Assignment 2 Write-Up

The dataset used in the analysis comes from the UCI Machine Learning Repository ([https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set#](https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set)). The data represents information collected from rural areas of Bangladesh about pregnant women. The women are classified based on the maternal risk level of their pregnancy. This is distinct from fetal risk level as it specifically refers to the potential for adverse effects for the mother. This data was collected in the context of the Internet of Things and was intended to help the United Nations accomplish the goals of “improving maternal health” and “reducing maternal and child mortality by 2030” (M. Ahmed).

The included risk factors are the mother’s age, blood pressure (systolic and diastolic), blood glucose level, and resting heart rate. The machine learning problem is to use these factors to predict which risk level (low, medium, or high) the mother belongs in.

Any classification algorithm is potentially suitable for this task. I chose to use a random forest as it is a robust method that minimizes the risk of overfitting while having strong predictive abilities. I also chose this method as it is a personal favorite of mine. A moderately sized forest was used with 500 trees and maximum depth of 14. The holdout method of validation was used with a 70-30 split. The model was able to achieve accuracy of 92.5% on the training data and accuracy of 82.13% on the testing set.

Here is some sample data from the testing set:

+-----+----------+

| label | prediction |

+-----+----------+

| 0.0| 0.0|

| 2.0| 1.0|

| 2.0| 2.0|

| 1.0| 1.0|

| 0.0| 0.0|

| 0.0| 1.0|

| 0.0| 0.0|

| 1.0| 0.0|

| 2.0| 2.0|

| 0.0| 0.0|

| 2.0| 2.0|

Note that when the model is incorrect, it is off by a factor of one and not two. This seems to indicate that the model is making reasonable predictions in the context of the problem.

The testing results were streamed in one file at a time with about 12 records per file. I chose to wait 5 seconds between loads so I could ensure that the streaming was working as intended. I have included some sample output from the streaming section below.

Current length of stream file: 12

Last 20 values read in:

Out[389]: [Row(label=0.0, probability=DenseVector([0.8382, 0.1598, 0.002]), prediction=0.0),

Row(label=2.0, probability=DenseVector([0.0028, 0.8118, 0.1853]), prediction=1.0),

Row(label=2.0, probability=DenseVector([0.0036, 0.0071, 0.9893]), prediction=2.0),

Row(label=1.0, probability=DenseVector([0.2001, 0.6097, 0.1902]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.8704, 0.0835, 0.0461]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.3799, 0.6158, 0.0044]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.6785, 0.3009, 0.0207]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.9097, 0.0759, 0.0144]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.9549, 0.0345, 0.0106]), prediction=0.0),

Row(label=1.0, probability=DenseVector([0.5837, 0.4093, 0.007]), prediction=0.0),

Row(label=2.0, probability=DenseVector([0.0043, 0.006, 0.9896]), prediction=2.0),

Row(label=0.0, probability=DenseVector([0.835, 0.159, 0.006]), prediction=0.0)]

Current length of stream file: 48

Last 20 values read in:

Out[390]: [Row(label=2.0, probability=DenseVector([0.0102, 0.0224, 0.9674]), prediction=2.0),

Row(label=1.0, probability=DenseVector([0.2328, 0.7345, 0.0326]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.5123, 0.482, 0.0057]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.4659, 0.5337, 0.0004]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.5275, 0.469, 0.0035]), prediction=0.0),

Row(label=2.0, probability=DenseVector([0.006, 0.1246, 0.8694]), prediction=2.0),

Row(label=0.0, probability=DenseVector([0.8948, 0.1048, 0.0004]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.7879, 0.2071, 0.005]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.9226, 0.0714, 0.006]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.6236, 0.3757, 0.0007]), prediction=0.0),

Row(label=1.0, probability=DenseVector([0.2522, 0.7466, 0.0012]), prediction=1.0),

Row(label=1.0, probability=DenseVector([0.3195, 0.6671, 0.0134]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.7793, 0.2072, 0.0135]), prediction=0.0),

Row(label=1.0, probability=DenseVector([0.1889, 0.6627, 0.1484]), prediction=1.0),

Row(label=1.0, probability=DenseVector([0.103, 0.1709, 0.726]), prediction=2.0),

Row(label=0.0, probability=DenseVector([0.8702, 0.1289, 0.0009]), prediction=0.0),

Row(label=1.0, probability=DenseVector([0.0136, 0.984, 0.0024]), prediction=1.0),

Row(label=0.0, probability=DenseVector([0.9266, 0.0654, 0.008]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.8214, 0.1703, 0.0083]), prediction=0.0),

Row(label=0.0, probability=DenseVector([0.8501, 0.1456, 0.0043]), prediction=0.0)]

Lastly, the data cleaning for this project was very straightforward. All the provided categories were already numeric and could be cast from integers to floating point numbers in the schema. The outcome variable was categorized using a string indexer inside the pipeline. The spark machine learning library made the data processing and fitting of the model quite easy.

The only problems encountered were relatively minor issues that were self-inflicted. They were:

1. Getting the random forest to perform multi-category classification (I was including the argument “max\_bins” which was causing it to not work)
2. Getting the streaming to write to a table (I was forcing the program to wait and this was causing me to be unable to query the table)

Works Cited

Ahmed, Marzia. "Maternal Health Risk Data Set." UCI Machine Learning Repository, University of California, Irvine, 21 Oct. 2009, [https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set#](https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set). Accessed 11 Mar. 2023.

M. Ahmed and M. A. Kashem, "IoT Based Risk Level Prediction Model For Maternal Health Care In The Context Of Bangladesh," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2020, pp. 1-6, doi: 10.1109/STI50764.2020.9350320.