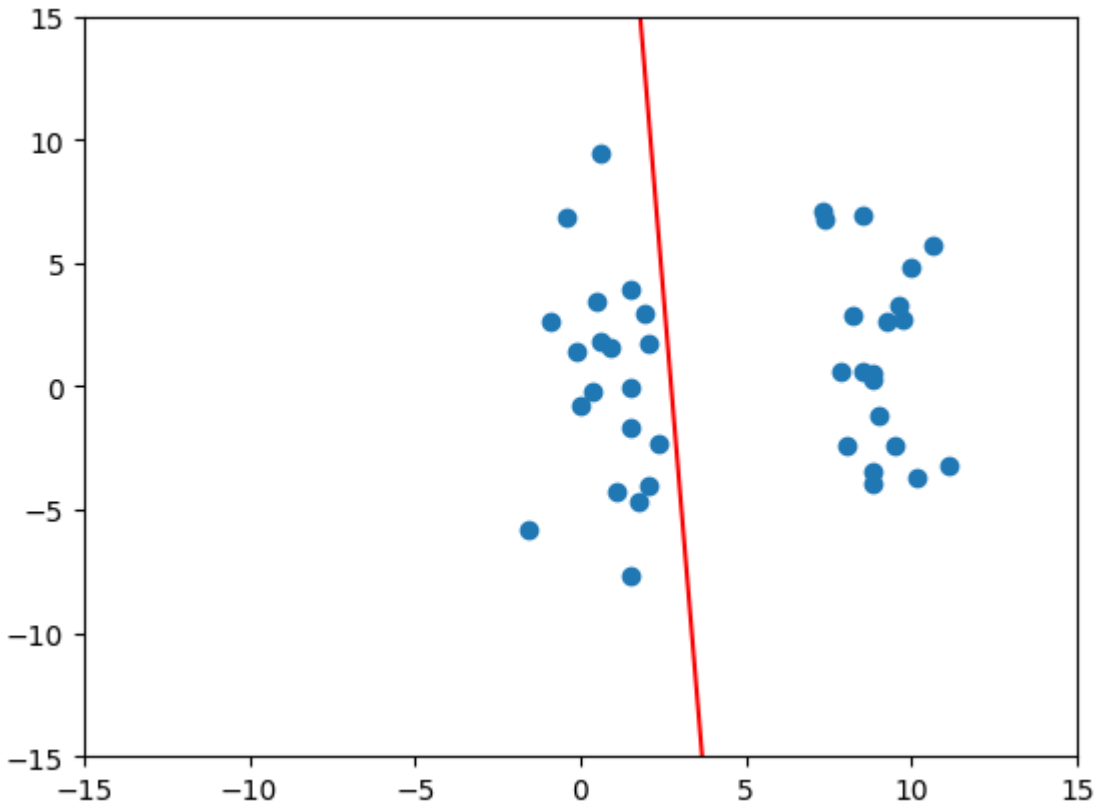
In [1]:

```
import numpy as np
import os.path
import matplotlib.pyplot as plt

class PerceptronC:
    def __init__(self, num_iter):
        self.num_iteration = num_iter
        self.weights = None
        self.predicted = None

    def step_function(self, sum):
        if sum >= 0:
            return 1
        return 0

    def fit(self, train_x, train_y):
        total_size, features = train_x.shape
        self.weights = np.random.rand(features)
        for num in range(self.num_iteration):
            for i, test in enumerate(train_x):
                output = self.step_function(np.dot(self.weights, test))
                if output != train_y[i]:
                    if output == 1:
                        self.weights = np.subtract(self.weights, test)
                    else:
                        self.weights = np.add(self.weights, test)

    def predict(self, test_x):
        total_size, features = test_x.shape
        self.predicted = np.zeros((total_size,1))
        for i, test in enumerate(test_x):
            self.predicted[i] = self.step_function(np.dot(self.weights, test
        return self.predicted

    def misclassification_error(self, test_y):
        total = 0
        for i, actual in enumerate(test_y):
            if self.predicted[i] != actual:
                total = total + 1
        return 100 - (total / len(test_y)) * 100

    def plot2D(self, train_x1, train_x2):
        plt.scatter(train_x1,train_x2)
        x = np.linspace(-10, 10, 100)
        b = self.weights[2]
        w1 = self.weights[0]
        w2 = self.weights[1]
        y = (-(b / w2) / (b / w1))*x + (-b / w2)
        plt.xlim(-15, 15)
        plt.ylim(-15, 15)
        plt.plot(x, y, '-r')
        plt.show()

def load_data(fname, directory='Data'):
```

```
    data = np.loadtxt(os.path.join(directory, fname))
    rows, cols = data.shape
    X_dim = cols - 1
    Y_dim = 1
    return data[:, :-1].reshape(-1, X_dim), data[:, -1].reshape(-1, Y_dim)
```

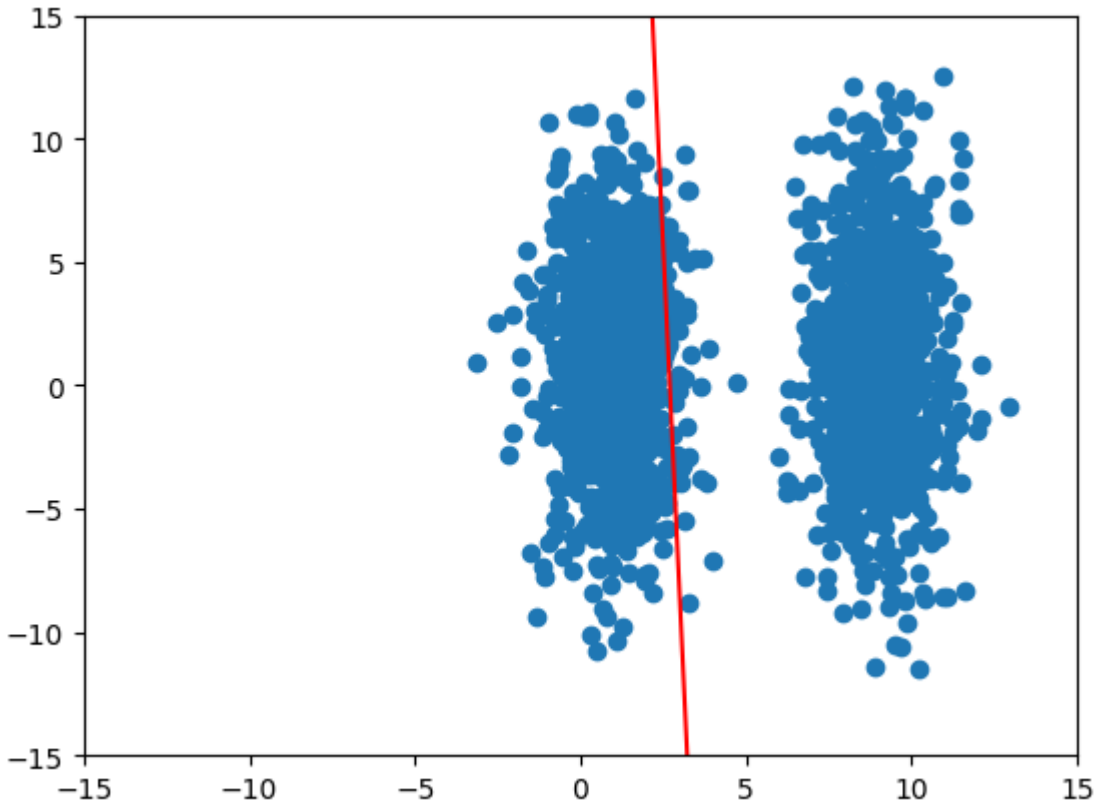
In [3]:
```
# Train data classifier for model 1
model1 = PerceptronC(20)
set_train_x, set_train_y = load_data("set1.train")
model1.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model1.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy", model1.misclassification_error(set_train_y))
model1.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

```
accuracy 100.0
```



In [172…
```
# model 1 corresponding to set 1 and its prediction on test data
test_x, test_y = load_data("set.test")
model1.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model1.misclassification_error(test_y))
model1.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

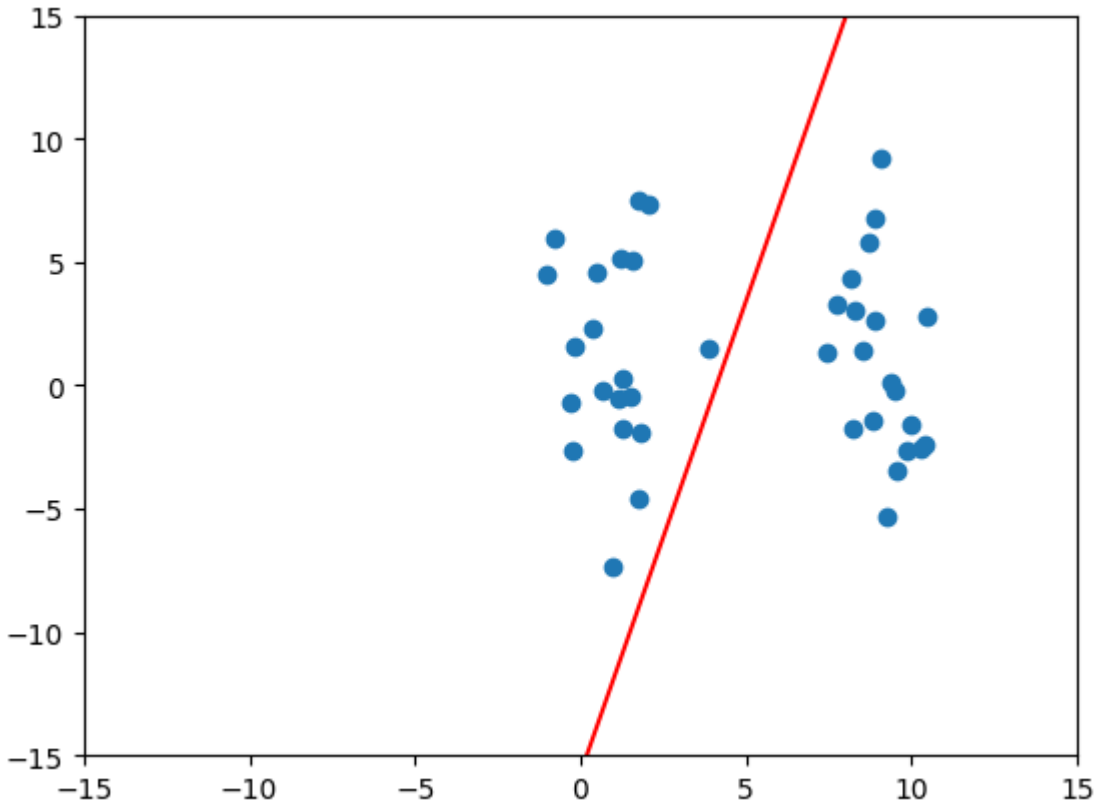
```
accuracy 97.55
```



In [144…

```
# Train data classifier for model 2
model2 = PerceptronC(20)
set_train_x, set_train_y = load_data("set2.train")
model2.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model2.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model2.misclassification_error(set_train_y))
model2.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

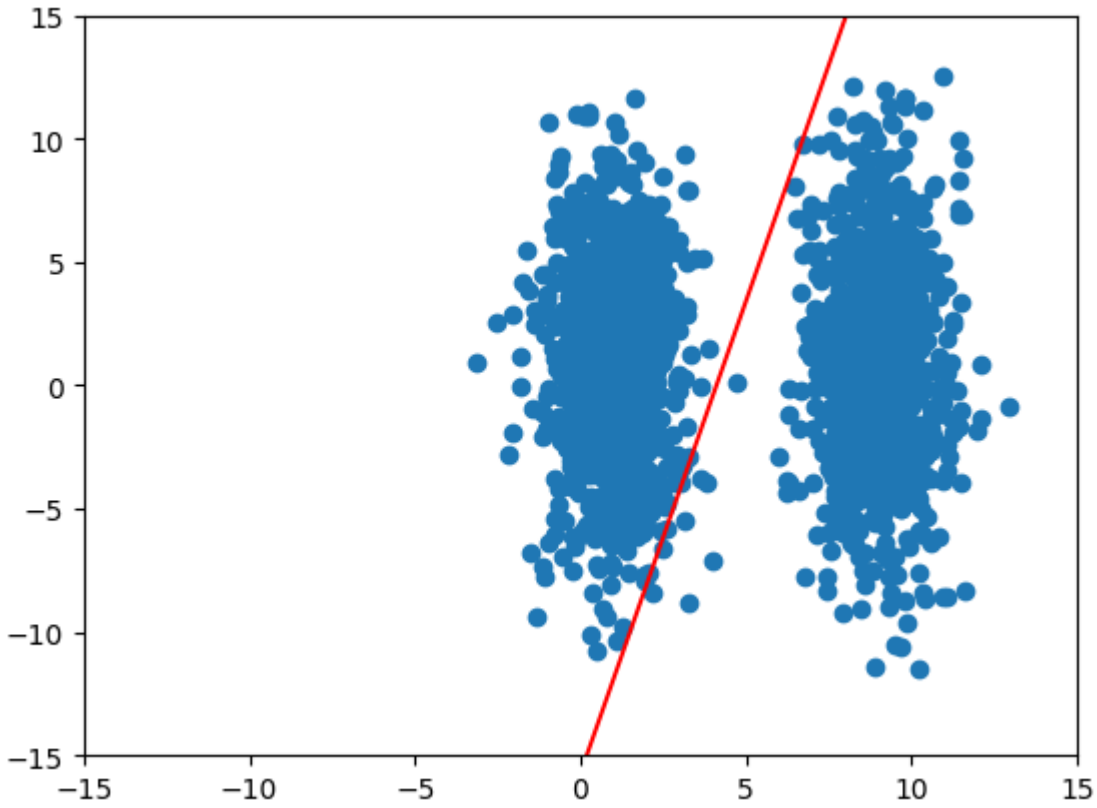
```
accuracy 100.0
```



In [145…

```
# model 2 corresponding to set 2 and its prediction on test data
test_x, test_y = load_data("set.test")
model2.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy", model2.misclassification_error(test_y))
model2.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

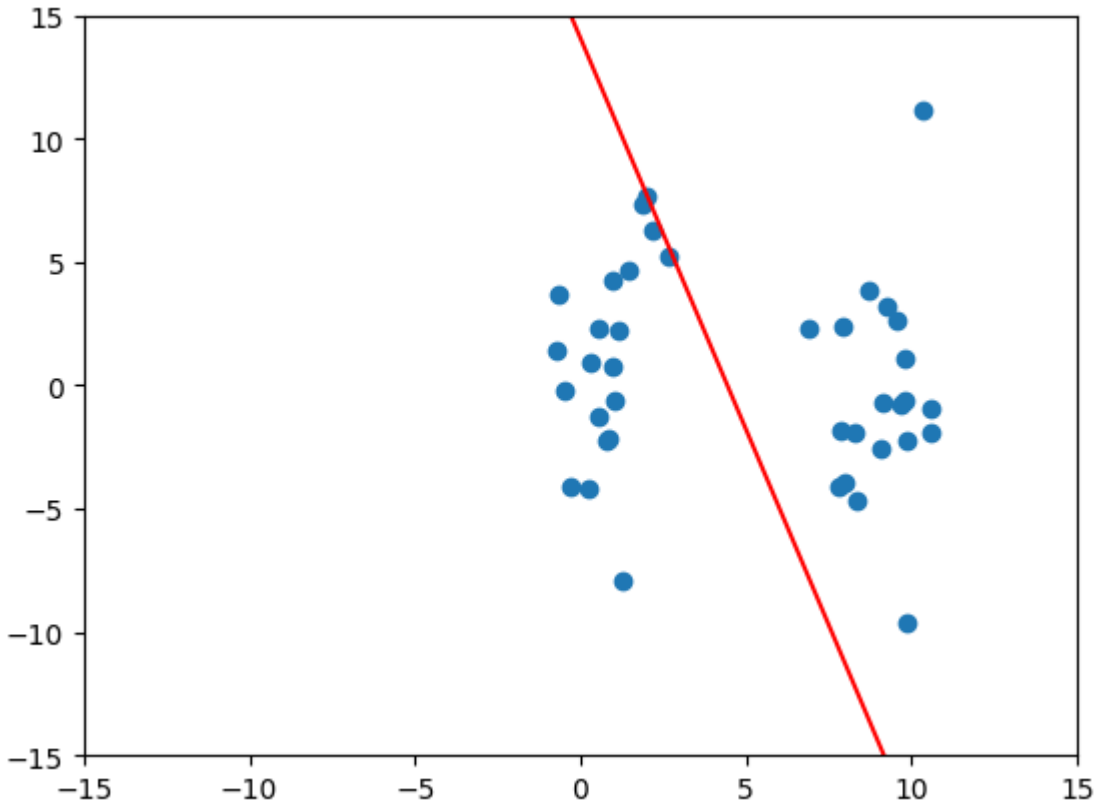
```
accuracy 99.55
```



In [173…

```
# Train data classifier for model 3
model3 = PerceptronC(20)
set_train_x, set_train_y = load_data("set3.train")
model3.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model3.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model3.misclassification_error(set_train_y))
model3.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

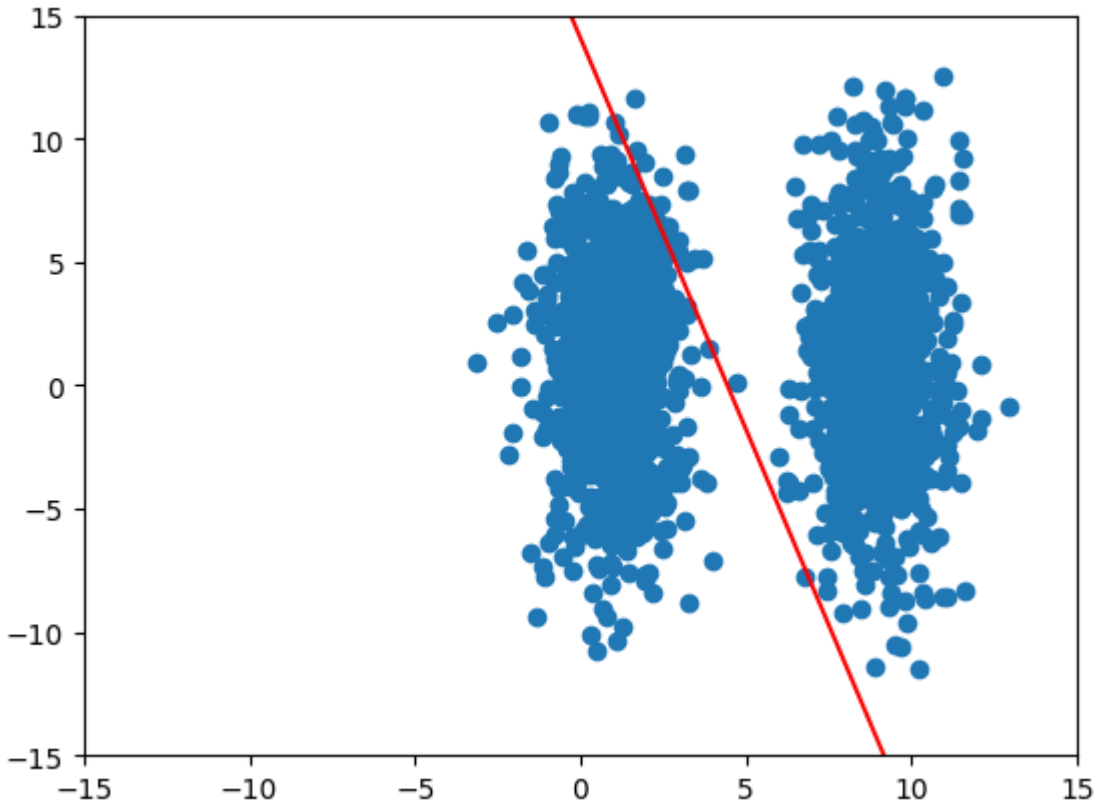
```
accuracy 100.0
```



In [174…

```
# model 3 corresponding to set 3 and its prediction on test data
test_x, test_y = load_data("set.test")
model3.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model3.misclassification_error(test_y))
model3.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

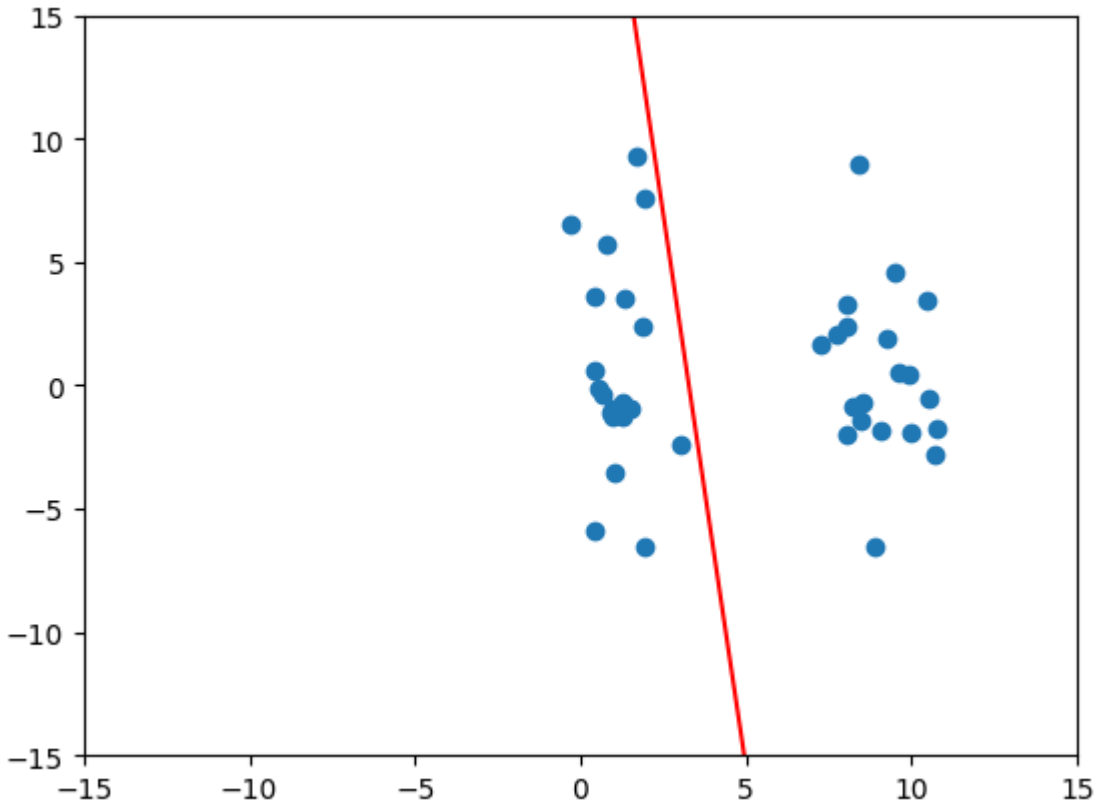
```
accuracy 98.95
```



In [179…

```
# Train data classifier for model 4
model4 = PerceptronC(20)
set_train_x, set_train_y = load_data("set4.train")
model4.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model4.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model4.misclassification_error(set_train_y))
model4.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

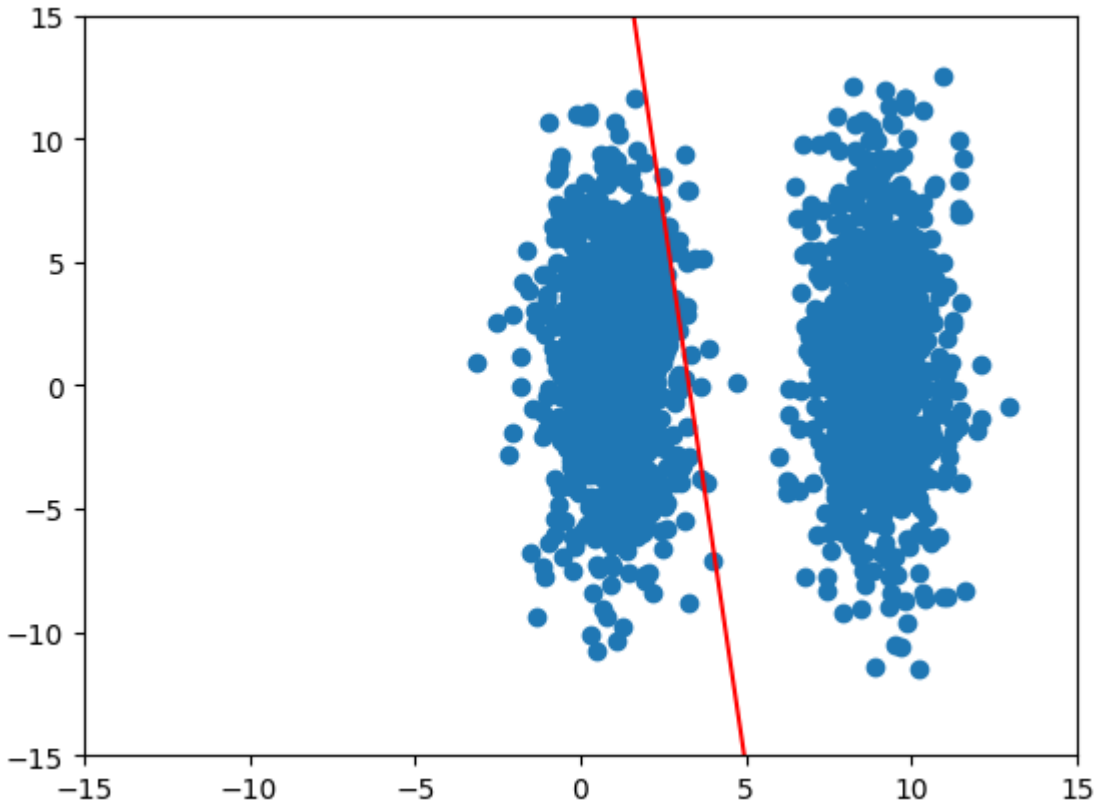
```
accuracy 100.0
```



In [180…

```
# model 4 corresponding to set 4 and its prediction on test data
test_x, test_y = load_data("set.test")
model4.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model4.misclassification_error(test_y))
model4.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

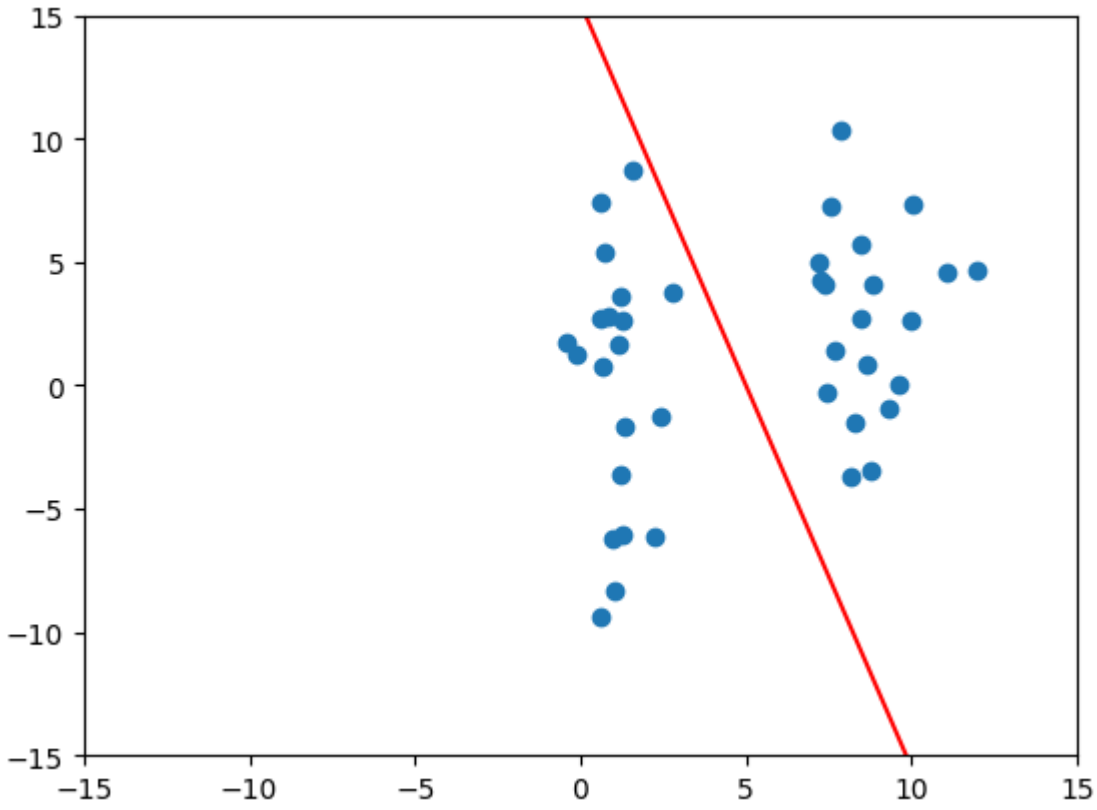
```
accuracy 98.9
```



In [182…

```
# Train data classifier for model 5
model5 = PerceptronC(20)
set_train_x, set_train_y = load_data("set5.train")
model5.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model5.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model5.misclassification_error(set_train_y))
model5.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

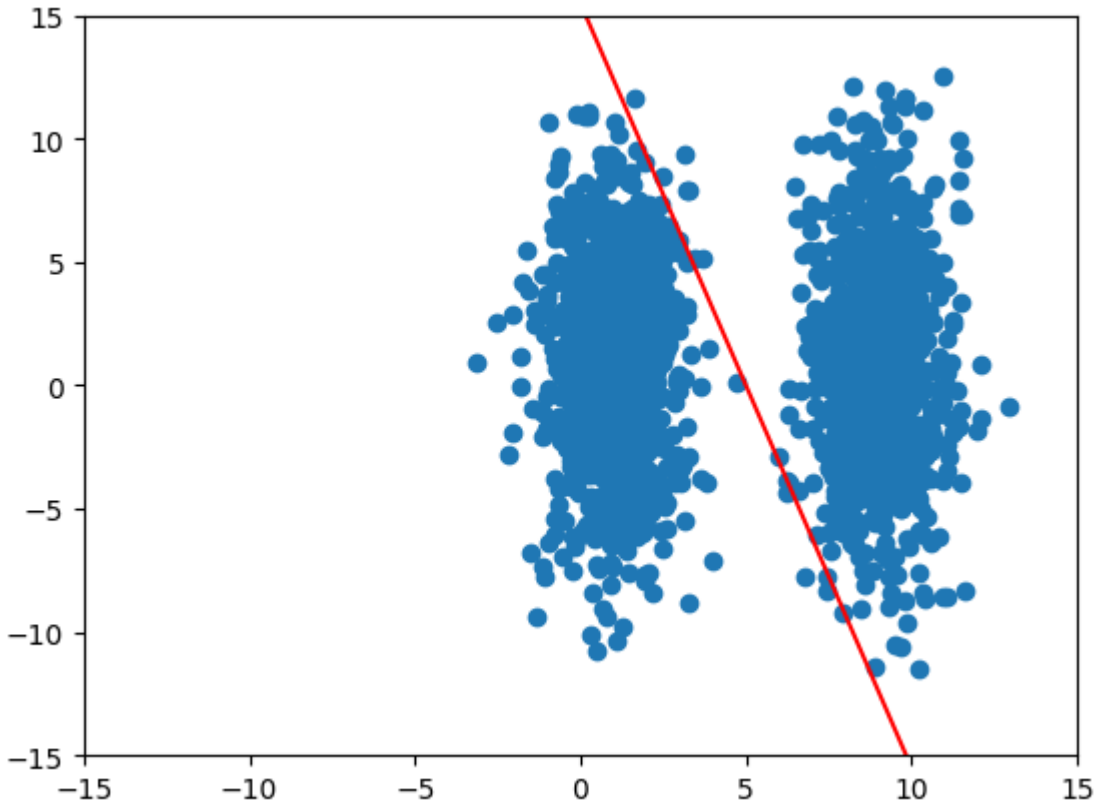
```
accuracy 100.0
```



In [183…

```
# model 5 corresponding to set 5 and its prediction on test data
test_x, test_y = load_data("set.test")
model5.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model5.misclassification_error(test_y))
model5.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

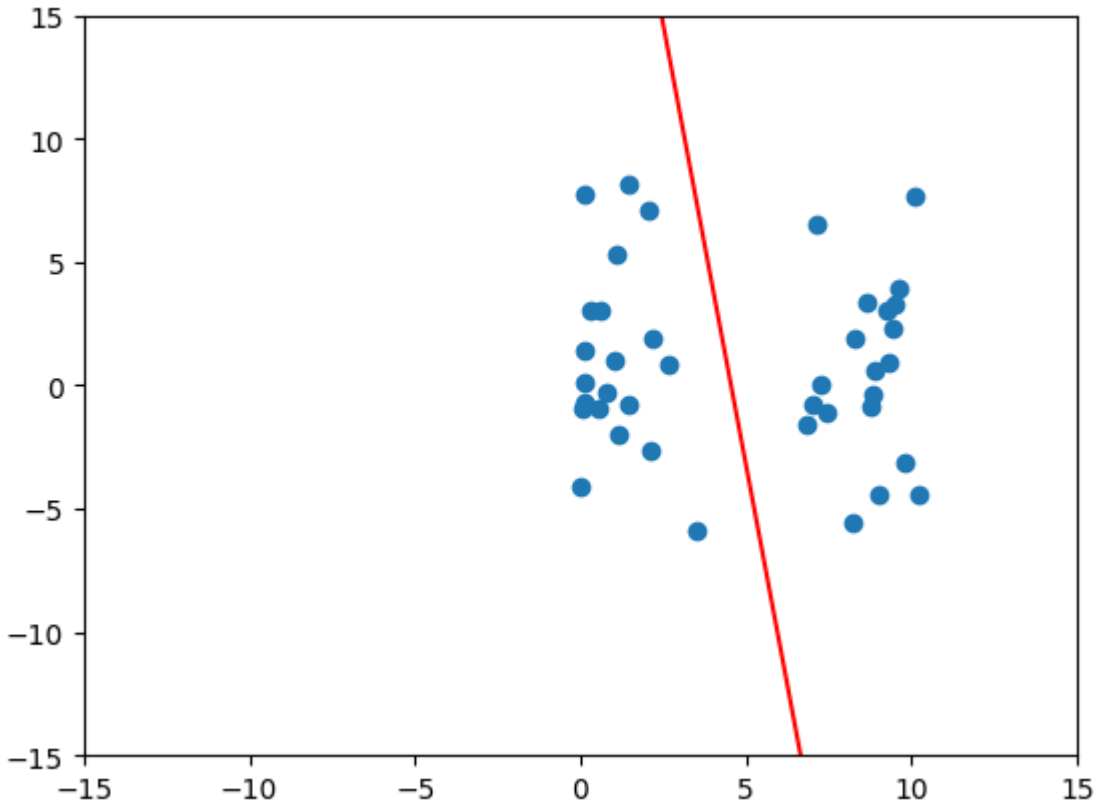
```
accuracy 99.35
```



In [184…

```
# Train data classifier for model 6
model6 = PerceptronC(20)
set_train_x, set_train_y = load_data("set6.train")
model6.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model6.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model6.misclassification_error(set_train_y))
model6.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

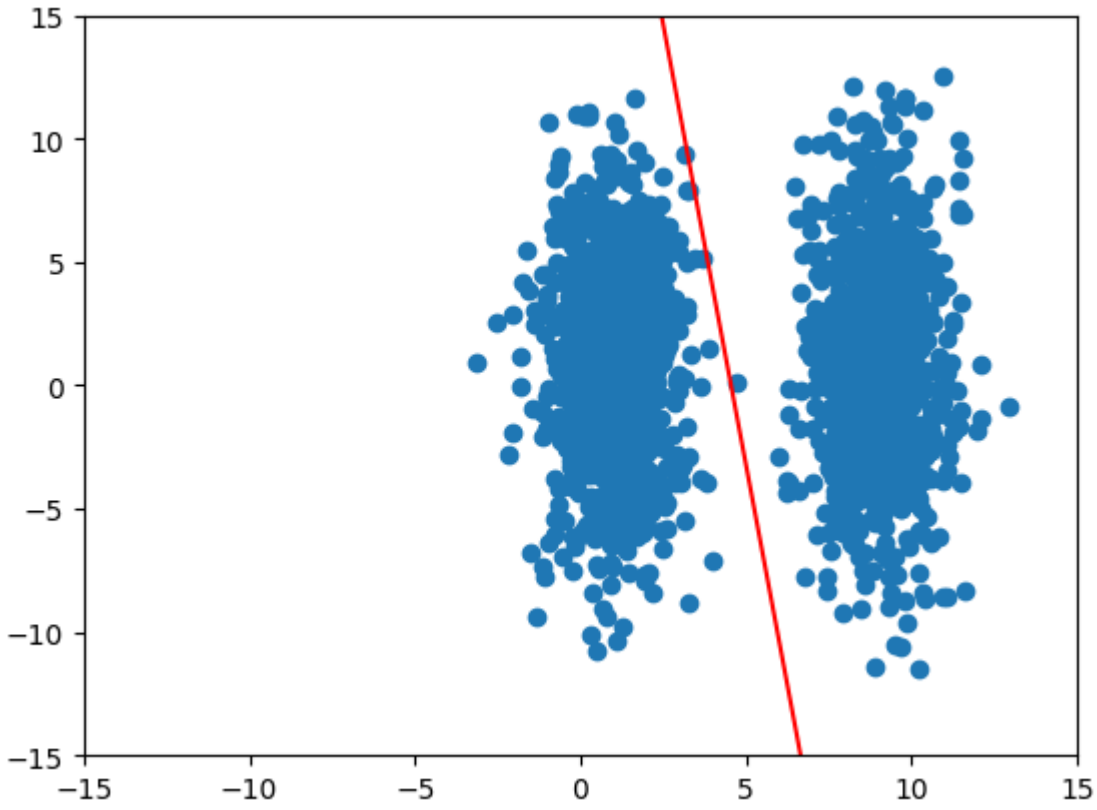
```
accuracy 100.0
```



In [185…

```
# model 6 corresponding to set 6 and its prediction on test data
test_x, test_y = load_data("set.test")
model6.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model6.misclassification_error(test_y))
model6.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

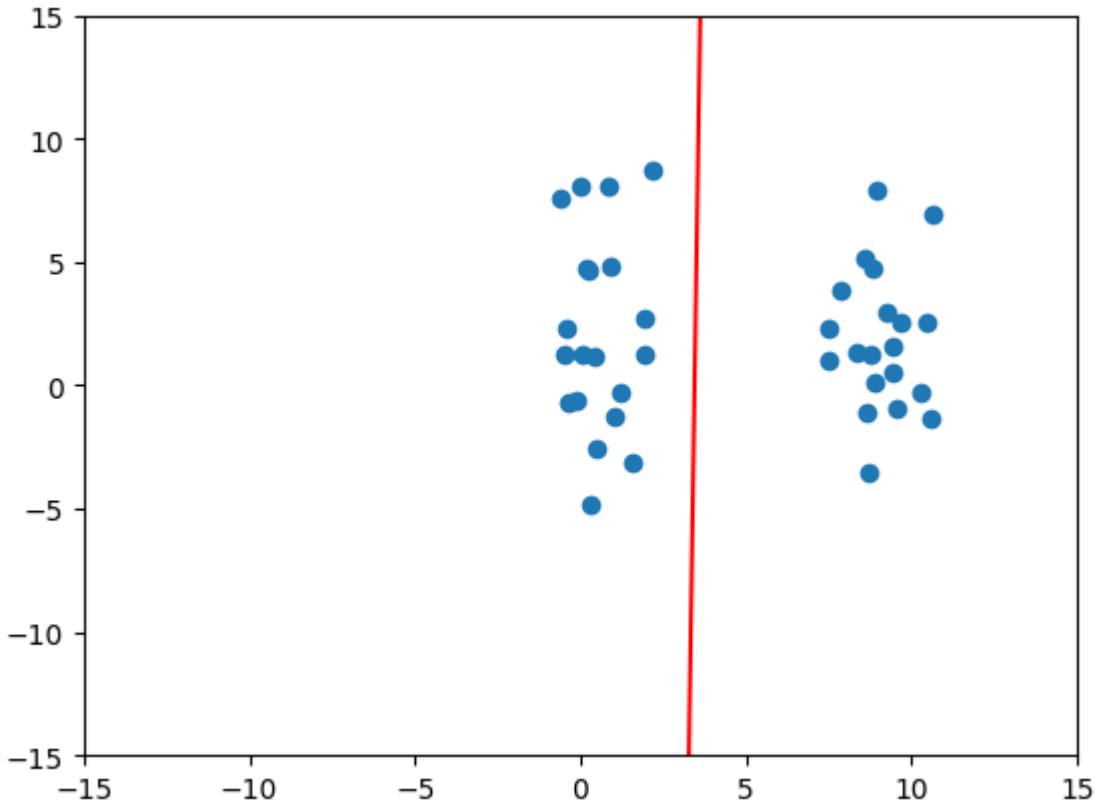
```
accuracy 99.95
```



In [186…

```
# Train data classifier for model 7
model7 = PerceptronC(20)
set_train_x, set_train_y = load_data("set7.train")
model7.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model7.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model7.misclassification_error(set_train_y))
model7.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

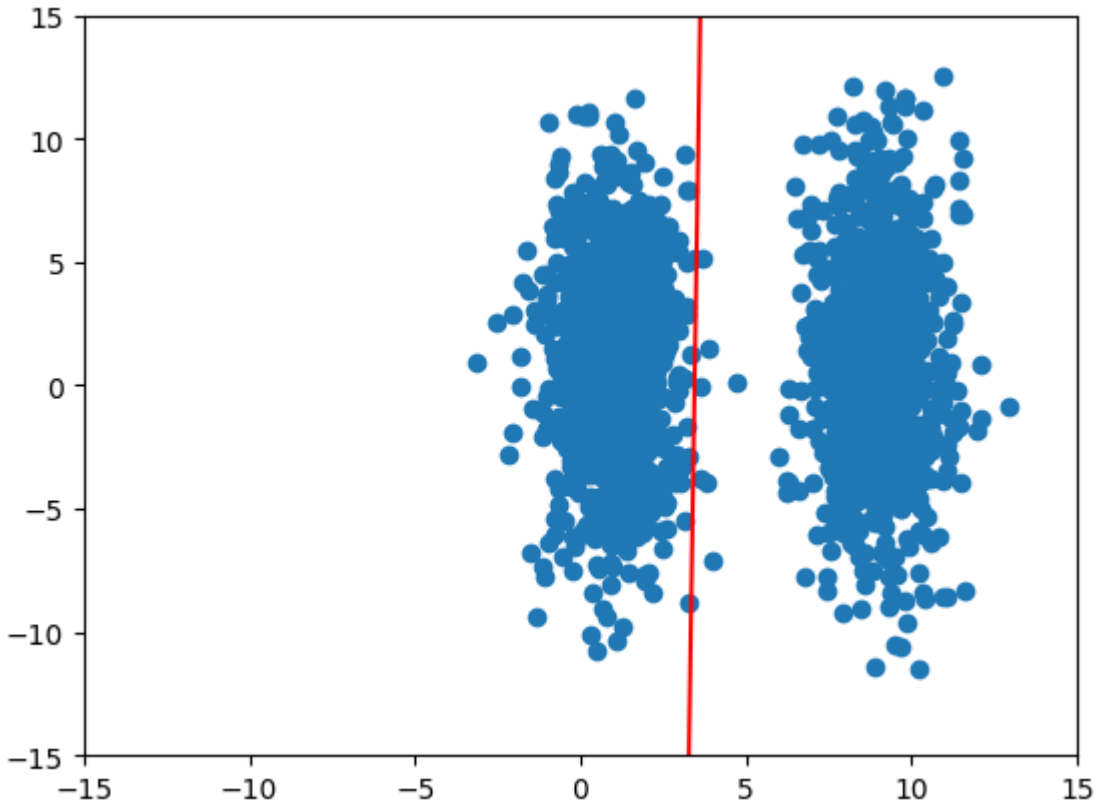
```
accuracy 100.0
```



In [187…

```
# model 7 corresponding to set 7 and its prediction on test data
test_x, test_y = load_data("set.test")
model7.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model7.misclassification_error(test_y))
model7.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

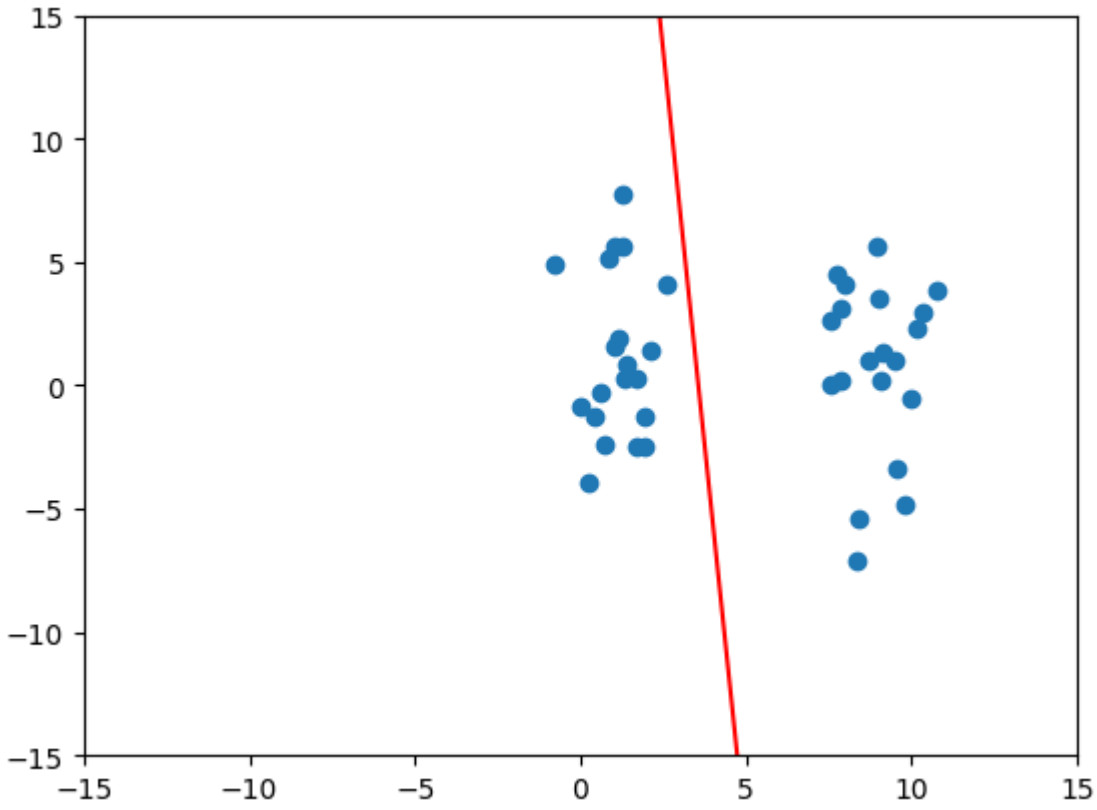
```
accuracy 99.65
```



In [190…

```
# Train data classifier for model 8
model8 = PerceptronC(20)
set_train_x, set_train_y = load_data("set8.train")
model8.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model8.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model8.misclassification_error(set_train_y))
model8.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

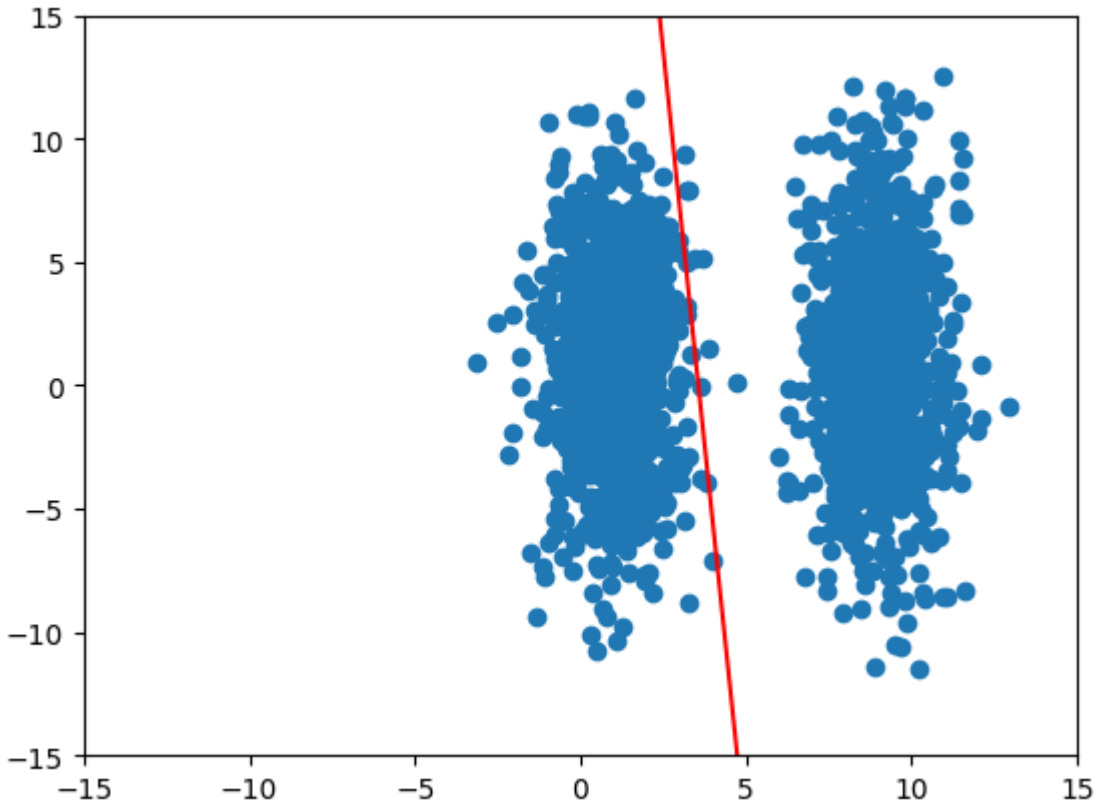
```
accuracy 100.0
```



In [191…

```
# model 8 corresponding to set 8 and its prediction on test data
test_x, test_y = load_data("set.test")
model8.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model8.misclassification_error(test_y))
model8.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

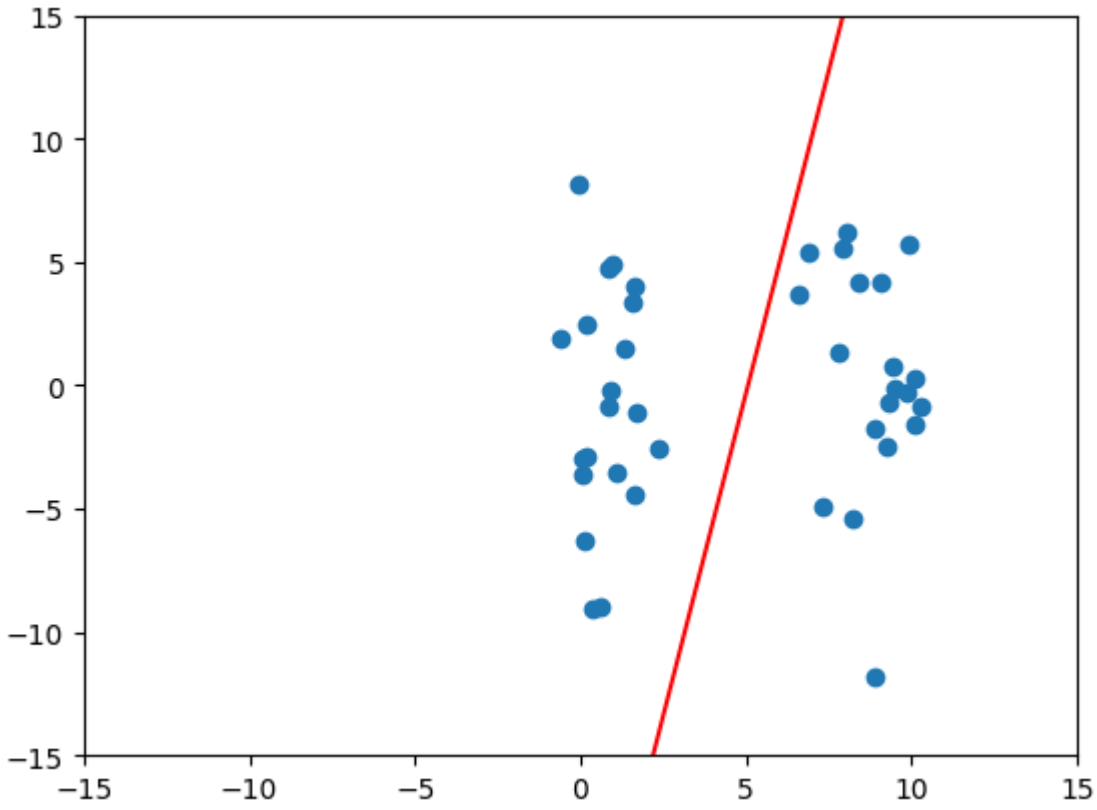
```
accuracy 99.55
```



In [194…

```
# Train data classifier for model 9
model9 = PerceptronC(20)
set_train_x, set_train_y = load_data("set9.train")
model9.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),set
model9.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1)
print("accuracy",model9.misclassification_error(set_train_y))
model9.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

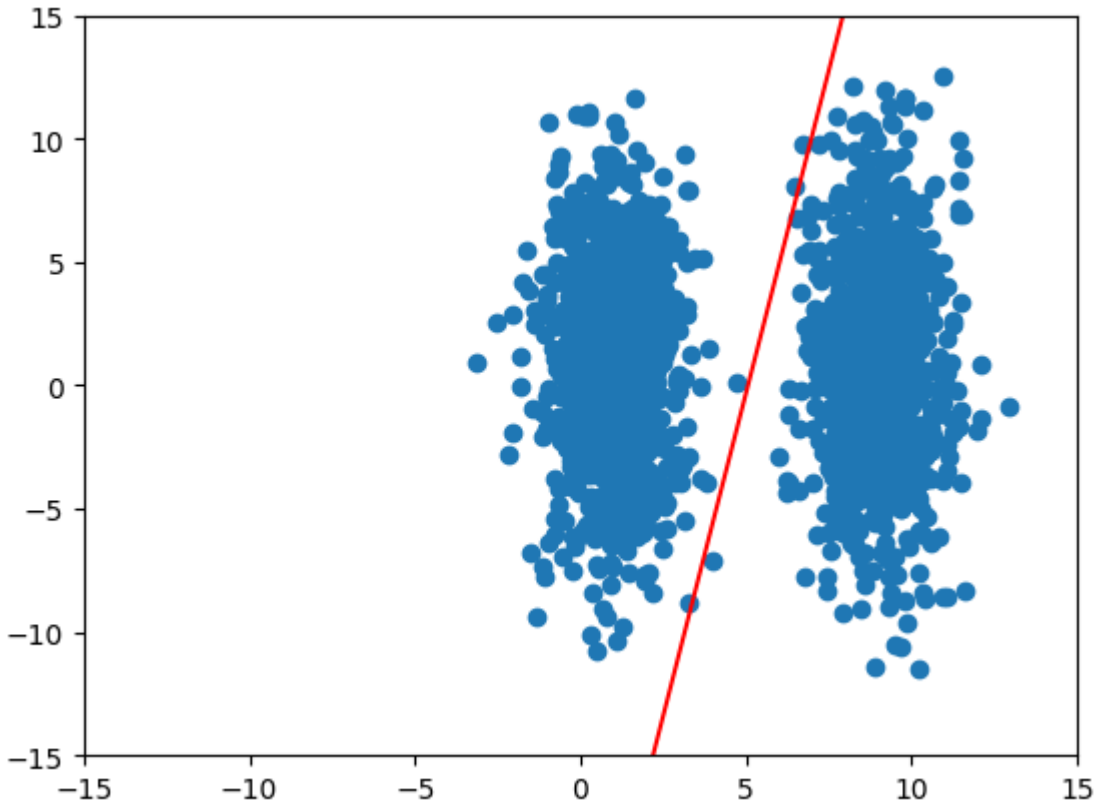
```
accuracy 100.0
```



In [195…

```
# model 9 corresponding to set 9 and its prediction on test data
test_x, test_y = load_data("set.test")
model9.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model9.misclassification_error(test_y))
model9.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

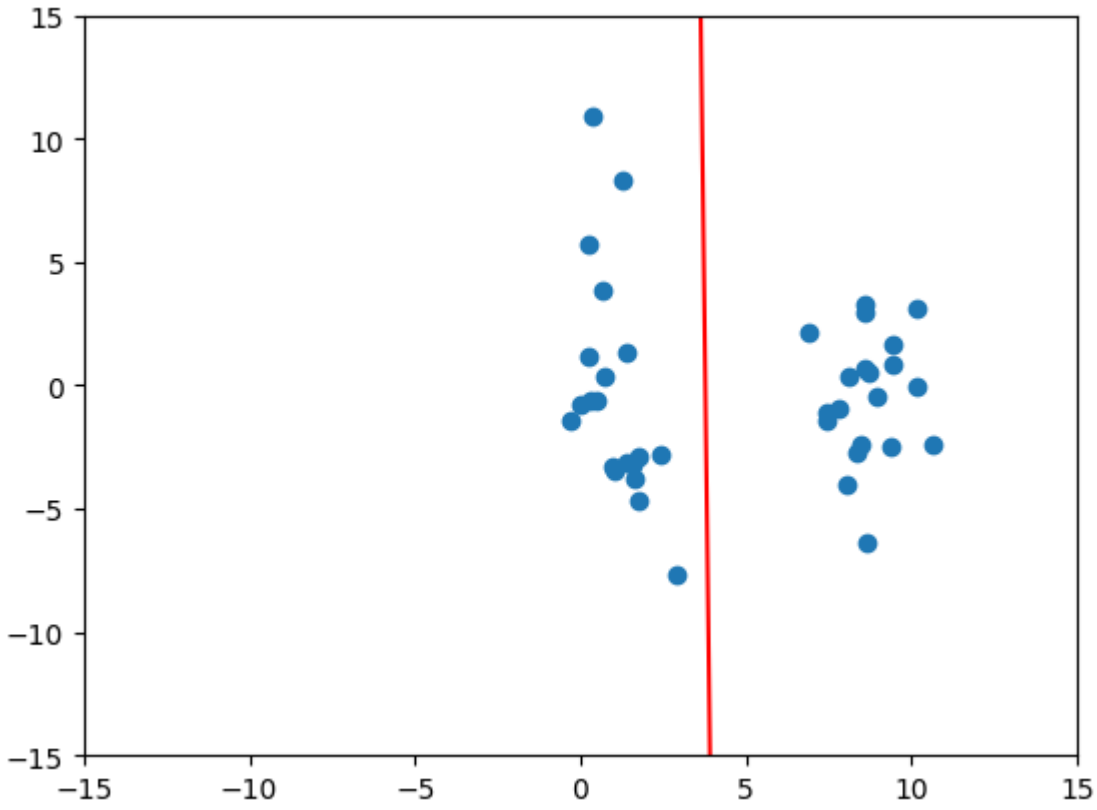
```
accuracy 99.85
```



In [208…

```
model10 = PerceptronC(20)
set_train_x, set_train_y = load_data("set10.train")
model10.fit(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1),se
model10.predict(np.append(set_train_x, np.ones((len(set_train_x),1)), axis=1
print("accuracy",model10.misclassification_error(set_train_y))
model10.plot2D(np.ravel(set_train_x[:,:-1]),np.ravel(set_train_x[:, -1]))
```

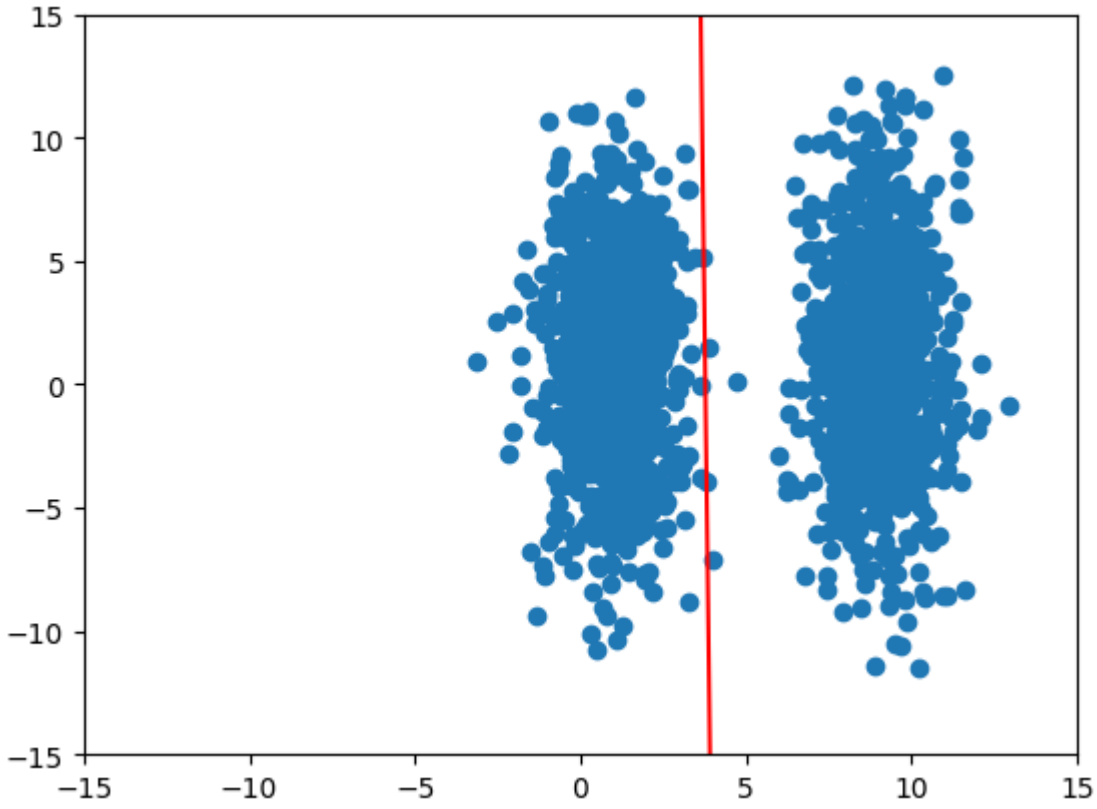
```
accuracy 100.0
```



In [209…

```
test_x, test_y = load_data("set.test")
model10.predict(np.append(test_x, np.ones((len(test_x),1)), axis=1))
print("accuracy",model10.misclassification_error(test_y))
model10.plot2D(np.ravel(test_x[:,:-1]),np.ravel(test_x[:, -1]))
```

```
accuracy 99.8
```



In [ ]: