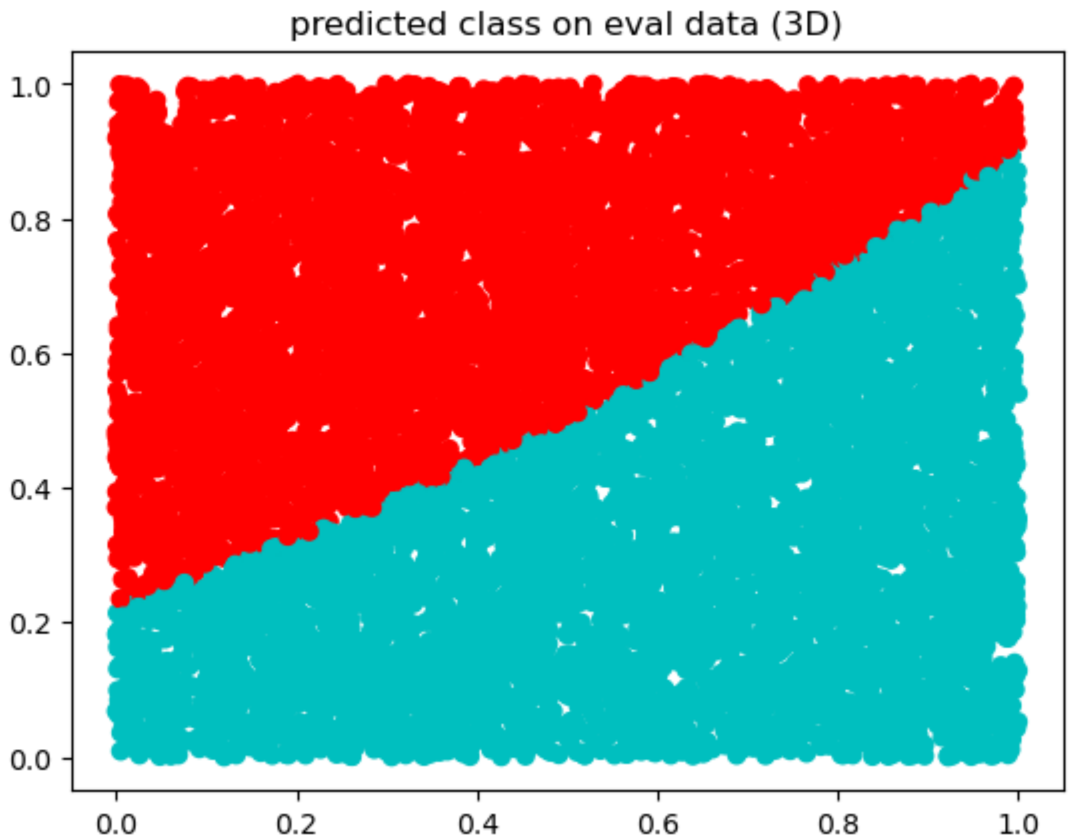


In [13]:
```
error_vec = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat, y_eval))]
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec])
plt.title('errors (3C)')
plt.show()

print('Errors: '+ str(sum(error_vec)))
print('Total Samples = ' ,x_eval.shape[0])
print("Percentage error = ", sum(error_vec)/x_eval.shape[0])
```

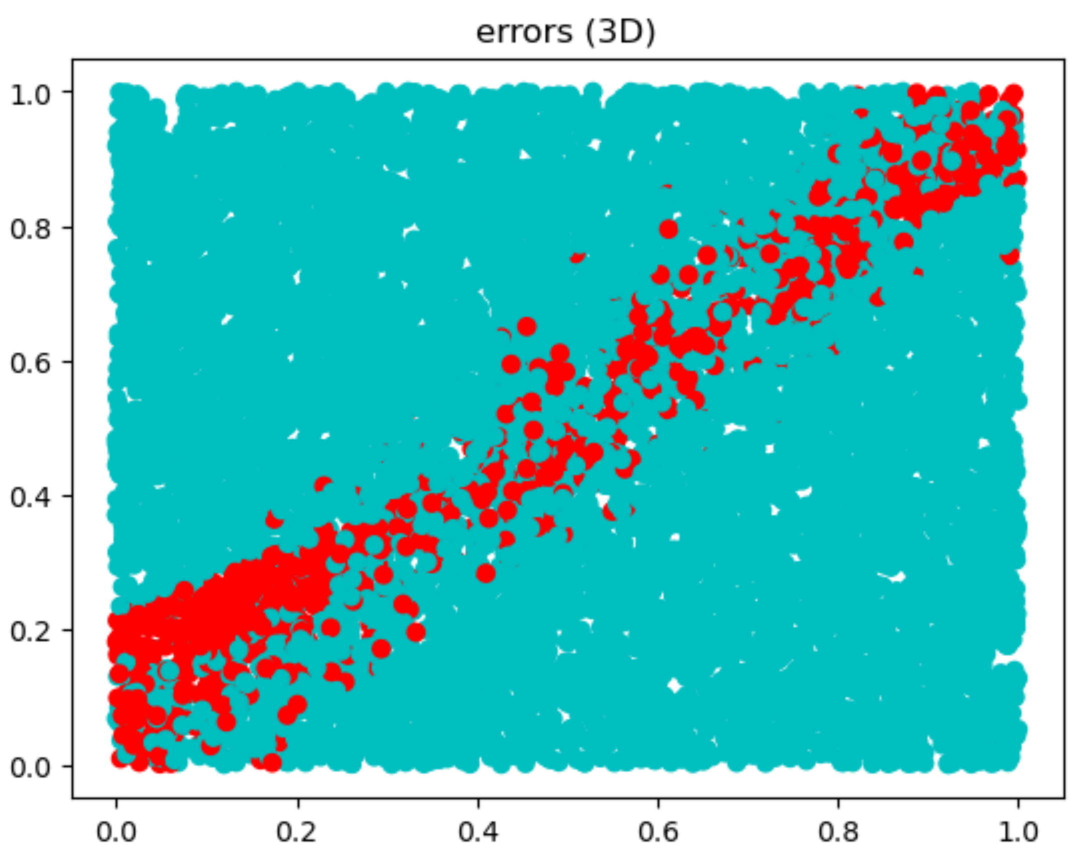



```
Errors: 759
Total Samples =  10000
Percentage error =  0.0759
```

In [14]:
```
### 3d ###

## Classifier 2
x_train_2 = np.hstack((x_train**2, x_train, np.ones((n_train,1)) ))
x_eval_2 = np.hstack((x_eval**2, x_eval, np.ones((n_eval,1)) ))

w_opt_2 = np.linalg.inv(x_train_2.transpose()@x_train_2)@x_train_2.transpose()@y_train
y_hat_2 = np.sign(x_eval_2@w_opt_2)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat_2[:,0]]
plt.title('predicted class on eval data (3D)')
plt.show()
```



In [15]:
```
error_vec_2 = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat_2, y_eval))]
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec_2])
plt.title('errors (3D)')
plt.show()

print('Error: '+ str(sum(error_vec_2)))
```

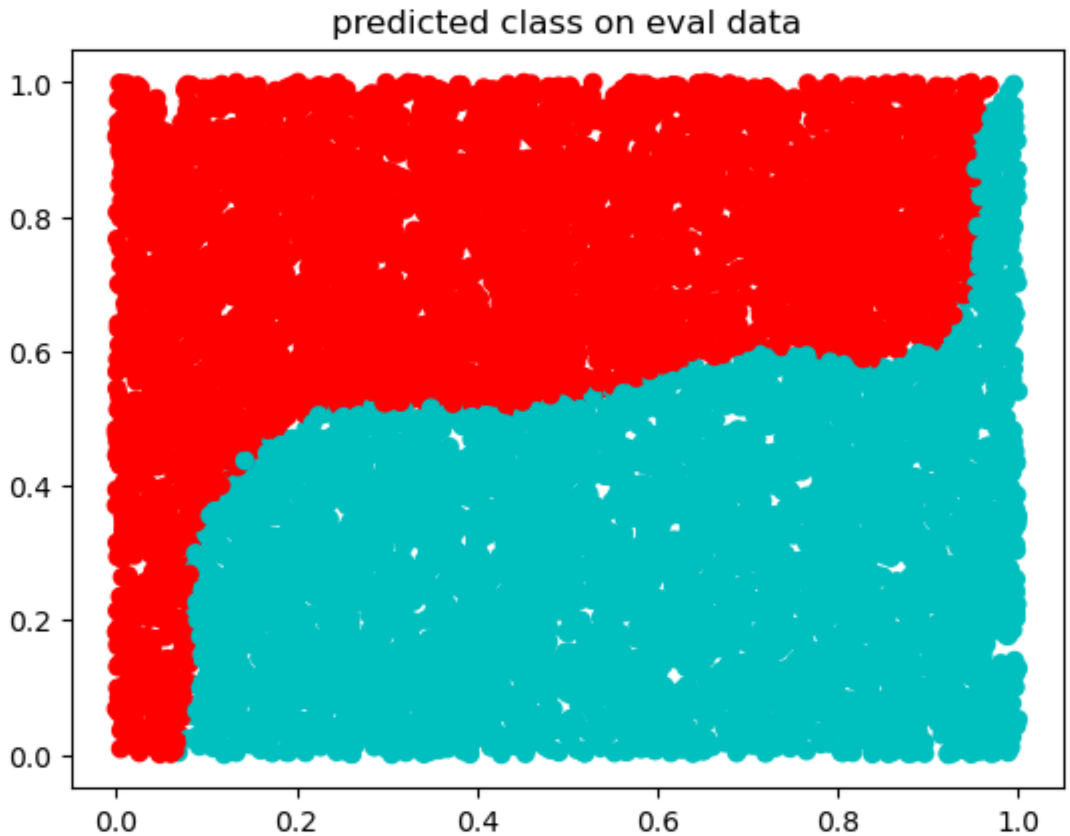



```
Error: 1066
```

In [16]:
```
### 3e ###
```

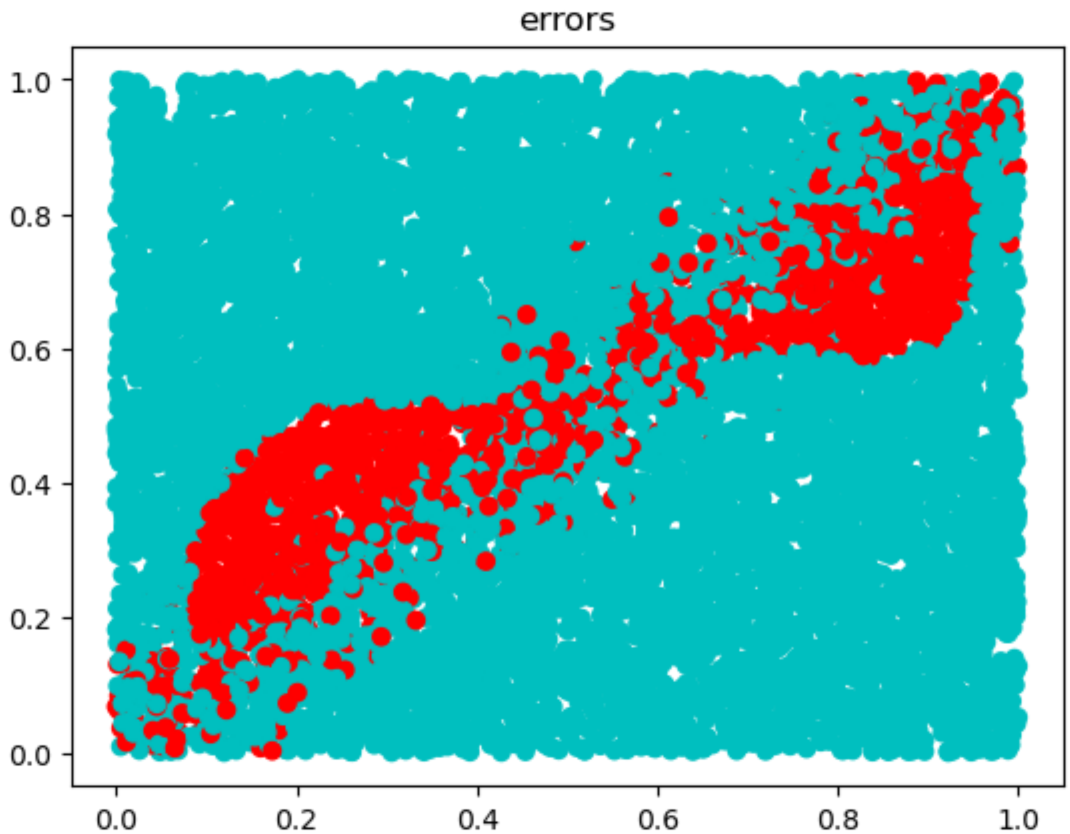
```
## Classifier 3
x_train_3 = np.hstack((x_train**6, x_train**5, x_train**4, x_train**3, x_train**2, x_tra
x_eval_3 = np.hstack((x_eval**6, x_eval**5, x_eval**4, x_eval**3, x_eval**2, x_eval, np.

w_opt_3 = np.linalg.inv(x_train_3.transpose()@x_train_3)@x_train_3.transpose()@y_train
print(w_opt_3.shape, x_eval_3.shape)
y_hat_3 = np.sign(x_eval_3@w_opt_3)

plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==-1 else 'r' for i in y_hat_3[:,0]]
plt.title('predicted class on eval data')
plt.show()
```

```
(13, 1) (10000, 13)
```



In [19]:

```
error_vec_3 = [0 if i[0]==i[1] else 1 for i in np.hstack((y_hat_3, y_eval))]
plt.scatter(x_eval[:,0],x_eval[:,1], color=['c' if i==0 else 'r' for i in error_vec_3])
plt.title('errors')
plt.show()

print('Error: '+ str(sum(error_vec_3)))
```




```
Error: 1677
```

In [ ]:
```
### 3f ###
# The highest order classifier 3 performed the worst because it overfits
# to noise in the small training sample set
```

4. A binary linear classifier based on three features x_1, x_2, and x_3 is $\hat{y} = \text{sign}\{\boldsymbol{x}^T \boldsymbol{w}\}$ where $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$. Hence the decision boundary is the set $\{x_1, x_2, x_3\}$ for which $\boldsymbol{x}^T \boldsymbol{w} = 0$.

The decision boundary for a two-dimensional classifier is a line. What type of geometric object is the decision boundary in three dimensions?

5. A decision boundary for a classification problem involving features x_1, x_2, and x_3 is defined as $\boldsymbol{x}^T \boldsymbol{w} = 0$ where $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 & 1 \end{bmatrix}$. Find $\boldsymbol{w}$ so that the decision boundary is parallel to the x_1-x_2 plane and includes the point $(x_1, x_2, x_3) = (0, 0, 1)$.

4.) $\hat{y} = \text{sign}\{\underline{x}^T \underline{w}\}$ $\quad \underline{x}^T = [x_1 \; x_2 \; x_3]$

Set $\{x_1, x_2, x_3\}$ so $\underline{x}^T \underline{w} = 0$

$$[x_1 \; x_2 \; x_3] \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = x_1 w_1 + x_2 w_2 + x_3 w_3 = 0.$$

$\rightarrow x_3 w_3 = -x_1 w_1 - x_2 w_2$

$\rightarrow x_3 = -{w_1}/{w_3} \, x_1 - {w_2}/{w_3} \, x_2$.

$\rightarrow$ x_3 is a 2-d plane of x_1 and x_2.

5. A decision boundary for a classification problem involving features x_1, x_2, and x_3 is defined as $\boldsymbol{x}^T\boldsymbol{w} = 0$ where $\boldsymbol{x}^T = \begin{bmatrix} x_1 & x_2 & x_3 & 1 \end{bmatrix}$. Find $\boldsymbol{w}$ so that the decision boundary is parallel to the x_1-x_2 plane and includes the point $(x_1, x_2, x_3) = (0, 0, 1)$.

$$\underline{x}^T \underline{w} = [x_1 \ x_2 \ x_3 \ 1] \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = 0$$

$$\Rightarrow x_1 w_1 + x_2 w_2 + x_3 w_3 + w_4 = 0$$ decision boundary

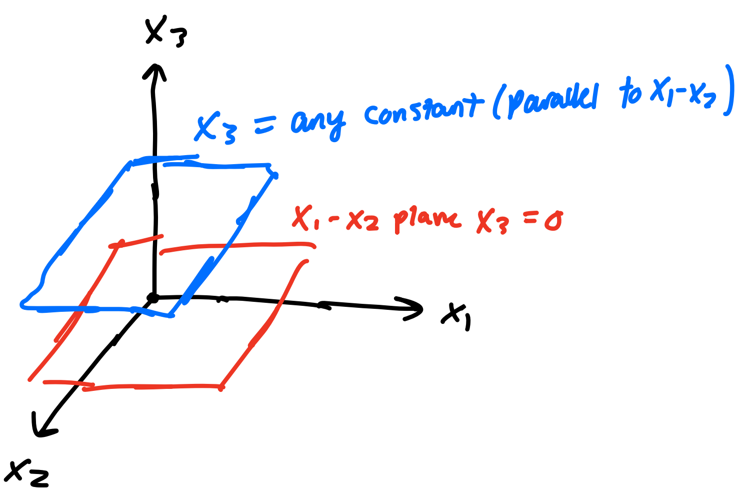
$$\Rightarrow \quad x_3 = -w_1/w_3 \, x_1 - w_2/w_3 \, x_2 - w_4/w_3$$

1.) $0w_1 + 0w_2 + 1w_3 + w_4 = 0$

$\rightarrow$ $w_3 + w_4 = 0$ & $-w_3/w_4 = 1$

2.) $\parallel$ to $x_1 - x_2$ plane:

$\Rightarrow w_1 = w_2 = 0.$



$$\rightarrow \quad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

Example point.

$(-2, 6, 1)$

$$\rightarrow [-2 \quad 6 \quad 1 \quad 1] \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

$$= 0 + 0 + 1 - 1 = 0$$

on decision boundary.