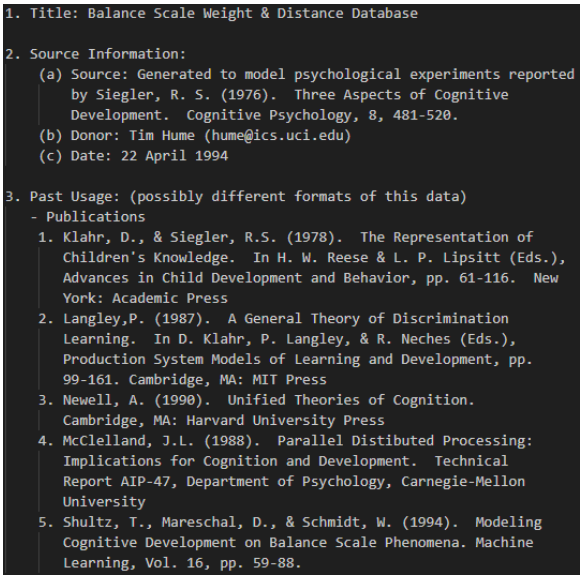
CMPUT 466 Machine Learning

Project Report

Dataset Introduction



```
1. Title: Balance Scale Weight & Distance Database

2. Source Information:
   (a) Source: Generated to model psychological experiments reported
       by Siegler, R. S. (1976).  Three Aspects of Cognitive
       Development.  Cognitive Psychology, 8, 481-520.
   (b) Donor: Tim Hume (hume@ics.uci.edu)
   (c) Date: 22 April 1994

3. Past Usage: (possibly different formats of this data)
   - Publications
     1. Klahr, D., & Siegler, R.S. (1978).  The Representation of
        Children's Knowledge.  In H. W. Reese & L. P. Lipsitt (Eds.),
        Advances in Child Development and Behavior, pp. 61-116.  New
        York: Academic Press
     2. Langley,P. (1987).  A General Theory of Discrimination
        Learning.  In D. Klahr, P. Langley, & R. Neches (Eds.),
        Production System Models of Learning and Development, pp.
        99-161. Cambridge, MA: MIT Press
     3. Newell, A. (1990).  Unified Theories of Cognition.
        Cambridge, MA: Harvard University Press
     4. McClelland, J.L. (1988).  Parallel Distibuted Processing:
        Implications for Cognition and Development.  Technical
        Report AIP-47, Department of Psychology, Carnegie-Mellon
        University
     5. Shultz, T., Mareschal, D., & Schmidt, W. (1994).  Modeling
        Cognitive Development on Balance Scale Phenomena. Machine
        Learning, Vol. 16, pp. 59-88.
```

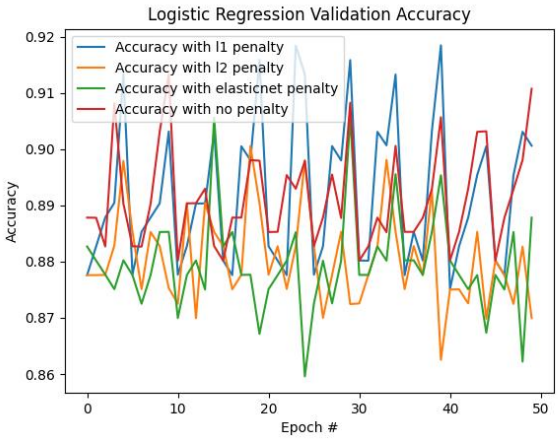
The balance scale database, data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.

The three algorithms we applied to classify the dataset are Logistic Regression with SGDClassifier, SVM with SVC, and KNN neighbors.

Project Introduction

In this project, we applied the systematic hyperparameter adjustment method to implement the train, test, validation infrastructure, We use cross_val_score through the whole project in order to find the cross validation accuracy. The goal of the project is to optimize the training accuracy by finding the best-fitting hyperparameters. Then we evaluates the performance of the trained model on test set for the test accuracy. At last, we obtain numerical and graphical results for accuracy with Logistic Regression, accuracy & loss & mean squared loss on SVM and KNN.

Logistic Regression



Logistic Regression is a widely used machine learning method, with an example input being fed in, the input with its weights being added together to obtain z value, $z = w_1x_1 + w_2x_2 + \cdots + w_mx_m + b = w^Tx + b$. Then it's fed into the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$. We tested from alpha 0.1 with it divided by 5 for next loops until it reaches 0.0001, and based on different alpha numbers, we train the model with SGDClassifier with penalties set as [l1, l2, elasticnet,

none], we can tell from the image that accuracy is maximized under l1 regularization. The results are reported below.

Logistic Regression Report:

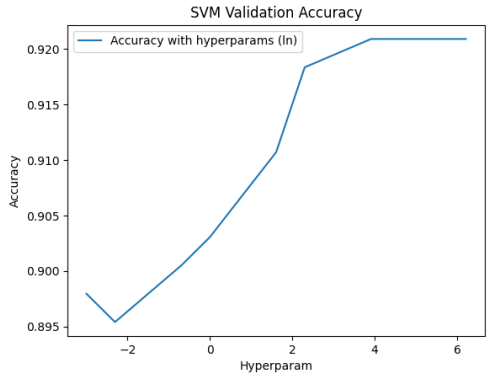
Best Penalty: l1 Best Alpha: 0.00016 Best Accu: 0.9184680298604351 Test Accu: 0.8882978723404256

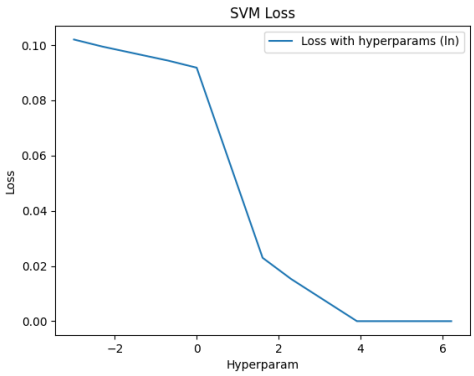
SVM

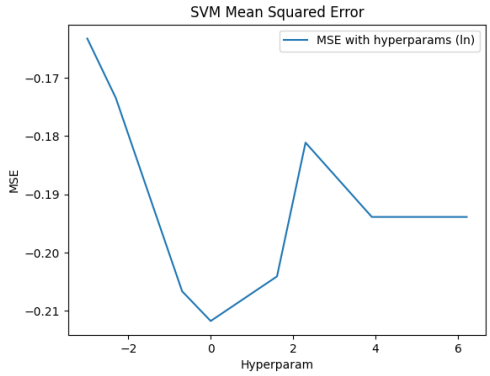
SVM maps feature vector to some points in the space, aiming at figuring out a line that separates types of points. To carry out the train under SVM, we made tuning hyperparameters as set of [0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500], the higher the hyperparameter is, the lower intensity of the regularization. We construct the SVM models under these hyperparameters and obtain numerical results with graphical accuracy & loss & mean squared loss. (hyperparameters are token log values in graphing)

CVM Report:

Best Hyperparam: 50 Best Accu: 0.9209183673469389 Test Accu: 0.9308510638297872







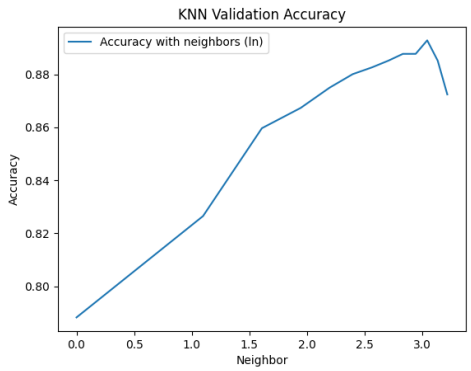
KNN neighbors

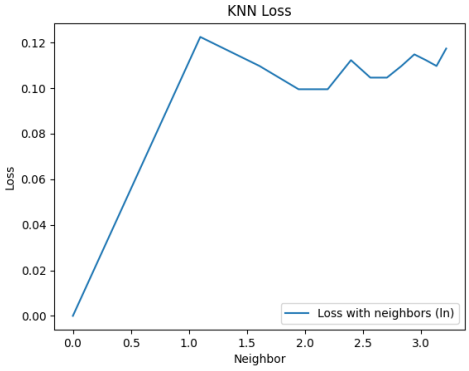
The KNN nearest neighbor classification algorithm takes all known samples as a reference, and classify the output value by calculating the distance between the known samples and the input value, and take all known samples within a certain distance into consideration, then input is classified. we made tuning hyperparameters neighbors as [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25] (take odd numbers is for avoiding ties). We construct the KNN neighbors models under these hyperparameters and obtain numerical results with graphical accuracy & loss & mean squared loss. (hyperparameters are token log values

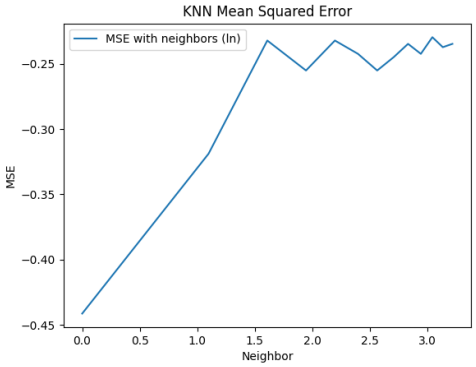
in graphing)

KNN Report:

Best Neighbor: 21 Best Accu: 0.8928571428571429 Test Accu: 0.8882978723404256







Result

Our report demonstrates an average accuracy around 90%. With Logistic Regression, we observe that accuracy is maximized under l1 regularization with alpha 0.00016. With SVM and KNN neighbors classification, accuracy increases as the hyperparameter increase but getting to a peak after reaching the certain values (turning points). With KNN algorithm, loss and mean squared error increases with increasing in the hyperparameter, With SVM algorithm, loss and mean squared error generally decreases with increasing in the hyperparameter.