Homework 4

1. Suppose we have the following confusion matrix outputted from a logistic regression using the probability threshold P (Y = P ositive) ≥ t , i.e. we classify the sample as Positive if P (Y = P ositive) is greater than t otherwise we classify as Negative.

(a) Compute the false positive and false negative rates.

Confusion Matrix:

FN	TP
TN	FP

=

9	36
25	10

False Positive Rate: FP / (TN + FP) = 0.2857

Percentage: 28.57%

False Negative Rate: FN / (TP + FN) = 0.2

Percentage: 20%

(b) How would you expect the confusion matrix to change if we increased t?

If we increase the probability threshold (t), it means we are making the decision boundary more strict. This would lead to classifying fewer samples as positive. Consequently, we would expect the following changes in the confusion matrix:

- The false positive rate (FPR) would decrease. This means that fewer negative samples would be classified as positive, resulting in a lower number of false positives (FP) in the confusion matrix.
- The false negative rate (FNR) would increase. This means that more positive samples would be classified as negative, resulting in a higher number of false negatives (FN) in the confusion matrix.

- The true positive rate (TPR) would decrease. This means that the classification model would correctly identify fewer positive samples as positive.
- The true negative rate (TNR) would increase. This means that the classification model would correctly identify more negative samples as negative.

2. Bayes Theorem. Consider that you own a small restaurant. You have a smoke detector

in your kitchen. The chances that a hazardous fire occurs in the kitchen is pretty rare,

say 1%. The smoke alarm is pretty accurate in detecting such fire and it sounds the

alarm 99% of the time. However, the alarm is poorly calibrated and it also sounds

an alarm sometimes when there is no fire, due to smoke detected from cooking. The

accuracy of the smoke alarm under non-fire condition is 90%.

(a) What is the probability that the smoke detector sounds an alarm?

P(A) = Smoke detector sounds the alarm

P(B) = There is a fire in the kitchen

P(B) = 0.01

P(A|B) = 0.99

P(A|B') = 0.10 where B' = when there is no fire in the kitchen

We have to calculate P(A)

By Law of total probability, $P(A) = P(B)P(A|B) + P(B')P(A|B')$

P(A) = 0.01*0.99 + 0.99*0.1 = 0.1089

(b) Given that you heard the alarm sound, what is the probability that there was

actually a fire?

Using Bayes Theorem,

P(B|A) = P(A|B)*P(B) / P(A)

P(B|A) = 0.99*0.01/0.1089 = 0.0909

(c) Comment on how useful the smoke detector is and would you consider replacing it?

Because the probability that there was actually a fire when the alarm sounds is 9% which is quite low, I would consider replacing the alarm.

3.

To prove that the derivative of the loss function with respect to βj is equal to:

dL(β)/dβj = n * Σ (1/(1 + e^(-β^T xi)) - yi) * xji

we will compute the derivative step-by-step using the given loss function. Let's start by taking the derivative of the loss function with respect to βj:

dL(β)/dβj = -n * Σ (yi * d/dβj log(1/(1 + e^(-β^T xi))) + (1 - yi) * d/dβj log(1 - 1/(1 + e^(-β^T xi))))

Now, we can focus on each term separately. Let's first compute the derivative of the log function:

d/dβj log(1/(1 + e^(-β^T xi)))

To simplify the calculation, let's rewrite the term as:

1/(1 + e^(-β^T xi)) = (1 + e^(-β^T xi))^(-1)

Using the chain rule, the derivative of the log function can be computed as:

d/dβj log(1/(1 + e^(-β^T xi))) = d/dβj (-β^T xi) - log(1 + e^(-β^T xi))

Taking the derivative of -β^T xi with respect to βj gives:

d/dβj (-β^T xi) = -xji

Now, compute the derivative of log(1 + e^(-β^T xi)) with respect to βj. We will use the chain rule again:

d/dβj log(1 + e^(-β^T xi)) = 1/(1 + e^(-β^T xi)) * d/dβj (1 + e^(-β^T xi))

Taking the derivative of (1 + e^(-β^T xi)) with respect to βj gives:

d/dβj (1 + e^(-β^T xi)) = -xi * e^(-β^T xi)

Combining these results, we can now compute the derivative of the log term:

d/dβj log(1/(1 + e^(-β^T xi))) = -xji - xi * e^(-β^T xi) * 1/(1 + e^(-β^T xi))

Simplifying this expression, we get:

d/dβj log(1/(1 + e^(-β^T xi))) = -xji + xji/(1 + e^(-β^T xi))

Now, let's compute the derivative of the second term:

d/dβj log(1 - 1/(1 + e^(-β^T xi)))

Using similar steps as before, we find:

d/dβj log(1 - 1/(1 + e^(-β^T xi))) = -xi/(1 + e^(-β^T xi))

Plugging these derivatives back into the original equation, we have:

dL(β)/dβj = -n * Σ (yi * (-xji + xji/(1 + e^(-β^T xi))) + (1 - yi) * (-xi/(1 + e^(-β^T xi))))

Simplifying further:

dL(β)/dβj = n * Σ (yi * (xji/(1 + e^(-β^T xi))) - (1 - yi)

4. In your own words, explain the following types of multi-class classification methods:

(a) One vs All

(b) All vs All

Provide the advantages and disadvantages of each method

(a) One vs All (also known as One vs Rest):
In the One vs All approach, we treat each class as a binary classification problem against all the other classes combined. For example, if we have K classes, we create K binary classifiers, where each classifier is trained to distinguish one class from the rest.

Advantages:

1. Simplicity: It is straightforward to implement and understand since it reduces the multi-class problem to multiple binary classification problems.
2. Efficiency: Training K classifiers is typically faster than training K(K-1)/2 classifiers in the All vs All approach (for large K).

Disadvantages:

1. Imbalanced class distribution: The One vs All approach may encounter imbalanced class distribution issues, especially when some classes have significantly fewer samples than others. This can lead to biased classifiers and suboptimal performance on minority classes.
2. Classification inconsistency: Since each classifier is trained independently, there is no guarantee of consistent classification results. Some samples may be classified as multiple classes or have no predicted class at all.

(b) All vs All (also known as One vs One):
In the All vs All approach, we build K(K-1)/2 binary classifiers, where each classifier is trained to distinguish between a pair of classes. Each class is compared against every other class, resulting in a voting or counting scheme to determine the final predicted class.

Advantages:

1. Balanced class distribution: The All vs All approach can handle imbalanced class distribution more effectively since each binary classifier is trained on a balanced subset of data that consists of samples from two classes only.
2. Classification confidence: By using a voting or counting scheme, the All vs All approach can provide probability or confidence estimates for the predicted class.

Disadvantages:

1. Computational complexity: Training K(K-1)/2 classifiers can be computationally expensive, especially when dealing with a large number of classes.
2. Increased training time: With a larger number of classifiers, the training time increases significantly compared to the One vs All approach.
3. Overlapping regions: In some cases, the decision boundaries between classes may overlap, leading to ambiguous or conflicting predictions for certain samples.

Overall, the choice between One vs All and All vs All depends on factors such as the number of classes, class distribution, computational resources, and the desired trade-offs between efficiency and classification performance.

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

(a) For a classification model, positive predictive value is the probability that a model classifies a sample as positive given that the true label of the sample is positive.

False. Positive predictive value (PPV), also known as precision, is the probability that a sample classified as positive by the model is actually positive. It is calculated as the number of true positives divided by the sum of true positives and false positives. It represents the accuracy of the model in predicting positive samples among all the samples it classified as positive.

(b) Assume we are working with a multinomial logistic regression such that P (Y = i|X) = eβ0,i+β1,iX P (Y = K|X) for 1 ≤ i ≤ K − 1. For a dataset with 1 feature and 4 possible class labels, the number of learnable parameters βj,i is 8.

True. In multinomial logistic regression, where there are K possible class labels, and assuming 1 feature, the number of learnable parameters βj,i is (K - 1) * (number of features). In this case, with 1 feature and 4 possible class labels, the number of learnable parameters is 3 * 1 = 3.

(c) If the log-odds function is modeled as a quadratic, logistic regression can provide a non-linear decision boundary.

True. Logistic regression models the log-odds function, also known as the logit, which is a linear function of the input features. However, if the log-odds function is modeled as a quadratic function of the input features, logistic regression can capture non-linear relationships between the features and the probability of the class labels. This allows logistic regression to provide a non-linear decision boundary.

(d) You are building a classifier to detect fraudulent credit card transactions. Your employer states that a 90% success in detection of fraudulent transactions is good enough. You test your model on the next 1000 transactions and get a 97% test accuracy. Therefore, your model is doing much better than what is required.

False. Test accuracy alone is not sufficient to determine the performance of a classifier, especially in imbalanced datasets. In the case of fraudulent credit card transactions, the dataset is likely to be highly imbalanced, with a small proportion of fraudulent transactions. A high test accuracy of 97% may be misleading if the model is not effectively detecting the fraudulent transactions. Additional evaluation metrics such as precision, recall, and F1 score should be considered to assess the performance accurately.

(e) For a very good classification model, we expect the confusion table to be dominated by diagonal entries.

False. For a very good classification model, we expect the confusion table to have high values on the diagonal (true positives and true negatives), but it does not necessarily mean that the diagonal entries dominate the table. The confusion table provides a summary of the model's predictions and the actual class labels, including true positives, true negatives, false positives, and false negatives. The distribution of values in the confusion table depends on the specific dataset and the model's performance.