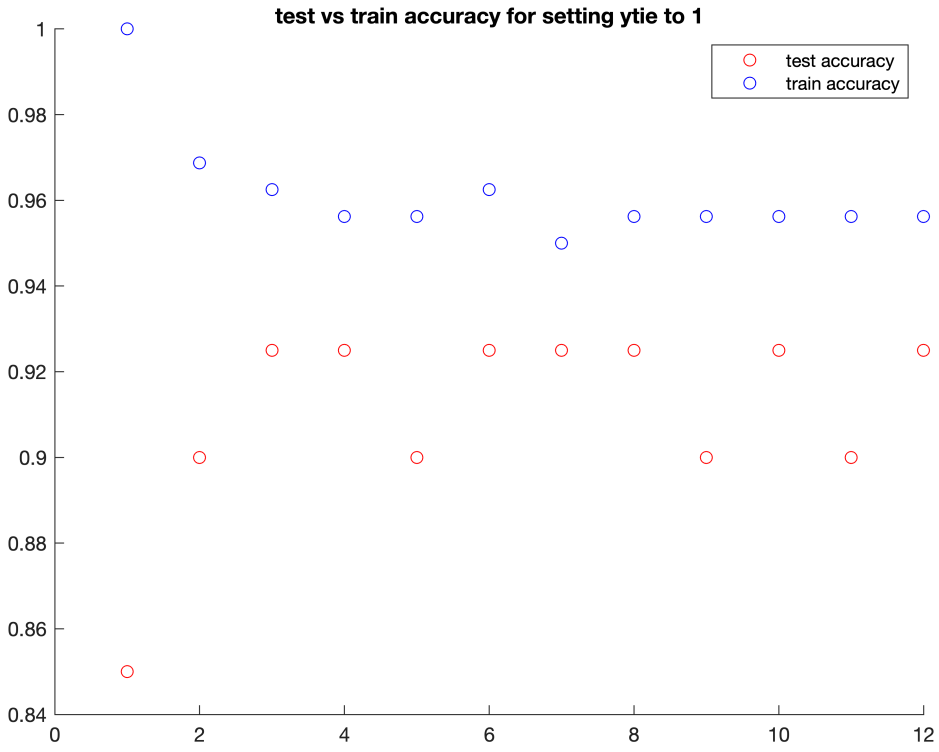
PROBLEM 1 ON HW 4

```
close all;
stat30 = readtable("UCLA_EE_grad_2030.csv");
stat31 = readtable("UCLA_EE_grad_2031.csv");
data = [stat30; stat31];
tes = table2array(data(1:40,:));
tra = table2array(data(41:end,:));
```

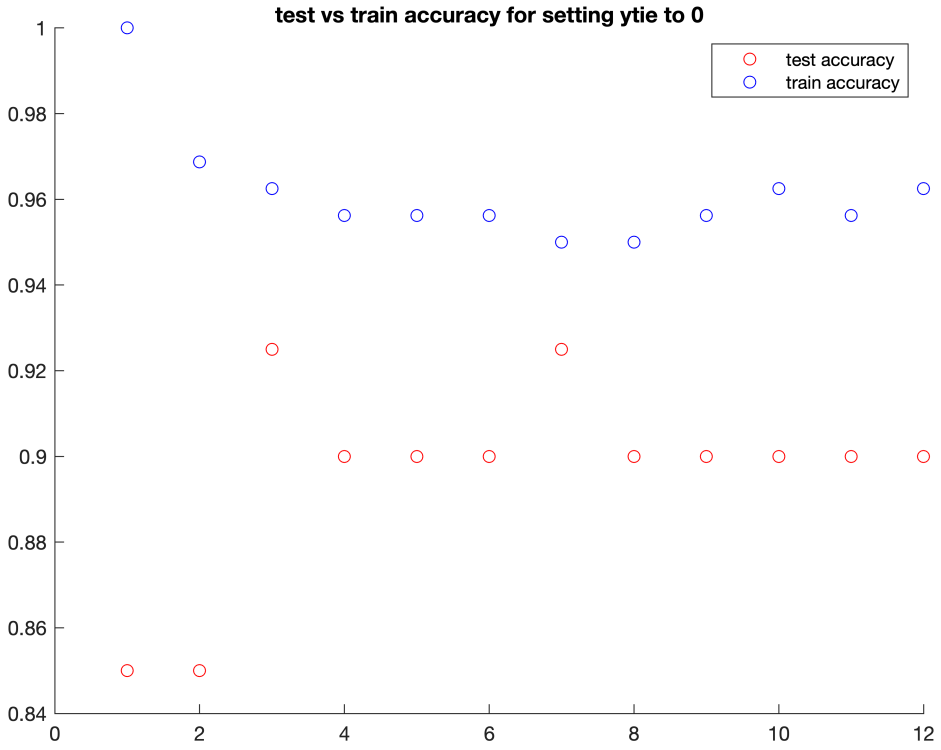
(a) $y_{tie} = 1$: implement k-NN algorithm and plot the training/testing accuracy for k = 1, 2, ...12.

```
testacc = zeros(2, 12);
traiacc = zeros(2, 12);
for k = 1:12
    testyay = [0;0];
    traiyay = [0;0];
    for i = 1:40
        d = sortrows([(tra(:,1)-tes(i,1)).^2+(tra(:,2)-tes(i,2)).^2 tra(:,3)]);
        y1 = sum(d(1:k,2));
        y2 = y1;
        if y1==0, y1 = 1; end
        if y2==0, y2 = -1; end
        y1 = y1 / abs(y1);
        y2 = y2 / abs(y2);
        if y1 == tes(i, 3)
            testyay(1) = testyay(1) + 1;
        end
        if y2 == tes(i, 3)
            testyay(2) = testyay(2) + 1;
        end
    end
    for i = 1:160
        d = sortrows([(tra(:,1)-tra(i,1)).^2+(tra(:,2)-tra(i,2)).^2 tra(:,3)]);
        y1 = sum(d(1:k,2));
        y2 = y1;
        if y1==0, y1 = 1; end
        if y2==0, y2 = -1; end
        y1 = y1 / abs(y1);
        y2 = y2 / abs(y2);
        if y1 == tra(i, 3)
            traiyay(1) = traiyay(1) + 1;
        end
        if y2 == tra(i, 3)
            traiyay(2) = traiyay(2) + 1;
        end
    end
    testacc(1,k) = testyay(1) / 40;
    testacc(2,k) = testyay(2) / 40;
    traiacc(1,k) = traiyay(1) / 160;
    traiacc(2,k) = traiyay(2) / 160;
end
figure(1);
hold on;
```

```
scatter((1:1:12), testacc(1,:)', 'red');
scatter((1:1:12), traiacc(1,:)', 'blue');
legend('test accuracy', 'train accuracy');
title('test vs train accuracy for setting ytie to 1');
```




```
figure(2);
hold on;
scatter((1:1:12), testacc(2,:)', 'red');
scatter((1:1:12), traiacc(2,:)', 'blue');
legend('test accuracy', 'train accuracy');
title('test vs train accuracy for setting ytie to 0')
```



(c) The accuracy against training and testing data is actually fairly good! As k increases the training accuracy drops quite a bit in the beginning but levels out at around 0.96 or 96%. As k increases the testing data accuracy increases by quite a lot in the beginning but levels out at around 0.9 or 90%. A larger k shouldn't make a big difference after 3 or 4. Looking at the graph for ytie = 1 an interesting pattern results. When k is even, the accuracy is noticeably higher than when k is odd. For the ytie = 0 though it doesn't really make a difference. Odd and even make a big difference, though, because when the two classes appear equally. Deciding comes down to your choice. They aren't contradictory to each other. It seems that when ytie is set to 0, even values of k do better, meaning that it is more likely that it is a 1. Setting ytie to 0 should not make it do better when it is even, and so it's the same odd vs even.