Robert Krajanci
1666020

This file displays the results obtained from running each of the three machine learning algorithms 3 times.

Softmax Regression:

Results from running softmax.py three times

#1	#2	#3
Using learning rate: 0.3 Epoch that yields the best validation performance: 43 Validation performance (accuracy) in that epoch: 0.8272	Using learning rate: 0.3 Epoch that yields the best validation performance: 33 Validation performance (accuracy) in that epoch: 0.8369	Using learning rate: 0.3 Epoch that yields the best validation performance: 9 Validation performance (accuracy) in that epoch: 0.8309
Using learning rate: 0.1 Epoch that yields the best validation performance: 8 Validation performance (accuracy) in that epoch: 0.8247	Using learning rate: 0.1 Epoch that yields the best validation performance: 42 Validation performance (accuracy) in that epoch: 0.8297	Using learning rate: 0.1 Epoch that yields the best validation performance: 33 Validation performance (accuracy) in that epoch: 0.8296
Using learning rate: 0.03 Epoch that yields the best validation performance: 16 Validation performance (accuracy) in that epoch: 0.8182	Using learning rate: 0.03 Epoch that yields the best validation performance: 32 Validation performance (accuracy) in that epoch: 0.8361	Using learning rate: 0.03 Epoch that yields the best validation performance: 31 Validation performance (accuracy) in that epoch: 0.8291
Using learning rate: 0.01 Epoch that yields the best validation performance: 36 Validation performance (accuracy) in that epoch: 0.811	Using learning rate: 0.01 Epoch that yields the best validation performance: 43 Validation performance (accuracy) in that epoch: 0.8355	Using learning rate: 0.01 Epoch that yields the best validation performance: 42 Validation performance (accuracy) in that epoch: 0.8389
Using learning rate: 0.003 Epoch that yields the best validation performance: 43 Validation performance (accuracy) in that epoch: 0.8192	Using learning rate: 0.003 Epoch that yields the best validation performance: 41 Validation performance (accuracy) in that epoch: 0.8317	Using learning rate: 0.003 Epoch that yields the best validation performance: 41 Validation performance (accuracy) in that epoch: 0.8286
Using learning rate: 0.001 Epoch that yields the best validation performance: 45 Validation performance (accuracy) in that epoch: 0.8111	Using learning rate: 0.001 Epoch that yields the best validation performance: 35 Validation performance (accuracy) in that epoch: 0.8317	Using learning rate: 0.001 Epoch that yields the best validation performance: 49 Validation performance (accuracy) in that epoch: 0.8352

Using learning rate: 0.0003 Epoch that yields the best validation performance: 18 Validation performance (accuracy) in that epoch: 0.8098 Using learning rate: 0.0001 Epoch that yields the best validation performance: 38 Validation performance (accuracy) in that epoch: 0.779 Best alpha value based on hyperparameter tuning: 0.3 **Test performance (accuracy) on testing data: 0.8183**	Using learning rate: 0.0003 Epoch that yields the best validation performance: 44 Validation performance (accuracy) in that epoch: 0.8159 Using learning rate: 0.0001 Epoch that yields the best validation performance: 47 Validation performance (accuracy) in that epoch: 0.7929 Best alpha value based on hyperparameter tuning: 0.3 **Test performance (accuracy) on testing data: 0.8206**	Using learning rate: 0.0003 Epoch that yields the best validation performance: 47 Validation performance (accuracy) in that epoch: 0.8192 Using learning rate: 0.0001 Epoch that yields the best validation performance: 43 Validation performance (accuracy) in that epoch: 0.7853 Best alpha value based on hyperparameter tuning: 0.01 **Test performance (accuracy) on testing data: 0.8269**

Robert Krajanci
1666020

One-vs-All Logistic Regression:
Results from running logistic regression.py three times

#1	#2	#3
Using learning rate: 0.3 Epoch that yields the best validation performance: 31 Validation performance (accuracy) in that epoch: 0.8302	Using learning rate: 0.3 Epoch that yields the best validation performance: 5 Validation performance (accuracy) in that epoch: 0.8144	Using learning rate: 0.3 Epoch that yields the best validation performance: 21 Validation performance (accuracy) in that epoch: 0.8224
Using learning rate: 0.1 Epoch that yields the best validation performance: 18 Validation performance (accuracy) in that epoch: 0.8265	Using learning rate: 0.1 Epoch that yields the best validation performance: 27 Validation performance (accuracy) in that epoch: 0.8108	Using learning rate: 0.1 Epoch that yields the best validation performance: 16 Validation performance (accuracy) in that epoch: 0.8111
Using learning rate: 0.03 Epoch that yields the best validation performance: 19 Validation performance (accuracy) in that epoch: 0.8303	Using learning rate: 0.03 Epoch that yields the best validation performance: 41 Validation performance (accuracy) in that epoch: 0.8085	Using learning rate: 0.03 Epoch that yields the best validation performance: 10 Validation performance (accuracy) in that epoch: 0.8166
Using learning rate: 0.01 Epoch that yields the best validation performance: 37 Validation performance (accuracy) in that epoch: 0.8261	Using learning rate: 0.01 Epoch that yields the best validation performance: 48 Validation performance (accuracy) in that epoch: 0.8125	Using learning rate: 0.01 Epoch that yields the best validation performance: 33 Validation performance (accuracy) in that epoch: 0.8224
Using learning rate: 0.003 Epoch that yields the best validation performance: 24 Validation performance (accuracy) in that epoch: 0.8363	Using learning rate: 0.003 Epoch that yields the best validation performance: 48 Validation performance (accuracy) in that epoch: 0.8111	Using learning rate: 0.003 Epoch that yields the best validation performance: 12 Validation performance (accuracy) in that epoch: 0.8435
Using learning rate: 0.001 Epoch that yields the best validation performance: 49 Validation performance (accuracy) in that epoch: 0.8507	Using learning rate: 0.001 Epoch that yields the best validation performance: 26 Validation performance (accuracy) in that epoch: 0.838	Using learning rate: 0.001 Epoch that yields the best validation performance: 38 Validation performance (accuracy) in that epoch: 0.8526
Using learning rate: 0.0003	Using learning rate: 0.0003 Epoch that yields the best	Using learning rate: 0.0003 Epoch that yields the best

Epoch that yields the best validation performance: 47 Validation performance (accuracy) in that epoch: 0.8514 Using learning rate: 0.0001 Epoch that yields the best validation performance: 49 Validation performance (accuracy) in that epoch: 0.8435 Best alpha value based on hyperparameter tuning: 0.0003 **Test performance (accuracy) on testing data: 0.8371**	validation performance: 48 Validation performance (accuracy) in that epoch: 0.8465 Using learning rate: 0.0001 Epoch that yields the best validation performance: 49 Validation performance (accuracy) in that epoch: 0.8354 Best alpha value based on hyperparameter tuning: 0.0003 **Test performance (accuracy) on testing data: 0.8359**	validation performance: 46 Validation performance (accuracy) in that epoch: 0.8567 Using learning rate: 0.0001 Epoch that yields the best validation performance: 48 Validation performance (accuracy) in that epoch: 0.8455 Best alpha value based on hyperparameter tuning: 0.0003 **Test performance (accuracy) on testing data: 0.835**

Robert Krajanci
1666020

Gaussian Naive Bayes Method:

Results from running naive-bayes.py three times

#1	#2	#3
Validation accuracy using equal prior for all categories : 0.7015 Validation accuracy using priors based on training data : 0.1036 The best prior is: using equal prior for all categories **Test performance (accuracy): 0.696**	Validation accuracy using equal prior for all categories : 0.7004 Validation accuracy using priors based on training data : 0.1018 The best prior is: using equal prior for all categories **Test performance (accuracy): 0.6948**	Validation accuracy using equal prior for all categories : 0.7036 Validation accuracy using priors based on training data : 0.0981 The best prior is: using equal prior for all categories **Test performance (accuracy): 0.6961**