		Shen Yang	
	Ideal	Team: YS	
	Score	Score	.----------------------------- Instructor Notes-------------------------------------.
Total Points	10	8.9	Score Points:08.90
Total Percentage	100	89	
Exceptional Work: One idea (**Required 7000**): Implement an optimization technique for logistic regression using mean square error as your objective function (instead of binary entropy). Your solution should be able to solve the binary logistic regression problem in one gradient update step.	10	1	You are not calucalting the gradient for the MSE objective function and therefore not addressign the given problem. You would need to re-solve the gradient for an MSE, then use the new Hessian and new gradient. You implementation still uses cross_entropy (despite having an updated cost function).
Explain the task and what business-case or use-case it is designed to solve (or designed to investigate). Detail exactly what the classification task is and what parties would be interested in the results.	20	20	Ok. I like the classification task. Seems interesting, and almost impossible to perform well on.
Define and prepare your class variables. Use proper variable representations (int, float, one-hot, etc.). Use pre-processing methods (as needed) for dimensionality reduction, scaling, etc. Remove variables that are not needed/useful for the analysis. **Describe the final dataset** that is used for classification/regression (include a description of any newly formed variables you created).	5	3	City code and sport code should be one hot encoded. > The remaining numeric data should be normalized.
Divide you data into training and testing data using an 80% training and 20% testing split. Use the cross validation modules that are part of scikit-learn. **Argue for or against splitting your data using an 80/20 split.** That is, why is the 80/20 split appropriate (or not) for your dataset?	5	5	yes!!
Create a custom, one-versus-all logistic regression classifier using numpy and scipy to optimize. Use object oriented conventions identical to scikit-learn. You should start with the template developed by the instructor in the course. You should add the following functionality to the logistic regression classifier: --Ability to choose optimization technique when class is instantiated: either steepest descent, stochastic gradient descent, or Newton's method. --Update the gradient calculation to include a customizable regularization term (either using no regularization, L1 regularization, L2 regularization, or both L1 and L2 regularization). Associate a cost with the regularization term, "C", that can be adjusted when the class is instantiated.	20	20	Thanks for citing everything and explaining how you changed it.
Train your classifier to achieve good generalization performance. That is, adjust the optimization technique and the value of the regularization term "C" to achieve the best performance on your test set. **Visualize the performance of the classifier versus the parameters you investigated**. Is your method of selecting parameters justified? That is, do you think there is any "data snooping" involved with this method of selecting parameters?	15	15	
Compare the performance of your "best" logistic regression optimization procedure to the procedure used in scikit-learn. Visualize the performance differences in terms of training time and classification performance. Discuss the results.	15	15	
Which implementation of logistic regression would you advise be used in a deployed machine learning model, your implementation or scikit-learn (or other third party)? Why?	10	10	