CSM148
Homework 2

1. Question 1
    a. You can tell that a model is overfitting when it is performing significantly better on the training data as compared to its accuracy on the test data.
    b. We want to avoid overfitting because we want to make a model which can predict future events. We do not need to have a model that can only perform well on test data which we already have.
    c.
        i. Explain how L1 and L2 regularization methods can mitigate the problem.
            1. Both L1 and L2 regularization are loss functions that help prevent models from becoming too complex. They help models have lower orders and refrain from overfitting to complex datasets. These loss functions attempt to reduce the distance between the true output and the predicted output of the model during training.
        ii. How do these two techniques affect the model weights?
            1. An L1 loss function penalizes based on the absolute value of the difference between the true output and the predicted output. This loss function will reduce less important features to have a weight of 0 which will result in the model having fewer features that actually affect the output.
            2. An L2 loss function penalizes based on the squared difference between the true output and the predicted output. This loss function will reduce less important features' weights to almost 0 which means that every feature will still affect the output albeit a very small amount.
        iii. When would you choose one over the other?
            1. We would use L1 when we have a lot of outliers because it is resistant to outliers and we would use L2 when we have few outliers. But generally, L2 is more commonly used because it is sometimes bad to 0 out coefficients of certain features.

2. Question 2
    a. k = 1
        i. (1,2)
            1. **blue**
                a. This point is closest to the blue point (0,2).
        ii. (2,3)
            1. **blue or red**
                a. This point is equally close to (1, 4) blue and (3, 2) red. So it could either be assigned at random or be left unassigned.
        iii. (10,10)
            1. blue
                a. This point is closest to (3, 6) which is blue.
    b. k = 3
        i. (1,2)
            1. blue
                a. This point is closest to blue (0, 2), red (1, 4), red (2,0)
        ii. (2,3)
            1. red
                a. This point is equidistant from red (3,2) and blue (1,4).
                b. This point is equidistant from red (4,4) and blue (0,2).
                c. The next closest point is red (2,0) so we will make this point red. Or we can randomly assign it.
        iii. (10,10)
            1. red
                a. This point is closest to blue (3,6), red (4,4), and red (3,2).
    c. This might be a problem because KNNs work best when similarly labeled data points are close together and clustered. In this scenario, we might not give much weight to the x-axis feature. To fix this we can normalize the x and y axes so that they can have the same weight in the model.
    d. We can run a KNN and vary the K value ranging from 1 to about ½ the size of the dataset, and then find the K value which gives us the best accuracy on our test set. But we must be careful not to overfit the test set if our K is optimal at low values.

3. Question 3
    a. log (P(Y=1)/1-P(Y=1)) = 1 - 1 * 0 + 2 * 0
    log (P(Y=1)/1-P(Y=1)) = 1
    e = P(Y=1)/1-P(Y=1)
    e - e * P(Y=1) = P(Y=1)
    e = P(Y=1) + e * P(Y=1)
    e = P(Y=1) * (1+e)
    e / (1+e) = P(Y = 1) = **0.73105857863**

    b. **X_1 = 1, X_2 = 0**
    log (P(Y=1)/1-P(Y=1)) = 1 - 1 * 1 + 2 * 0
    log (P(Y=1)/1-P(Y=1)) = 0
    1 = P(Y=1)/1-P(Y=1)
    1-P(Y=1) = P(Y=1)
    P(Y=1) = 1/2

    X_1 = 0, X_2 = 1
    log (P(Y=1)/1-P(Y=1)) = 1 - 1 * 0 + 2 * 1
    log (P(Y=1)/1-P(Y=1)) = 3
    e**3 = P(Y=1)/1-P(Y=1)
    e**3 - e**3 * P(Y=1) = P(Y=1)
    e**3 = (1+e**3) * P(Y=1)
    P(Y=1) = e**3 / (1+e**3) = **0.95257412682243**

    By increasing **X_1** by 1 the P(Y=1) decreases by 0.23, and by increasing **X_2** by 1 the P(Y=1) increases by 0.22.

    c. P(Y=1) = 1 / (1 + e**(-(B0 + B1 * X1 + B2 * X2)))
        i. If X1 and X2 are positive then increasing B0, B1, or B2 will increase P(Y=1), and decreasing B0, B1, or B2 will decrease P(Y=1)
        ii. If X1 or X2 is 0 then increasing B1 and B2 respectively will not do anything.
        iii. If X1 or X2 is negative then increasing B1 and B2 respectively will decrease P(Y=1) and increasing B1 and B2 respectively will decrease P(Y=1)
        iv. More generally, if we increase B0 our probability curve will shift right, this means that P(Y=1) will increase when X values increase and vice versa.
        v. Also if we increase B1 or B2, P(Y=1) will increase in steepness and therefore it will increase more quickly as X values increase, and when we decrease B1 or B2 the opposite occurs.

4. Question 4
    a. FP = 2 (top right)
       FN = 50 (bottom left)
       TP = 45 (bottom right)
       TN = 735 (top left)
    b. Precision = TP/ (TP+FP)
       = 45/(45+2) = **0.9574468085**
       Recall = TP / (TP+FN)
       = 45 / (45+50) = **0.4736842105**
       F1 Score = 2 * (Recall * Precision) / (Recall + Precision)
       = 2 * ((45 / (45+50)) * (45/(45+2))) / ((45 / (45+50)) + (45/(45+2))) =
       0.6338028169
    c. If we have a lower threshold then this model would predict Y_hat = 1 more often. This would increase the FP and the TP rate. By increasing those two factors and decreasing FN and TN I would expect precision to decrease and for recall to increase. The F1 score will most likely remain the same but it would depend on how much the threshold was changed. This can be attributed to the robustness of the F1 score.

       Precision\/ = TP^/ (TP^+FP^)

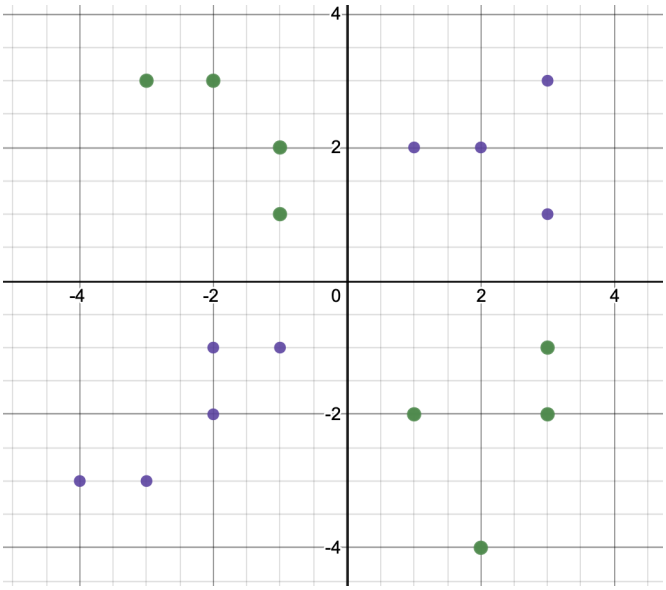
       Recall^ = TP^ / (TP^+FN\/)

       F1 Score~ = 2 * (Recall^ * Precision\/) / (Recall^ + Precision\/)
    d. We cannot compute the AUC because we only have one point on the ROC. We need a range of threshold values to make a ROC so we can calculate the AUC.

5. Question 5
    a. Removing one of the non-circled points will not change our decision boundary. With an SVM the points not circled are not within the margin so we do not look at them when making the decision boundary.
    b. A soft margin is when you try to find a line to separate, but tolerate one or a few misclassified dots and we can tolerate some misclassifications.
    A hard margin is when you can completely linearly separate the data points and when we do not want to misclassify anything.
    In this case, we can use a hard margin because the data is linearly separable.
    c. In this case, if we use an RBF our decision boundary might get closer to the "-"s because there are more of them. It might bend towards them and it might overfit the training data set. If we used a linear SVM the line would pass vertically down the middle of the dataset. Because our data is linearly separable we should use a linear SVM.
    d. Gamma is a parameter of RBF kernel SVM which is the "spread" of the kernel a.k.a. the decision region.
    Low Gamma = "curve" of decision boundary is low and the decision boundary is broad.
    High Gamma = "curve" of decision boundary is high and this creates islands of decision boundaries around certain data points.

    C is a parameter of the RBF kernel SVM which is a penalty for misclassifying a data point.
    Low C = the model is ok with a misclassified point
    High C = the model is not ok with a misclassified point and gets heavily penalized for misclassified data points.

6. Question 6
    a.
        i. Latitude and room type
            1. This cross can help us classify how different room types are worth more or less depending on location (specifically latitude). This could be interesting because this would allow us to be able to make decisions based on the interaction of location and type of room.
        ii. Latitude and Longitude
            1. This would help us pinpoint certain houses. It would allow us to be able to get a pretty good location instead of splitting up just longitude and just latitude.
    b.



        i. Here is an example dataset where it has two features X and Y and it would perform poorly without crossing the features X and Y to separate the data. A linear regression model could not linearly separate this data in this 2-dimensional setup but if the 3rd dimension of X*Y was introduced and a hyperplane could be formed which clearly splits these data points.