Matt Hyatt 00001555727

COMP 379 HW2

My dataset was the 'banknote dataset' I found from www.kaggle.com. It contains various feature of banknotes and a target value of authentic (1) or inauthentic (0). I decided to measure my classifiers with f1 score since it seemed important that the classifier should be equally accurate at predicting both classes.

I used the logistic regression class from scikit-learn to train my binary classifier. It averaged about a 0.98 f1 score when used with default parameters. I used nested for loops to iterate through all solver and penalty hyperparameters as well as [0.001, 0.01, 0.1, 1, 10, 100, 1000] C regularization values. I found that a given solver penalty pair generally did not see further improvements on its f1 score if the C value was above 1. In other words C=1000 performed as well as C=1 or saw marginal differences. L2 models performed better than elasticnet models which performed better than L1 models. Different solvers did not affect the performance of the model. I ended up choosing an elasticnet (L1 ratio=0.5) model over an L2 model anyways since it made more sense conceptually to balance what L1 and L2 were trying to achieve and the performance between L2 and elasticnet was negligible at higher C values.

I developed my KNN algorithm by using pandas DataFrames. Since the distance between two points is $\sqrt{(A^2 + B^2 + \ldots)}$ I calculated the distance of all points in the training set from the given datapoint by subtracting it from the test set with respect to each dimension, squaring those values, and summing them; distance = $\sqrt{((x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2 + (w1-w2)^2)}$. I picked the k class labels from the training set with the lowest distance, and the most prevalent class among them became the predicted class for that datapoint. I used the development set to optimize k and found that my model had the highest f1 score (between 0.99 and 1.0 inclusive) when choosing only one nearest neighbor; k = 1.

On the test set, the knn algorithm maintains 1.0 f1 score most of the time and the logistic classifier with elasticnet penalty and C=1 averaged an f1 score of about 0.98. Both of these models performs significantly better than the baseline DummyClassifier. I used a for loop to try all the dummy strategies. The most accurate was the constant strategy with an f1 score of 0.62. The least accurate were most frequent and prior which scored 0.0.

Performance of classifiers shows the performance of various models: knn is orange, elasticnet is green, L2 is blue, L1 is red, dummy classifiers are purple. Performance of KNN shows the KNN's performance at different K values.