Homework 3

1. Assume you have a dataset D with n samples. You want to create bootstrappeddatasets of size k using sampling with replacement.

   (a) Assume you create one bootstrapped dataset of size k. Additionally, assume thatwe fix a data point x ∈ D. What is the probability that x does not appear in the bootstrapped dataset?

probability that the first bootstrap observation is not x is $1 - 1/n$

probability that any bootstrap observation is not x is $1 - 1/n$ (the selections are independent)

probability that x does not appear in the bootstrapped dataset = $(1 - 1/n)^k$

Explanation: $(1 - 1/n)(1 - 1/n) \dots (1 - 1/n)[ktimes]$

(b) Now, assume that k = n. What does the probability converge to as n goes to infinity? What does this limit imply about the percentage of the original dataset that will not be sampled at n gets large?

We know, $e^x = lim(n \to infinity)(1 + x/n)^n$

If we set x = -1, we get: e^(-1) $= lim(n \to infinity)(1 - 1/n)^n$

So, as n goes to infinity, the probability converges to e^(-1)

The value of e^(-1) is 0.3678 so the percentage of the original dataset that will not be sampled as n gets large is 36.8%

(c) Assume that you create r bootstrapped datasets of size k each. Additionally, assume that we fix a data point x ∈ D. What is the probability that x does not appear in any bootstrapped dataset?

The probability that a particular data point x is not selected in any of the r bootstrapped datasets of size k is: $[(1 - 1/n)^k]^r$

This is the probability that x is not selected in any single bootstrapped dataset raised to the power of the number of bootstrapped datasets. Therefore, the probability that x is not selected in any of the r bootstrapped datasets is the product of the probability that x is not selected in each individual bootstrapped dataset.

2. In this question, let us consider the difference between lasso and ridge regularization.Recall that the lasso regularization of a vector β is λ Pki=1 |βi| and that the ridge regu-larization is λ Pki=1 β2i . Consider two vectors x1 = [4, 5] and x2 = [−2, 2]. Additionally,set λ = 1.

   (a) What is the lasso regularization of x1 and x2? What is the change in the lasso regularization when going from x1 to x2?

The lasso regularization of x1 is $\lambda(|4| + |5|) = 1 * 9 = 9$

The lasso regularization of x2 is $\lambda(|-2| + |2|) = 1 * 4 = 4$

The change in lasso regularization when going from x1 to x2 is $9 - 4 = 5$

(b) What is the ridge regularization of x1 and x2? What is the change in the ridge regularization when going from x1 to x2?

The ridge regularization of x1 is $\lambda(4^2 + 5^2) = 41$

the ridge regularization of x2 is $\lambda((-2)^2 + 2^2) = 8.$

The change in ridge regularization when going from x1 to x2 is $41 - 8 = 33$

(c) In your own words, explain the effects of ridge vs lasso regularization.

Ridge regularization and lasso regularization are two common techniques for preventing overfitting in machine learning models. Both methods add a penalty term to the loss function that the model is trying to optimize. Ridge regularization adds a penalty proportional to the sum of the squared values of the coefficients, while lasso regularization adds a penalty proportional to the sum of the absolute values of the coefficients.

The effect of ridge regularization is to shrink the coefficients towards zero, but not to zero. This can be useful in situations where all of the features are potentially relevant, but some may have very small effects. Ridge regularization can help to prevent overfitting by reducing the overall magnitude of the coefficients.

The effect of lasso regularization is to shrink some coefficients all the way to zero, effectively removing those features from the model. This can be useful in situations where there are many features, some of which may be irrelevant or redundant. Lasso

regularization can help to prevent overfitting by effectively performing feature selection, leading to simpler and more interpretable models.

3. Coding Question - Plot the Voronoi regions for k = 1, 2, 3, 4 using the k-nearestneighbours classifier on the points: [[1, 1], [4, 1], [2, 3], [3, 3], [3, 4], [5, 4], [6, 5], [4,5]]. The first 4 points are in class 0 and the rest are in class 1. A .ipynb file has beenprovided with starter code to get you started. Did you find anything curious about the plots? How do you explain them?

```
# Define the data points
X = np.array([[1, 1], [4, 1], [2, 3], [3, 3], [3, 4], [5, 4], [6, 5], [4, 5]])
y = np.array([0, 0, 0, 0, 1, 1, 1, 1])

# Define the KNN classifier with k=1,2,3,4
for i, k in enumerate(range(1, 5)):
    clf = KNeighborsClassifier(n_neighbors=k)
    clf.fit(X, y)

    # Plot the decision boundary for the classifier
    draw_contour(X, y, clf, class_labels=["Class 0", "Class 1"])
    plt.title(f"KNN Classifier with k={k}")
    plt.show()
```

One curious thing about the plots is that as k increases, the decision boundary becomes smoother and more generalized. This is because the classification is based on a larger number of neighbours, which can result in a more stable decision boundary that is less affected by outliers or noise in the data. However, increasing k too much can also lead to underfitting and a loss of accuracy in the classification.

4. Coding Question - Plot the logistic function $11+e-(\beta0+\beta1\times x)$ for $x \in [-10, 10]$ and thefollowing parameter values:

   (a) $\beta0 = 2$ and $\beta1 = 1$

   (b) $\beta0 = 10$ and $\beta1 = 2$

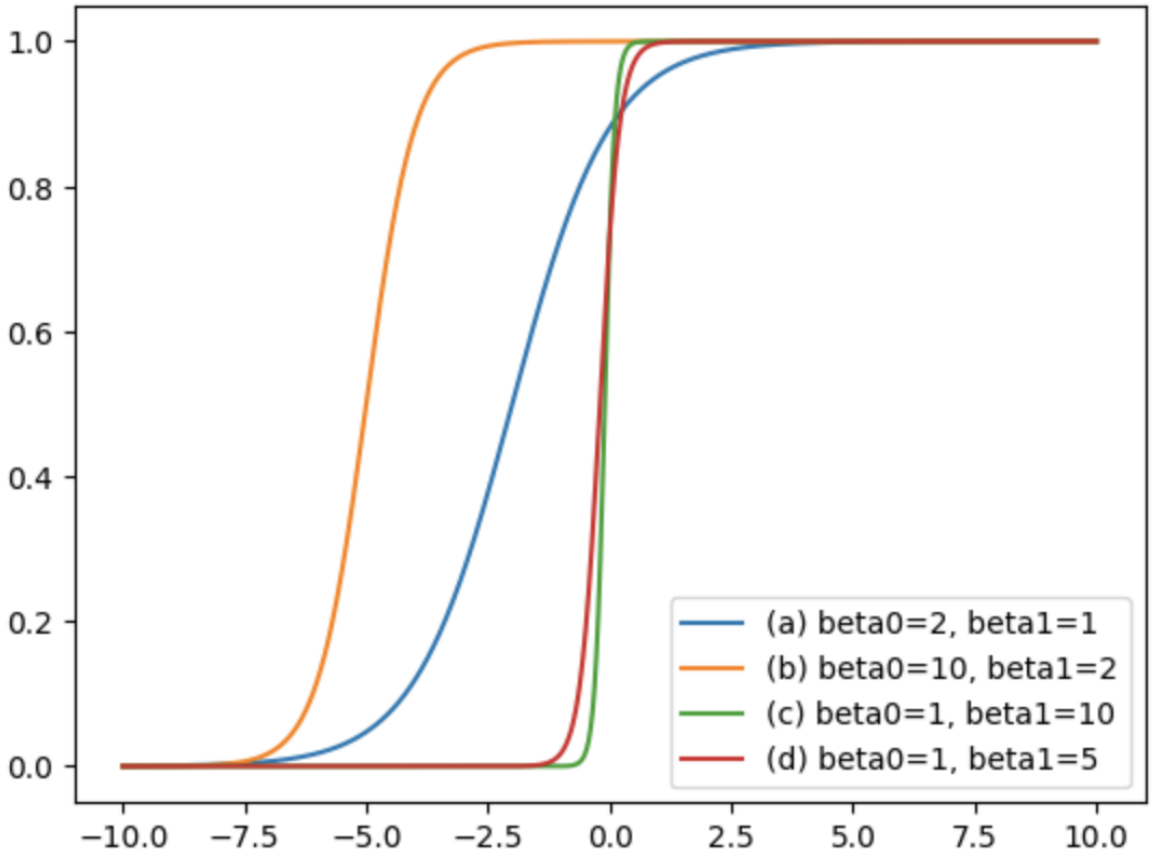
   (c) $\beta0 = 1$ and $\beta1 = 10$

   (d) $\beta0 = 1$ and $\beta1 = 5$

   For what choices of $\beta0$, $\beta1$ does the function become steeper?

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

def logistic(x, b0, b1):
    return 1 / (1 + np.exp(-(b0 + b1 * x)))

x = np.linspace(-10, 10, 1000)

# (a) beta0 = 2 and beta1 = 1
y_a = logistic(x, 2, 1)

# (b) beta0 = 10 and beta1 = 2
y_b = logistic(x, 10, 2)

# (c) beta0 = 1 and beta1 = 10
y_c = logistic(x, 1, 10)

# (d) beta0 = 1 and beta1 = 5
y_d = logistic(x, 1, 5)

#Print the slopes
slope, intercept = np.polyfit(x, y_a, 1)
print(slope)
slope, intercept = np.polyfit(x, y_b, 1)
print(slope)
slope, intercept = np.polyfit(x, y_c, 1)
print(slope)
slope, intercept = np.polyfit(x, y_d, 1)
print(slope)

# Plot the functions
plt.plot(x, y_a, label='(a) beta0=2, beta1=1')
plt.plot(x, y_b, label='(b) beta0=10, beta1=2')
plt.plot(x, y_c, label='(c) beta0=1, beta1=10')
plt.plot(x, y_d, label='(d) beta0=1, beta1=5')
plt.legend()
plt.show()
```

```
0.06947959697917168
0.055616572528207564
0.07489297233251276
0.07479673947990118
```



The function for all choices of β0, β1 becomes steeper. The (c) β0 = 1 and β1 = 10 has the steepest curve, closely followed by (d) β0 = 1 and β1 = 5.

5. Recall the problem of ridge linear regression with n points and k features:LRidge(β) = 1nnXi=1(yi − βT xi)2 + λkXj=1β2jwhere λ is a hyper-parameter. The goal is to minimize LRidge(β) in terms of β for afixed training dataset (yi, xi) and parameter λ.

   (a) In your own words, explain the purpose of using ridge regression over standardlinear regression.

The purpose of using ridge regression over standard linear regression is to deal with the problem of overfitting, which occurs when the model fits the training data too closely and fails to generalize to new data. Ridge regression introduces a penalty term to the

objective function that shrinks the regression coefficients towards zero, thereby reducing the model complexity and improving its generalization performance.

(b) As λ gets larger, how will this affect β? What value do we expect β to convergeon?

As λ gets larger, the penalty term in the objective function becomes more dominant, and the regression coefficients shrink towards zero. In other words, as λ increases, the model becomes more constrained, and the effect of the regularization term becomes stronger. Eventually, as λ approaches infinity, all the regression coefficients will be shrunk to zero, and the model will become a simple intercept-only model.

(c) Consider parameters βλ that were trained using ridge linear regression with aspecific lambda. Let us consider the test MSE using βλ. Note that the test MSEis the following formula for the test data1nnXi=1(yi − βTλ xi)2 and does not include regularization.Sketch a plot of how you expect the Test MSE to change as a function of λ. Yoursketch should be a smooth curve that shows how the test MSE changes as λ goesfrom 0 to ∞. Provide justification for your plot. Assume that the right most edge of the graph is where λ is at ∞. Additionally, assume that the linear regressionwithout regularization is overfitting.

The test MSE is a measure of how well the ridge regression model generalizes to new data. As λ increases, the size of the coefficients β decreases, leading to a simpler and more constrained model. Initially, the test MSE will decrease as the model becomes less prone to overfitting. However, as λ increases further, the model will become too constrained, leading to an increase in test MSE due to underfitting. We expect the test MSE to follow a U-shaped curve as a function of λ, with a minimum point at some intermediate value of λ.

6. True of False questions. For each statement, decide whether the statement is True orFalse and provide justification (full credit for the correct justification).

   (a) In L2 regularization of linear regression, many coefficients will generally be zero.

False. In L2 regularization of linear regression, coefficients will tend to be small but not necessarily zero. The degree of shrinkage depends on the strength of the regularization parameter.

(b) In the leave one out cross validation over the data set of size N, we create andtrain N/2 models.

False. In leave-one-out cross validation, we create and train N models, one for each data point in the dataset. For each model, we use all the data except for one point to train the model and use the remaining point to test the model.

(c) 95% confidence interval refers to the interval where 95% of the training data lies.

False. A confidence interval is an interval calculated from the sample data that is likely to contain the true population parameter with a certain level of confidence (e.g., 95%). It has nothing to do with where the training data lies.

(d) If K out of J features have already been selected in Stepwise Variable Selection,then we will train J − K new models to select the next feature to add.

False. In stepwise variable selection, if K out of J features have already been selected, we will train only J-K new models, one for each of the remaining unselected features, to select the next feature to add.

(e) P (A|B) = P (B|A) if P(A) = P (B) and P (A) is not zero.

True. Bayes' Theorem states P(A|B) = P(B|A) P(A) / P(B) if P(A) = P(B) then P(A|B) = P(B|A).