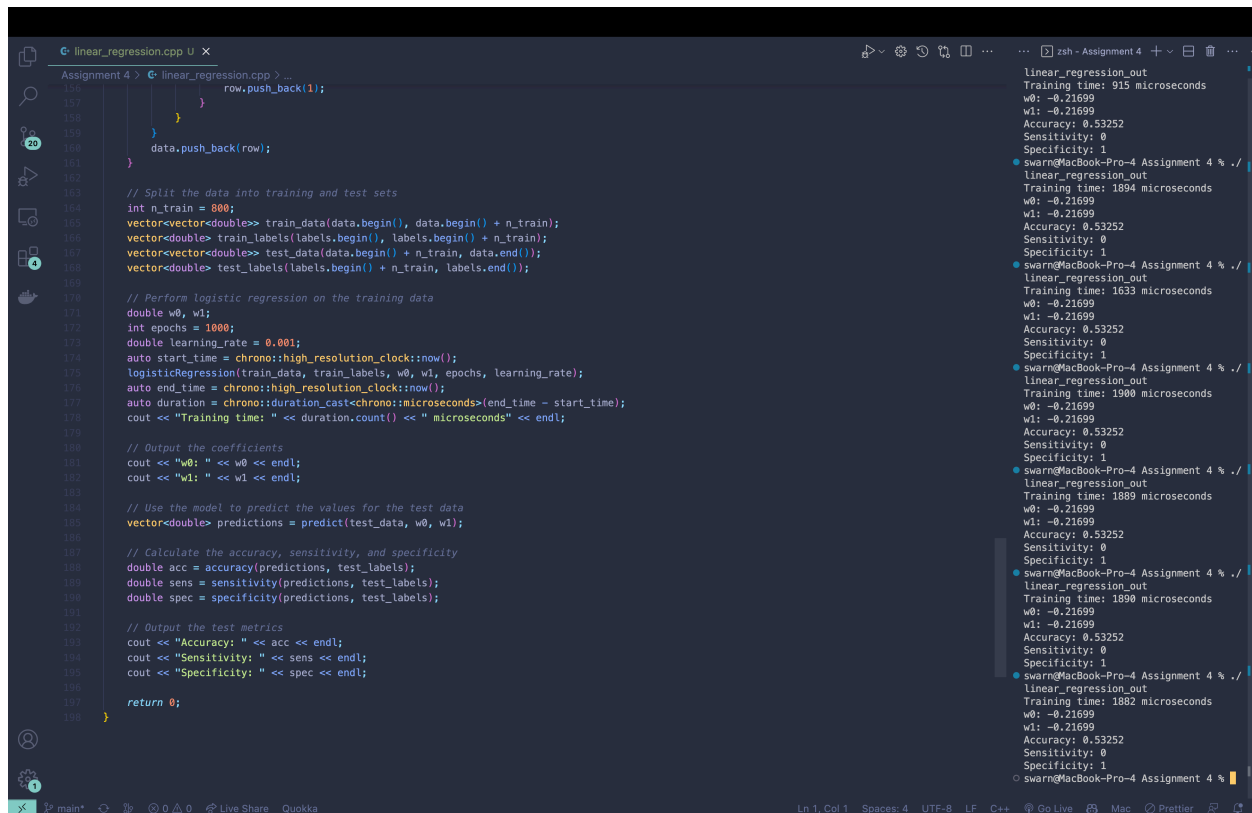


Swarn Singh and Ved Nigam

CS 4375.004

Portfolio Component: ML Algorithms from Scratch

## Performance of Models:



```
113 linear_regression.cpp U x
114 Assignment 4 > linear_regression.cpp > ...
115     row.push_back(1);
116 }
117 }
118 data.push_back(row);
119 }
120
121 // Split the data into training and test sets
122 int n_train = 800;
123 vector<vector<double>> train_data(data.begin(), data.begin() + n_train);
124 vector<double> train_labels(labels.begin(), labels.begin() + n_train);
125 vector<vector<double>> test_data(data.begin() + n_train, data.end());
126 vector<double> test_labels(labels.begin() + n_train, labels.end());
127
128 // Perform logistic regression on the training data
129 double w0, w1;
130 int epochs = 1000;
131 double learning_rate = 0.001;
132 auto start_time = chrono::high_resolution_clock::now();
133 logisticRegression(train_data, train_labels, w0, w1, epochs, learning_rate);
134 auto end_time = chrono::high_resolution_clock::now();
135 auto duration = chrono::duration_cast<chrono::microseconds>(end_time - start_time);
136 cout << "Training time: " << duration.count() << " microseconds" << endl;
137
138 // Output the coefficients
139 cout << "w0: " << w0 << endl;
140 cout << "w1: " << w1 << endl;
141
142 // Use the model to predict the values for the test data
143 vector<double> predictions = predict(test_data, w0, w1);
144
145 // Calculate the accuracy, sensitivity, and specificity
146 double acc = accuracy(predictions, test_labels);
147 double sens = sensitivity(predictions, test_labels);
148 double spec = specificity(predictions, test_labels);
149
150 // Output the test metrics
151 cout << "Accuracy: " << acc << endl;
152 cout << "Sensitivity: " << sens << endl;
153 cout << "Specificity: " << spec << endl;
154
155 return 0;
156 }
```

```
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: 1980 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: 1890 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.

```
passenger class likelihood matrix
```

```
0.269231 0.352564 0.942308
```

```
0.266393 0.168033 0.204918
```

```
sex likelihood matrix
```

```
0.25 0.840164
```

```
0.679487 0.204918
```

```
means:
```

```
18.5388 11.235
```

```
variances:
```

```
0.513056 0.502574
```

We were not able to complete the Naïve-Bayes model on time, but we did complete a lot of the components that go into the Naïve-Bayes model. From previous models, we know that a Naïve-Bayes model predicts conditional probabilities: the effect of a predictor on the target variable. What is also special about this model is that it implements disjoint probability so that the effect of a predictor is not dependent upon the effect of another predictor in the model. Due to the predictors being disjoint, this would make for a more efficient algorithm at the cost of accuracy. It will be less accurate because it is treating the predictors independently and not considering how the predictors affect the model together.

### **Generative Classifiers vs Discriminative Classifiers:**

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.

### **Reproducible Research in Machine Learning:**

Reproducible research is a growing discussion in the computational sciences academic community because it has begun to question the credibility of the work (LeVeque). A lot of the work that is being done in the computational sciences realm can only be reproduced with certain conditions and certain types of data. With such restrictions and vast sources of data, it is difficult to be sure on how accurate and untampered the data is. Eventually, there are two sides to this discussion: those who think data should be easily accessible by individuals and those who believe that it sharing code/data is a concern.

An application of Machine Learning and Artificial Intelligence is in the medical field and how recent technologies can be used to identify tumors and models can be used to find trends in electronic health records. All of this comes with the breach of patient privacy (Carter). The authors of this article believe that reproducibility is extremely important because all of the medical field deserves to have the same tools. In the process of sharing these tools, the doctors/scientists are required to reveal their data source (patients) which is a question of the breach of privacy. This group of authors believe that the scientific community needs to establish some rules for sharing certain data because it is integral that members of the community share

their findings. By sharing their findings, they hope to be more efficient at diagnosing patients and enhancement in medical sciences.

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>)
- "A Few Useful Things to Know About Machine Learning" by Pedro Domingos  
(<https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>)
- "Reproducible Research for Scientific Computing: Tools and Strategies for Changing the Culture" by Randall J. LeVeque, Ian M. Mitchell, and Victoria Stodden  
(<https://staff.washington.edu/rjl/pubs/cise12/CiSE12.pdf>)

- “Pragmatic Considerations for Fostering Reproducible Research in Artificial Intelligence”

by Ricket E. Carter, Schi l. Attia, Francisco Lopez-Jimenez, and Paul A. Friedman

(<https://www.nature.com/articles/s41746-019-0120-2>)