RISK IDENT

CODING CHALLENGE

Imagine you are tasked with implementing a component that issues predictions of machine learning models on a stream of incoming data.

The stream of incoming data records for prediction consists of elements each of which have a feature vector with 5 entries (x1 - x5) which are assumed to be real numbers, and in addition a class label (either 1.0 or 0.0)

At any point in time over its life cycle, the predictor should be able to make predictions, i.e. have a model available. Furthermore, it should be able to switch to using an updated model without being restarted. To accomplish this, the predictor is listening on a second stream consisting of machine learning models. As soon as a new model is available in the models stream, the component should stop predicting, switch over to the new model and then resume predicting using the new model (to simplify the setup, you can assume, that the predictor polls the model stream frequently, e.g. after every 100 input records it's processed, but more general solutions are welcome).

You can assume the stream of models to consist of five logistic regression models with weights w1 through w5 and bias b

w1: 6.08, w2: 7.78, w3: 6.34, w4: 8.05, w5: 3.14, b: 61.35
w1: 8.46, w2: 1.74, w3: 6.08, w4: 4.25, w5: 1.92, b: 71.37
w1: 6.53, w2: 5.46, w3: 0.0, w4: 9.95, w5: 6.29, b: 43.3
w1: 3.2, w2: 7.32, w3: 1.46, w4: 2.29, w5: 4.26, b: 94.81
w1: 2.71, w2: 0.82, w3: 8.54, w4: 0.21, w5: 2.1, b: 66.25

In addition, using the labels which are part of the stream of input records, the component should compute a running update of the accuracy of the classifier and log this performance metric after every 10 predictions.

Your task for the challenge:

- Implement the component in either Python or Scala
- Feel free to introduce simplifying assumptions as you see fit
- Choose appropriate representations for the incoming data points and the models
- Test the component with a model stream containing the models specified above and a randomly generated stream of input data records
- Sketchy implementations are OK, working code is great, we care most about concept & design