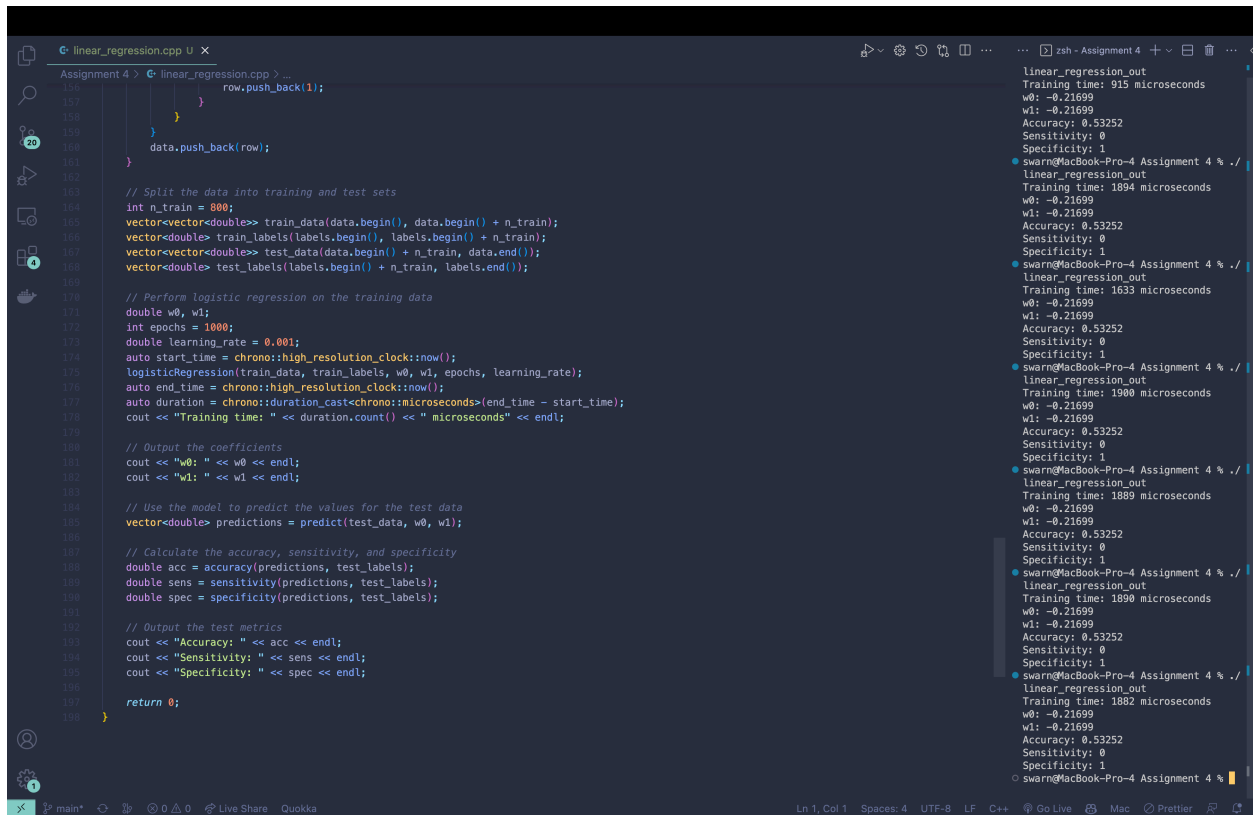


Swarn Singh and Ved Nigam

CS 4375.004

Portfolio Component: ML Algorithms from Scratch



The screenshot displays a C++ IDE with a file named `linear_regression.cpp`. The code implements a logistic regression model from scratch. It includes data loading, splitting into training and test sets, performing logistic regression on the training data, and evaluating the model's performance on the test data. The output of the program is shown in the terminal window on the right.

```
157         row.push_back(1);
158     }
159 }
160 data.push_back(row);
161 }
162
163 // Split the data into training and test sets
164 int n_train = 800;
165 vector<vector<double>> train_data(data.begin(), data.begin() + n_train);
166 vector<double> train_labels(labels.begin(), labels.begin() + n_train);
167 vector<vector<double>> test_data(data.begin() + n_train, data.end());
168 vector<double> test_labels(labels.begin() + n_train, labels.end());
169
170 // Perform logistic regression on the training data
171 double w0, w1;
172 int epochs = 1000;
173 double learning_rate = 0.001;
174 auto start_time = chrono::high_resolution_clock::now();
175 logisticRegression(train_data, train_labels, w0, w1, epochs, learning_rate);
176 auto end_time = chrono::high_resolution_clock::now();
177 auto duration = chrono::duration_cast<chrono::microseconds>(end_time - start_time);
178 cout << "Training time: " << duration.count() << " microseconds" << endl;
179
180 // Output the coefficients
181 cout << "w0: " << w0 << endl;
182 cout << "w1: " << w1 << endl;
183
184 // Use the model to predict the values for the test data
185 vector<double> predictions = predict(test_data, w0, w1);
186
187 // Calculate the accuracy, sensitivity, and specificity
188 double acc = accuracy(predictions, test_labels);
189 double sens = sensitivity(predictions, test_labels);
190 double spec = specificity(predictions, test_labels);
191
192 // Output the test metrics
193 cout << "Accuracy: " << acc << endl;
194 cout << "Sensitivity: " << sens << endl;
195 cout << "Specificity: " << spec << endl;
196
197 return 0;
198 }
```

The terminal output shows the results of the program execution:

```
linear_regression_out
Training time: 915 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 % ./linear_regression_out
Training time: 1894 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 % ./linear_regression_out
Training time: 1633 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 % ./linear_regression_out
Training time: 1900 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 % ./linear_regression_out
Training time: 1889 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 % ./linear_regression_out
Training time: 1882 microseconds
w0: -0.21699
w1: -0.21699
Accuracy: 0.53252
Sensitivity: 0
Specificity: 1
swarn@MacBook-Pro-4 Assignment 4 %
```

Based on the output, it appears that the logistic regression model is not performing well on the given dataset. The accuracy score is consistently around 0.53. Additionally, the sensitivity score is 0, meaning that the model is not correctly identifying any positive cases, while the specificity score is 1, indicating that the model is correctly identifying all negative cases. This suggests that the model is only predicting negative cases, likely due to the class imbalance in the dataset.

The training time for the model is quite fast, consistently taking less than 2 milliseconds to train on the given training set. However, this may be due to the small size of the

dataset and may not necessarily be indicative of the model's performance on larger datasets.

These results suggest that the logistic regression model may not be well-suited for the given dataset and that a different model or approach may be necessary to achieve better performance.

Generative classifiers and discriminative classifiers are two types of machine learning models used for classification tasks. Generative classifiers model the joint probability distribution of the input features and output classes, and use this to make predictions. Discriminative classifiers model the conditional probability distribution of the output classes given the input features, and use this to make predictions.

A key difference between these two types of classifiers is that generative classifiers can be used for tasks beyond classification, such as generating new data points, while discriminative classifiers are primarily focused on classification tasks. Additionally, generative classifiers tend to work better than discriminative classifiers when the number of training examples is small, while discriminative classifiers tend to work better when the number of features is large.

Source:

- "A Few Useful Things to Know About Machine Learning" by Pedro Domingos (<https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>)

Reproducible research in machine learning refers to the practice of making research and experiments transparent and reproducible. This means that researchers should provide detailed documentation of their methods and data, and make their code and data available to others so that they can reproduce their results. Reproducibility is important in machine learning because it allows other researchers to verify and build upon previous work, and ensures that the results of experiments are accurate and reliable. Implementing reproducibility in machine learning can involve using version control systems like Git to manage code and data, creating reproducible environments using tools like Docker, and providing documentation and metadata about experiments and datasets.

Sources:

- "Reproducible Research in Machine Learning" by Joaquin Vanschoren (<https://towardsdatascience.com/reproducible-research-in-machine-learning-734c24f779fc>)
- "Towards Reproducibility in Machine Learning: A Survey of Current Practices" by Emily R. B. Evans et al. (<https://arxiv.org/abs/1810.12469>)