**Name:** \_Cameron Stark\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CS 455 / CS 595a –Exam 2 – Spring 2020**

**Instructions:**

This exam is being given to you as a take-home examination. The deadline for submission is 11:59pm ET Friday, April 17, 2020 via Canvas.

For submission, please either type your work, or handwrite and scan/photograph. If you scan or photograph the exam, I ask that you compile the work into a single document such as cutting and pasting the images into a MS Word document. Please make sure that everything is readable and your handwriting is dark enough on the exam.

Each problem is worth 10 points. To receive maximum credit, you must show your work. Partial credit will be awarded if you are demonstrating progress toward the solution, but a misstep within the process results in an incorrect answer.

You are not limited to just the pages provided. If you need more space, please use additional pages, but try to assemble the exam with those pages inserted with the problem they are supporting.

If you believe you have found a typographic error, please email Dr. Stansbury ([stansbur@erau.edu](mailto:stansbur@erau.edu)). Check Canvas periodically to see if any corrections/clarifications have been posted.

**Grades:**

|  |  |  |
| --- | --- | --- |
|  | **Available** | **Awarded** |
| **Problem #1** | 10 |  |
| **Problem #2** | 10 |  |
| **Problem #3** | 10 |  |
| **Problem #4** | 10 |  |
| **Problem #5** | 10 |  |
| **Problem #6** | 10 |  |
| **Problem #7** | 10 |  |
| **Problem #8** | 10 |  |
| **Problem #9** | 10 |  |
| **Problem #10** | 10 |  |
|  | **Final Grade** |  |

**Problem #1: Short-Answer – Theory and Concepts – Linear and Logistic Regression**

Answer each question in complete sentences. Be sure to clearly justify your answer.

1. **What is a reason to choose a gradient descent optimization method to train a linear regression model versus utilizing the closed-form solution to solve for ?**

Choosing a gradient descent optimization method for training a linear regression model is a more optimal solution because with the closed-form method it suffers from a long run time which leads to greater costs but are returned the optimal set of parameters at the end. However, with gradient descent method every iteration the parameters are modified by changing the parameter that had the most impact on the error for that iteration, this approach allows the method to run faster and learn more efficiently per iteration.

1. **If implementing a linear regression model, would you consider utilizing polynomial features if your model were underfitting or overfitting? Why?**

Polynomial features would be implemented if the linear regression model were underfitting, because that means the data is not entirely linear, therefore polynomial features can be applied to the model and with increasing the degree can try and match the non-linear path of the data.

1. **If implementing a linear regression model, would you consider regularizing your model if your model were underfitting or overfitting? Why?**

Regularization whether it is Ridge or Lasso, would be used on a linear regression model if the model was overfitting the data, the regularization in this case would constrain the model, by reducing or restricting the range of data that can be used as a possible parameter, which can lead to a more accurate model by reducing overfit but also because there is a less to data to model the performance is improved.

1. **Describe how logistic regression, a member of the regression family of algorithms, is suitable for implementing a binary classifier.**

Logistic Regression is useful in that it can implement binary classifiers because the regression model works as a method of testing for positive and negative cases, and searches for a division of the two cases.

**Problem #2: Linear Regression**

Apply **batch gradient descent of a linear regression model with ridge regularization** for the following training set and initial set of weights ( and bias () for training step.

Determine for one training step with **batch size m=4:**

1. **Error gradient** (using MSE as loss)
2. **Updated feature weight vector**,

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Training Inputs and Targets**     |  |  |  | | --- | --- | --- | | X1 | X2 | Y | | 0.75 | 0.10 | 0.7 | | 0.5 | 0.45 | 0.8 | | 0.25 | 0.5 | 0.1 | | 0.86 | 0.93 | 0.6 | | **Model Initial Parameters and Hyperparameters:**  **Initial Bias + weights:** ,  **Regularization term**:  **Learning Rate**: |

a.

MSE =

0.83\*0.75 + 0.72\*0.1 = 0.69 – 0.7\*0.75 = 0.165

0.83\*0.5 + 0.72\*0.45 = 0.73 – 0.8\*0.75 = 0.13

0.83\*0.25 + 0.72\*0.5 = 0.56 – 0.1\*0.75 = 0.485

0.83\*0.86 + 0.72\*0.93 = 1.38 – 0.6\*0.75 = 0.93

MSE = (0.165^2 + 0.13^2 + 0.485^2 + 0.93^2) \* (2/4) = 0.572

b.

**Problem #3: Short-Answer – Theory and Concepts – Support Vector Machines**

Answer each question in complete sentences. Be sure to clearly justify your answer.

1. **The margin limits of a support vector machine can be adjusted via the slack term with low slack (few to no violations) to high slack (many violations). Name one reason you would consider increasing the slack hyperparameter for your model, and briefly justify your answer.**

A reason to consider increasing the margin of valid results that a SVM would return, would be the dataset that is being worked on wants not a direct answer but rather a set of answers that still gives an optimum solution for example different types of food can all give the same amount of nutrients, therefore increasing the margin could allow more food type options into the answer pool.

1. **What are similarity features, and how could they be used within an SVM to address non-linearity?**

Similarity Features are a new set of data points, with landmarks being chosen from the original non-linear data set as points to be passed into the Similarity functions for the SVM models. The output of the function is a new dataset where a comparison can be made that could not have been made on the original non-linear dataset.

1. **Why are kernel-based algorithm methods preferred for non-linear SVM models? How does the Kernel-trick enable their benefits?**

Kernel-based algorithms are preferred because they act similarly regarding the polynomial features for regression models in that they reduce dataset scope and also optimize data underfitting and overfitting based on the defined input to the kernel algorithm

**Problem #4: Support Vector Machine**

Given the plot of a linear SVC’s training set, decision boundaries, and margin.

1. In the figure below showing the decision boundary, margin, and training set instances, **circle** all **support vector** instances.
2. In the figure, **put an X through** all **margin violations** (if any exist).
3. **Specify** the **predictions** given the SVC model in the table below of test data.
4. From your results for c, **complete** **the confusion matrix** below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street  Decision Boundary | **Test Dataset:**   |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | | 0.25 | 0. 5 | 0 | 0 | | 0.25 | 0.75 | 1 | 0 | | 0.5 | 1.0 | 1 | 1 | | 0.875 | 0.5 | 1 | 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted** | | |
| **Target** |  | **Negative (0)** | **Positive (1)** |
| **Negative (0)** | 1 | 0 |
| **Positive (1)** | 1 | 2 |

**Problem #5: Short-Answer – Theory and Concepts – Decision Trees**

Answer each question in complete sentences. Be sure to clearly justify your answer.

1. **Describe how the CART algorithm utilizes the impurity of a node’s children to guide the selection of the node’s decision attribute and threshold when training a decision tree.**

The CART algorithm which is an algorithm for finding the optimal feature and threshold of a binary balanced tree, which is where all non-leaf nodes have two child nodes one being the true condition and the other being the false condition. The CART algorithm optimizes two values by testing the purity at each level by use of the Gini impurity metric where the value is the output which is the ratio of classes among instances for the ith node of the tree. The CART then picks the lowest impurity value.

1. **What are the advantages of using Entropy vs. Gini as the impurity metric for a decision tree?**

Entropy and Gini are both options as the impurity metric for CART traversing a decision tree, however Entropy is typically the better option for better predictions with a more balanced tree, and when compared to Gini tends to not isolate itself in a branch

1. **Describe at least two techniques than can be utilized to regularize a decision tree. i.e. how do they impact the size/shape of the tree?**

Two techniques to the regularize a decision tree are min\_samples\_split and max\_leaf\_nodes.

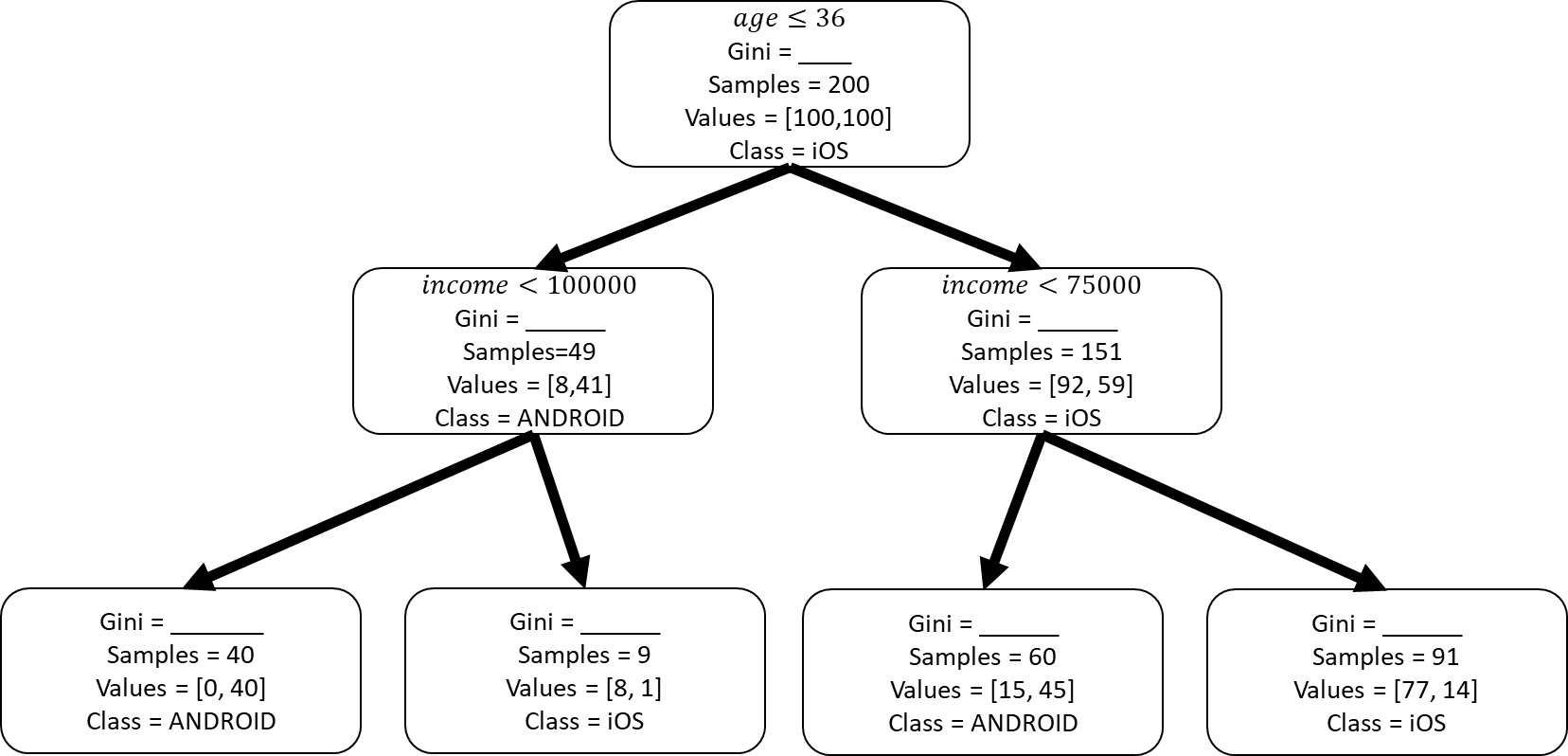
min\_sample\_splits is where in order for a node to split it must be a specified number of samples, this can have the effect of reducing the depth of the tree but forcing a node to have a larger amount of samples which restricts it from splitting or by reducing the amount of samples needed which can lead to much more nodes splitting as it would be easier to reach the sample count

max\_leaf\_nodes is where the tree has a max number of nodes, which can drastically affect the depth of the tree in that the nodes can be very limited or very unlimited depending on the max leaf nodes, because every non-leaf nodes must have two child nodes the number of options can change.

**Problem #6: Decision Tree**

Given the following decision, address the questions below.

1. **Calculate the Gini impurity for each of the decision tree nodes below.**

****

0.26

0.375

0.20

0.0

0.48

0.27

0.5

Note: for sample counts by values, the order is values[num\_ios, num\_android]

1. **For the following test set, compute the predictions and complete the confusion matrix.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | | 32 | $105,000 | iOS | iOS | | 19 | $25,000 | ANDROID | ANDROID | | 52 | $92,000 | ANDROID | iOS | | 35 | $123,000 | ANDROID | iOS | | |  |  |  |  | | --- | --- | --- | --- | |  | **Predicted** | | | | **Target** |  | **Negative (iOS)** | **Positive (ANDROID)** | | **Negative (IOS)** | 1 | 0 | | **Positive (ANDROID)** | 2 | 1 | |

**Problem #7: Short-Answer – Theory and Concepts – Ensemble Methods**

Answer each question in complete sentences. Be sure to clearly justify your answer.

1. **Compare and contrast Bagging vs. Boosting ensemble methods. How does each of these paradigms leverage the power of the ensemble?**

Bagging is used to reduce the variance of a decision tree, by means of taking averages of all the predictions from multiple trees and using those averages into one decision tree. Boosting is the means of creating a collection of prediction models, where when an input is misclassified by a hypothesis, it is weighted extra so that the next prediction is less likely to be misclassified again.

1. **What are “decision stump” and why are they sufficiently complex when used in a random forest classifier?**

Decision stumps are 1 level decision tree with the root node going to two leaf nodes as the only branches and nodes in the tree, these are useful in random forest classifier because you can use numerous stumps to create the “forest” of trees that can be randomly sampled.

1. **Describe out-of-bag validation and how it is used with bagging ensemble methods?**

Out of bag validation is comparing a selected group of values to the values that were not in the selected group of values, in bagging this could be used with the average tree being compared to the trees it was formed from, or compared to other trees that were not included in the averaged data

**Problem #8: Perceptron**

For **one training epoch (training each training input for one round),** **determine** the perceptron output , the error, and the **new weights and**  Use the table to capture your answers, but show your work.

|  |
| --- |
| **Initial model parameters and Learning Rate**  ,  (learning rate) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | **new** | **new** |
| 0 | 0 | 0 | 0 | 0 | 0.1 | -0.2 |
| 0 | 1 | 0 | 0 | 0 | 0.1 | -0.2 |
| 1 | 0 | 0 | 0 | 0 | 0.1 | -0.2 |
| 1 | 1 | 1 | 0 | 1 | 0.22 | -0.05 |

Show your work below…

**Iteration 1:**

(0.1\*0 + -0.2\*0 – 0.1) = y = -0.3 -> 0

(0 - 0) = e = 0

(0.1 + (0.1 \* 0 \* 0)) = new w1 = 0.1

(-0.2 + (0.1 \* 0 \* 0)) = new w2 = -0.2

**Iteration 2:**

(0.1\*0 + -0.2\*1 - 0.3) = y = -0.5 -> 0

(0 - 0) = e = 0

(0.1 + (0.1 \* 0 \* 0)) = new w1 = 0.1

(-0.2 + (0.1 \* 1 \* 0)) = new w2 = -0.2

**Iteration 3:**

(0.1\*1 + -0.2\*0 – 0.3) = y = -0.2 -> 0

(0 - 0) = e = 0

(0.1 + (0.1 \* 1 \* 0)) = new w1 = 0.1

(-0.2 + (0.1 \* 0 \* 0)) = new w2 = -0.2

**Iteration 4:**

(0.1\*1 + -0.2\*1 – 0.3) = y = -0.33 -> 0

(1 - 0) = e = 1

(0.12 + (0.1 \* 1 \* 1)) = new w1 = 0.22

(-0.15 + (0.1 \* 1 \* 1)) = new w2 = -0.05

**Problem #9: Short-Answer – Theory and Concepts – Neural Networks**

1. **Explain why the perceptron can be trained on two input binary features can be trained to successfully output the correct Boolean value for AND, OR, or NOT, but not XOR.**

Perceptrons work by making use of the equation w1\*x1 + w2\*x2 = θ, which no matter the inputs create a dividing line across the two regions, which is why for the AND, OR and NOT they are all able to be divided into two regions and can therefore be solved by perceptron, but XOR because of its exclusive nature can not be divided in to regions.

1. **Under the artificial neural network analogy with biological neural networks, match the biological neuron to its artificial neuron equivalent.**

|  |
| --- |
| **Artificial Neuron Terms**  Input, Output, Neuron, Weights |

**DENDRITES** \_\_\_\_\_\_Input\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SOMA** \_\_\_\_\_Neuron\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

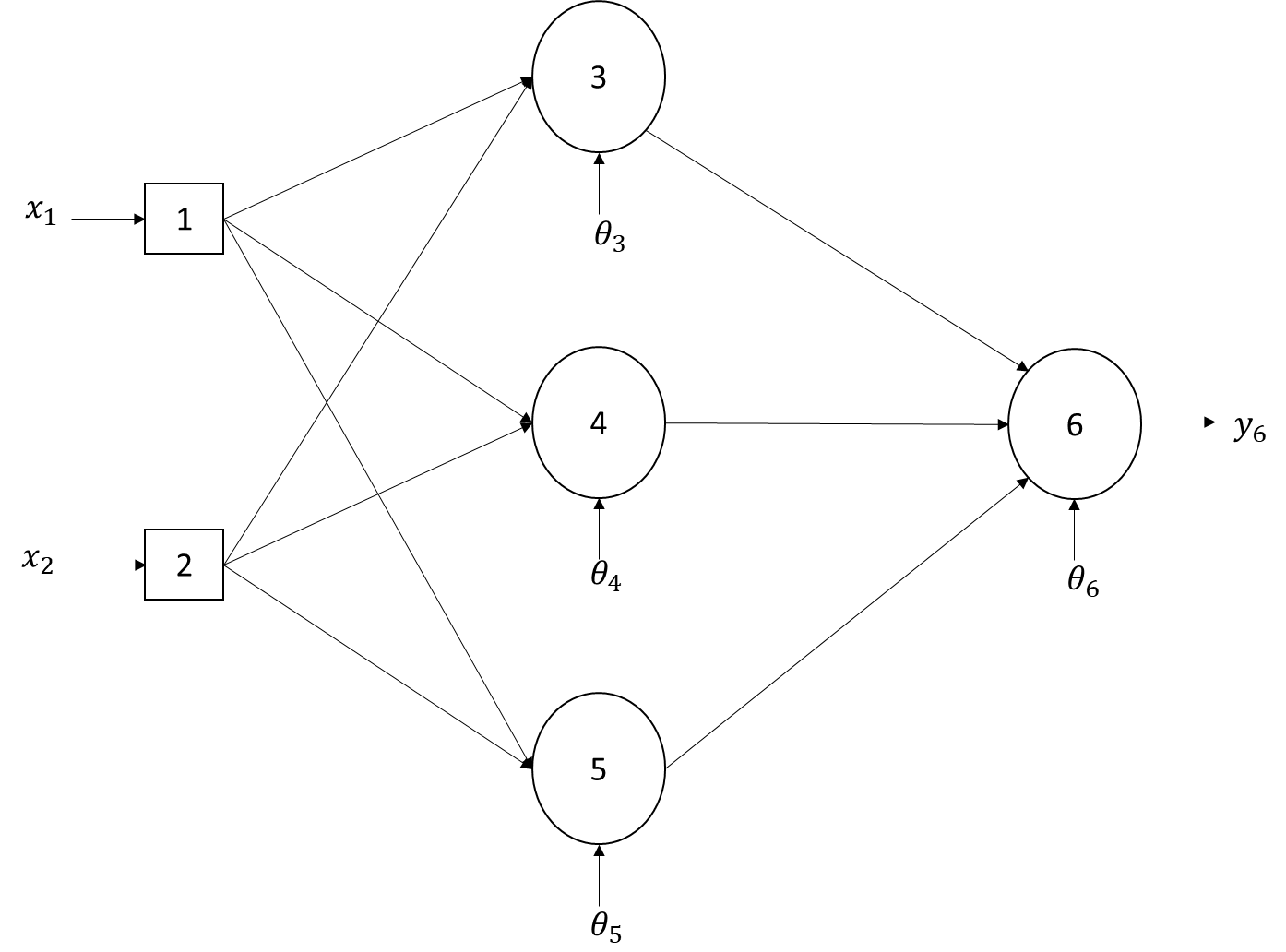
**SYNAPSE** \_\_\_\_Weight\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**AXON** \_\_\_\_Output \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Problem #10: Neural Network**

For the following artificial neural network, **determine the output** of the network () via feedforward given the input  **and** and the **error gradient for neurons 3, 4, 5, and 6** if the desired output of the network via back propagation. You do not need to calculate the change in weights for this problem. Show your work

|  |  |  |  |
| --- | --- | --- | --- |
| **Initial Model Parameters** | | | |
| **,**  **,** | **,**  **,** | **,**  **,** | **,**  **,**  **,** |

****

The sigmoid equation will be used as the activation function for all neurons. Let learning rate, .

Answers and work below…

0.1\*1 + (-0.2\*-1) – 0.1 = 0.2 = y3 -> y3 = sig(0.2) = 0.55

-0.3\*1 + (-0.2\*-1) – 0.3 = -0.4 = y4 -> y4 = sig(-0.4) = 0.4

0.1\*1 + (0.2\*-1) – 0.2 = -0.3 = y5 -> y5 = sig(-0.3) = 0.43

0.55\*0.1 + 0.4\*0.2 + 0.43\*0.2 - -0.1 = 0.321 = y6 -> y6 = sig(0.321) = 0.58

0.75 – 0.58 = 0.17 = e

0.58\*(1 – 0.58)\*0.17 = 0.04 = e6

0.43\*(1-0.43)\*0.04 = 0.0098 = e5

0.4\*(1-0.4)\*0.04 = 0.0096 = e4

0.55\*(1-0.55)\*0.04 = 0.0099 = e3