資料分析

Final Project

題目：

MNIST WITH KNIME

組員(Cover Page)：

00557103游俊弘

00657207林弈呈

00457131丁兆文

## ㄧ、Abstract

本次的資料集使用的是Kaggle上Kannada MNIST資料集中的train.csv，其資料內容是使用數字0~9用手寫而成的28x28灰階影像(總共784pixels)，每個像素值都介在0~255之間，像素值越高，則代表越暗。

訓練資料集(train.scv)中，總共有60000筆(列)資料，有785行，第一行為label標籤，其餘為784個像素欄位。針對每個手寫數字，正確的辨識出實際的答案。

本次專案針對train.csv做研究，針對60000筆資料，隨機挑選出1200筆進行分析，將1200筆中，50%進行訓練，50%進行測試，使用SVM進行分析，並使用cross validation進行驗證(validation)，再利用ROC curve直觀地看出此研究分析的準確性。

本次報告希望利用課程所學到的分析技巧，knime的I/O、分群、測試訓練、分析等等......，多加利用並融會貫通，達到合理的正確率。

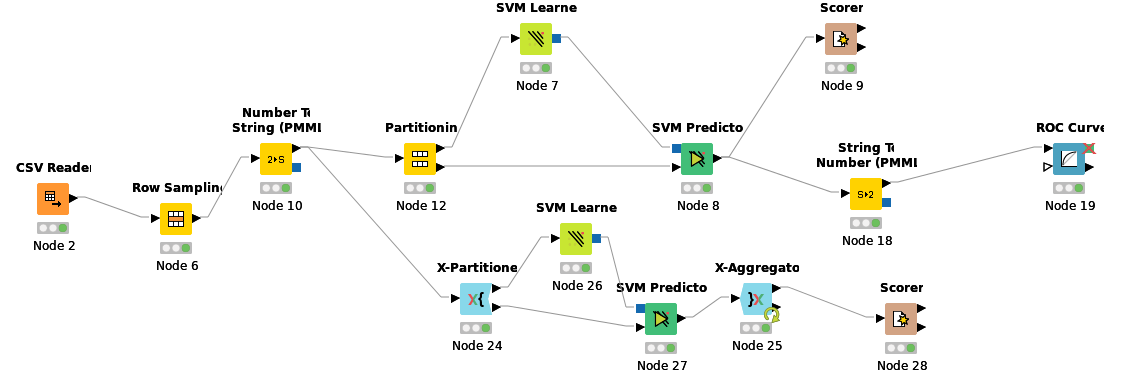
## 二、Introduction

一開始讀入csv檔時，會遇到"Encountered duplicate row ID "0""的問題，不過實際上之前作業就曾經出現過了，所以解決起來相對容易些，只要讀入時別勾選row header即可解決。

起初在隨機挑選資料時，在bootstrap sampling和row sampling中選擇，前者隨機挑取資料後，會保留未挑選過的資料，所以每次執行時都會使用到不同的資料；後者則是每次都是隨機取樣，故有可能會使用到已經取樣過的資料，後來則是選擇使用後者來取樣。

當中有遇到資料型態不合，導致有些node無法執行，比方說資料要進svm learner時，必須先將label的型態轉成string才行，而svm predictor跑完後也無法直接丟進ROC curve裡，起初也無法劃出想要的線，後來才發現得將label的型態再次轉回integer才可以，不然該節點是無法使用label的資料。而大多數的問題主要都是利用number to string、string to number這兩個節點來完成轉換型態。

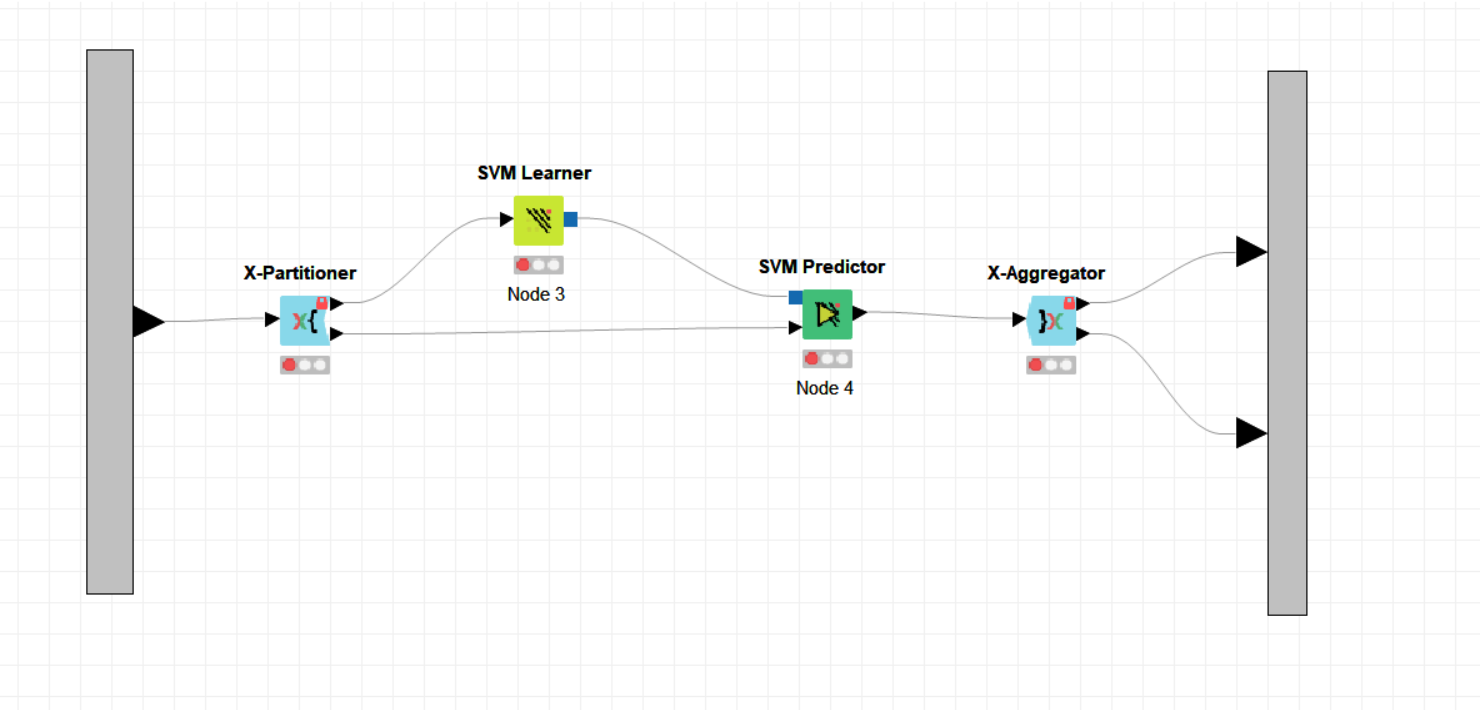
## 三、Methods

workflow

首先將MNIST的train.csv檔讀入，接著利用Row Sampling來進行隨機取樣(1200筆)，將label這列的資料型態轉成字串，然後做分群(50%)，其中一部分拿去SVM Learner進行訓練，另外一部分拿來做比對測試，再將SVM Predictor的結果分別送到scorer和ROC curve，其中送到ROC curve前，必須再將label的型態轉換成整數才可以，否則該node會無法選取label這欄的資料，只能選取到其他pixel的欄位，導致無法畫出想要的圖。

回過頭來看前面的number to string這個節點，當label轉成字串後，額外拉一條線去做驗證方法。送到X-partitioner做分群，同樣地用svm訓練，將SVM predictor的結果送到X-Aggregator，最後將結果傳到scorer，即可完成。

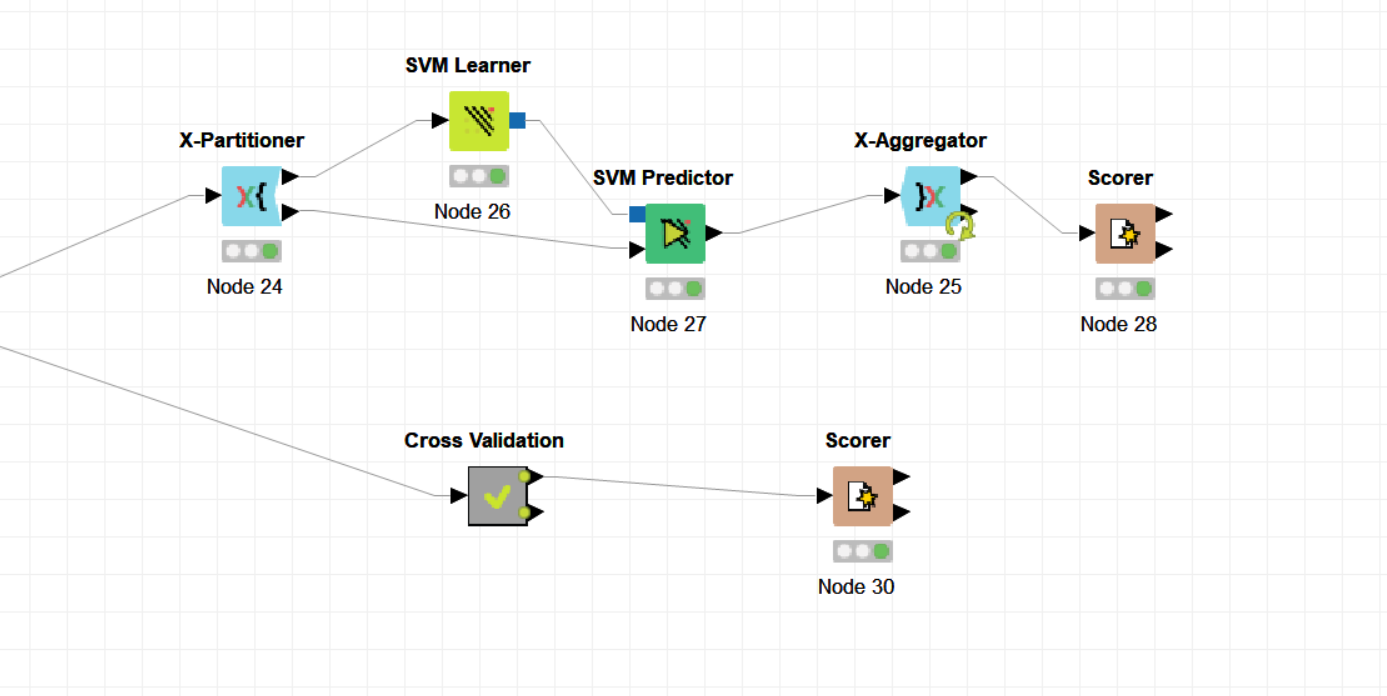
其中，X-partitoner此節點是交叉驗證循環中的第一個節點。 在循環結束時，必須有一個X聚合器來收集每次迭代的結果。 執行這兩次節點之間的所有節點的次數均應執行。而X-Aggregator必須在交叉驗證循環的結尾，並且必須在X-Partitioner節點之後。 它從預測變量節點收集結果，比較預測的類別和真實的類別，並輸出預測結果和迭代統計量。



workflor for 10-Fold Cross-Validation

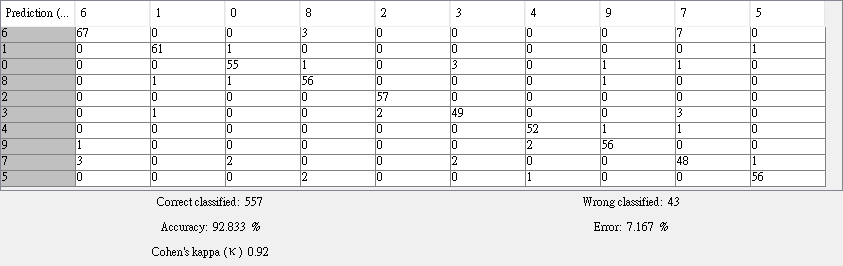
驗證方法(validation method)，我們是用10-Fold Cross-Validation，路徑X-partitioner到X-Aggregator的nodes就是Cross-Validation的workflow，原本我們是使用meta nodes的cross validation節點來使用，但會跑出以下warnning，“WARN Decision Tree Learner 2:0:16 Class column "class" not found or incompatible”，發現其內部節點執行的是Decision Tree的方法，所以我們果斷使用把節點一個個拆開的方法，這樣可讀性也比較高。

後來我們在做完報告的時候，又去測試一下meta nodes的cross validation到底能不能用，發現其實可以從metanode->open把節點打看來進行更改，可以直接從內部修改，也能達到我們想要的效果，如下圖。

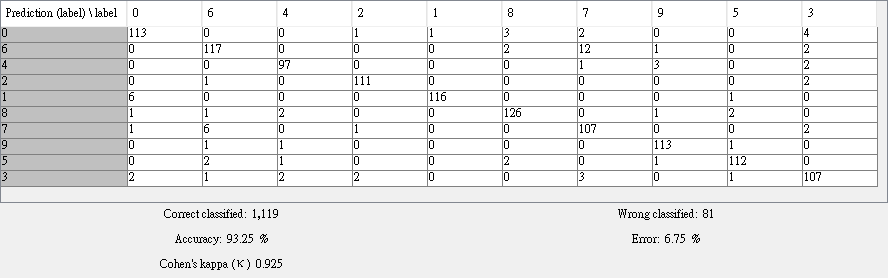


metanodes : Cross-Validation

## 四、Results

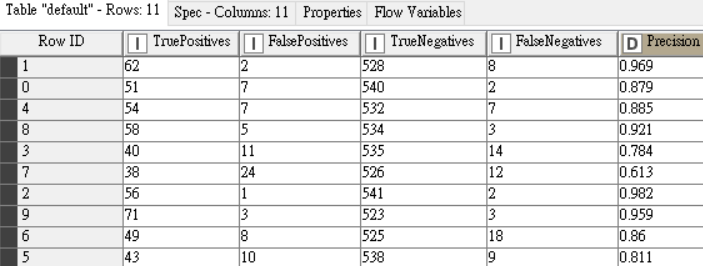


confusion matirx(accurancy:92.833%,kappa:0.92K)

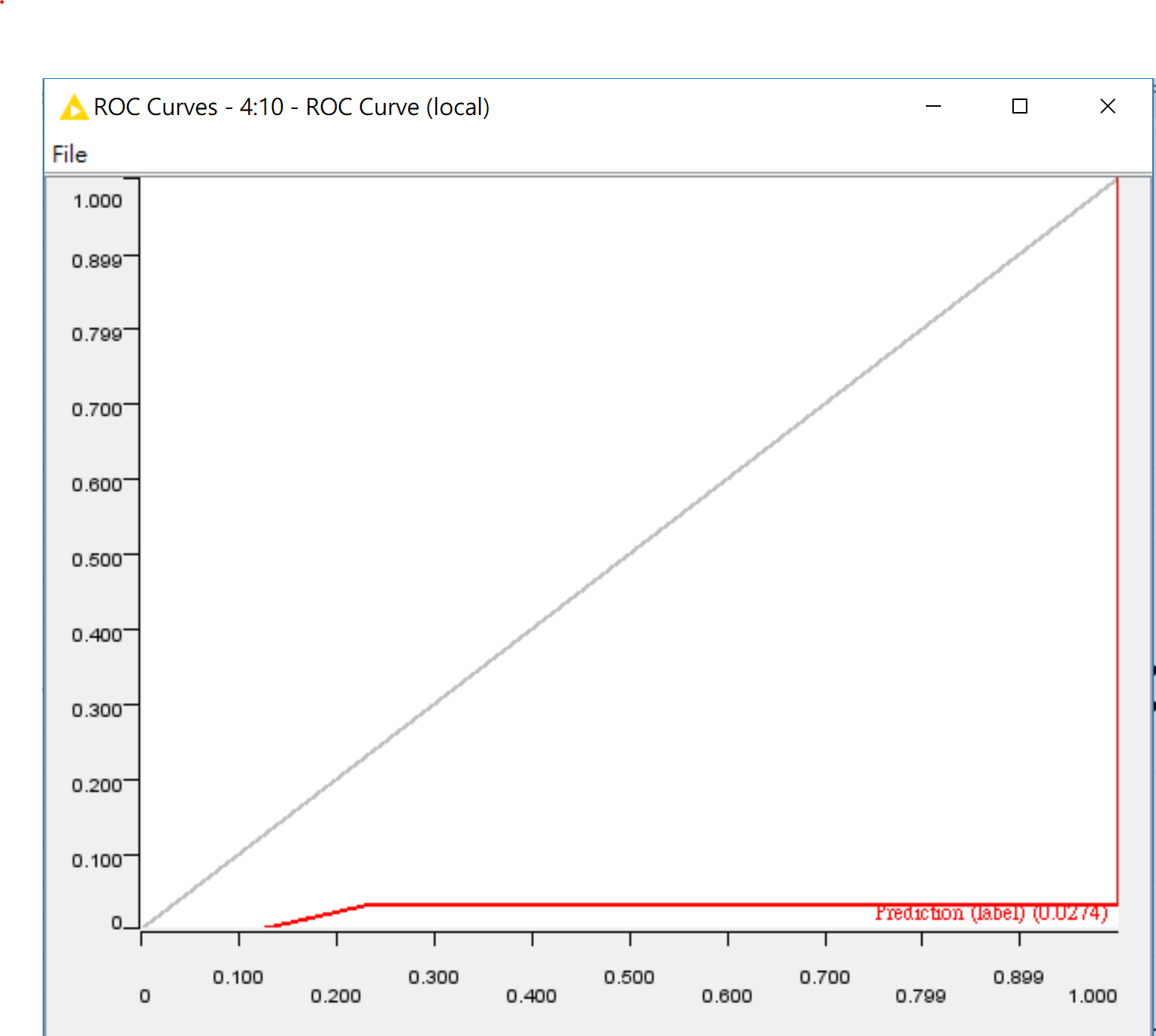


confusion matrix of 10-Fold Cross-Validation

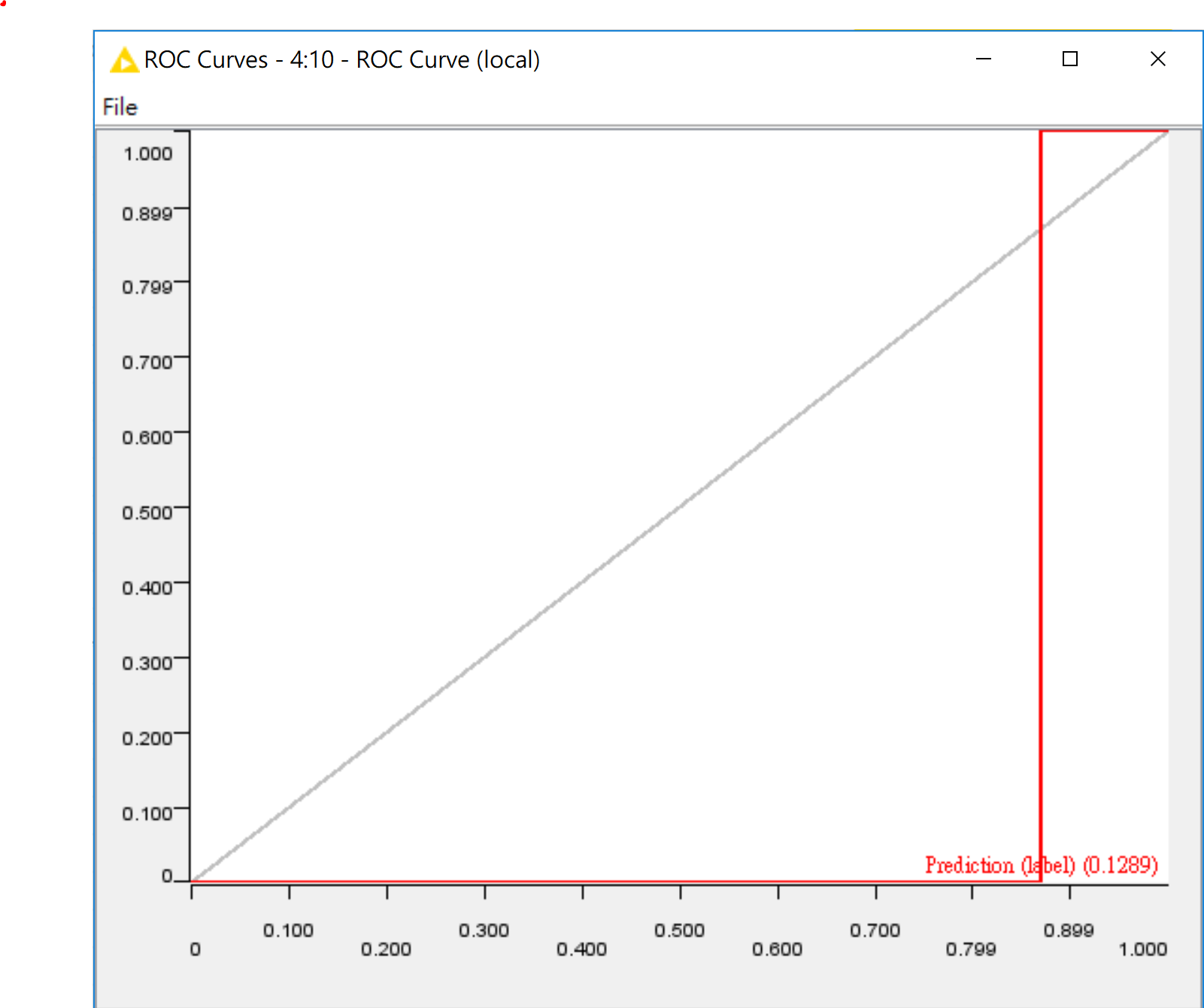
(accurancy:93.25%,kappa:0.925K)



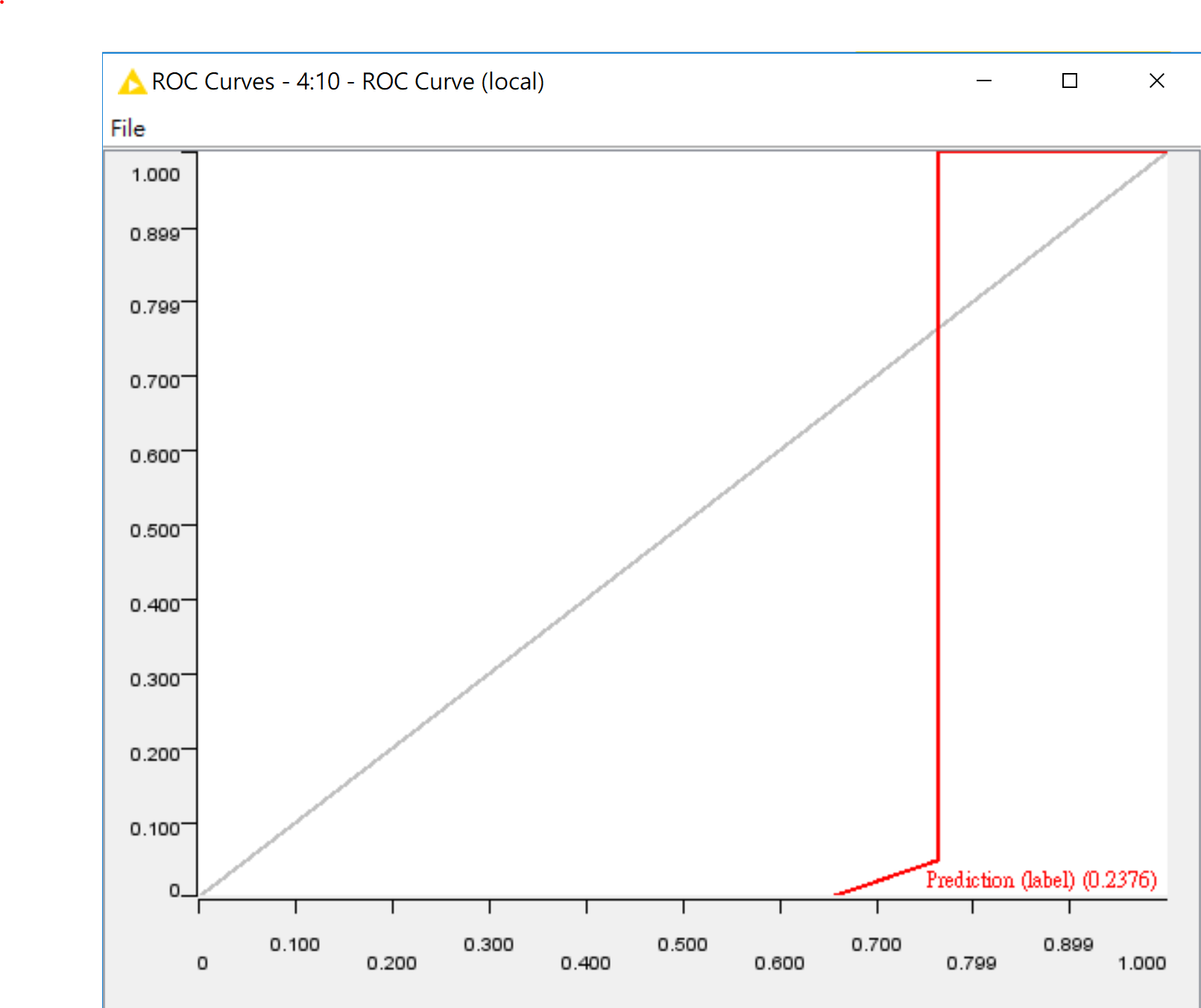
TPr vs. FPr table



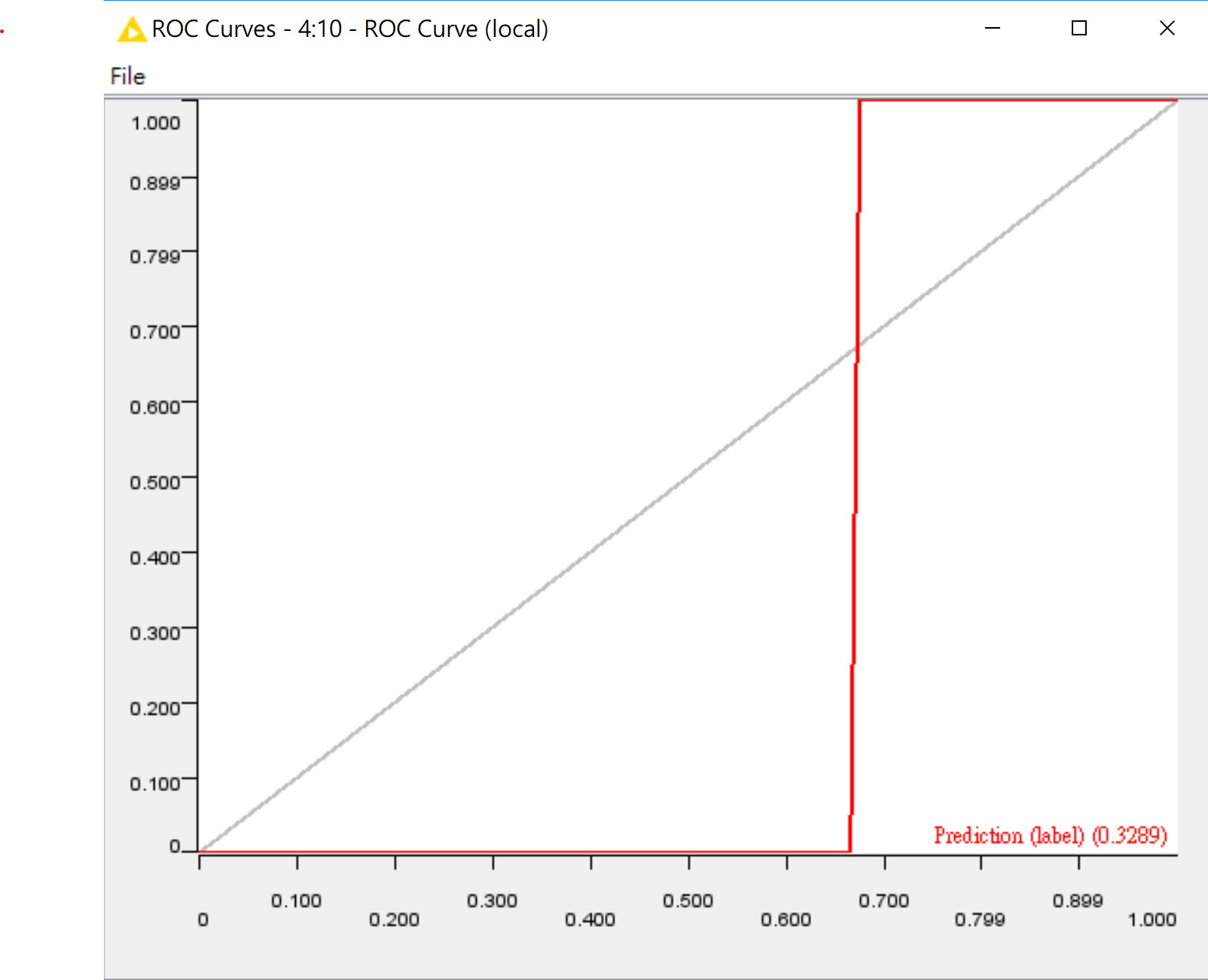
ROC cruve when positive class value is 0



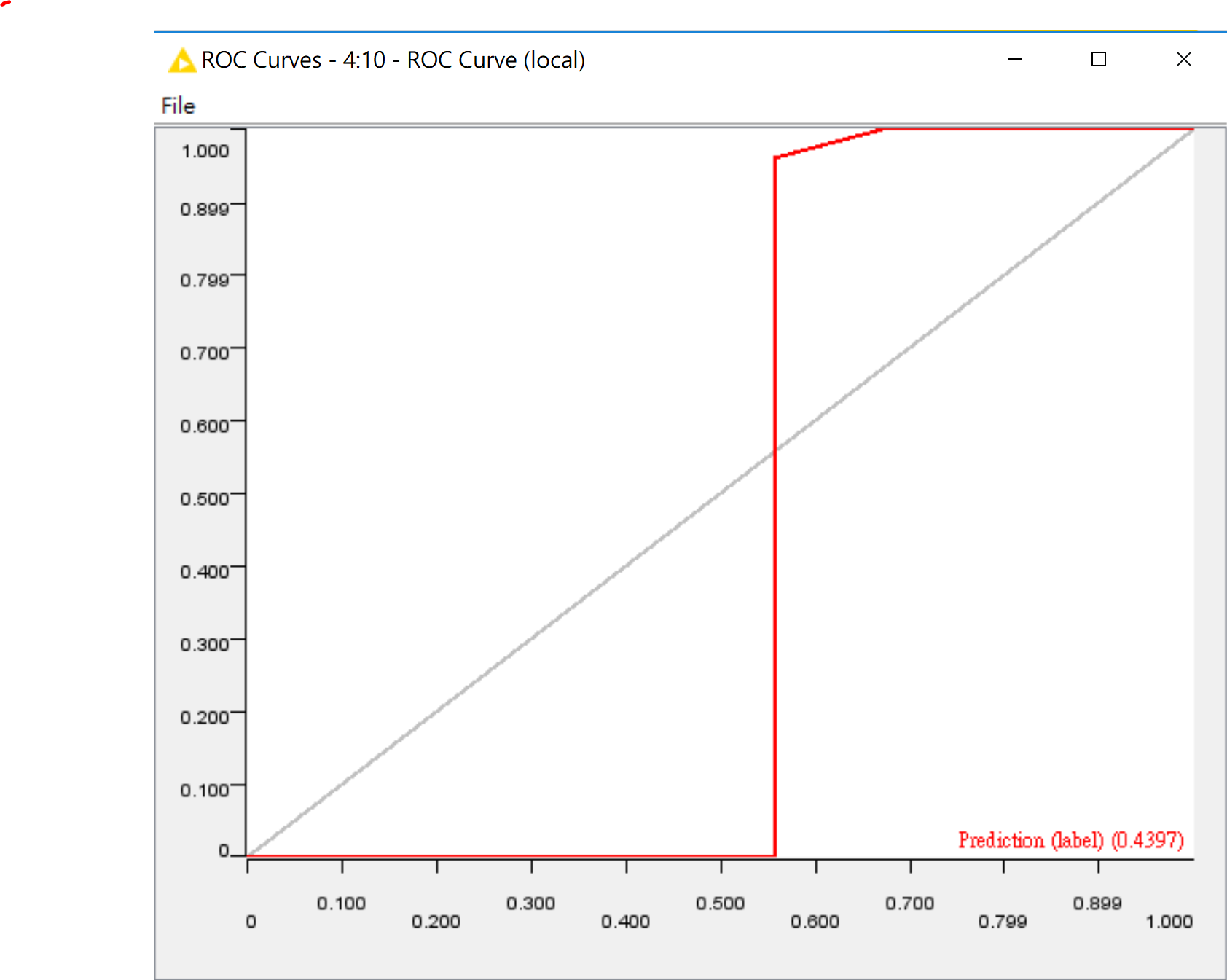
ROC cruve when positive class value is 1



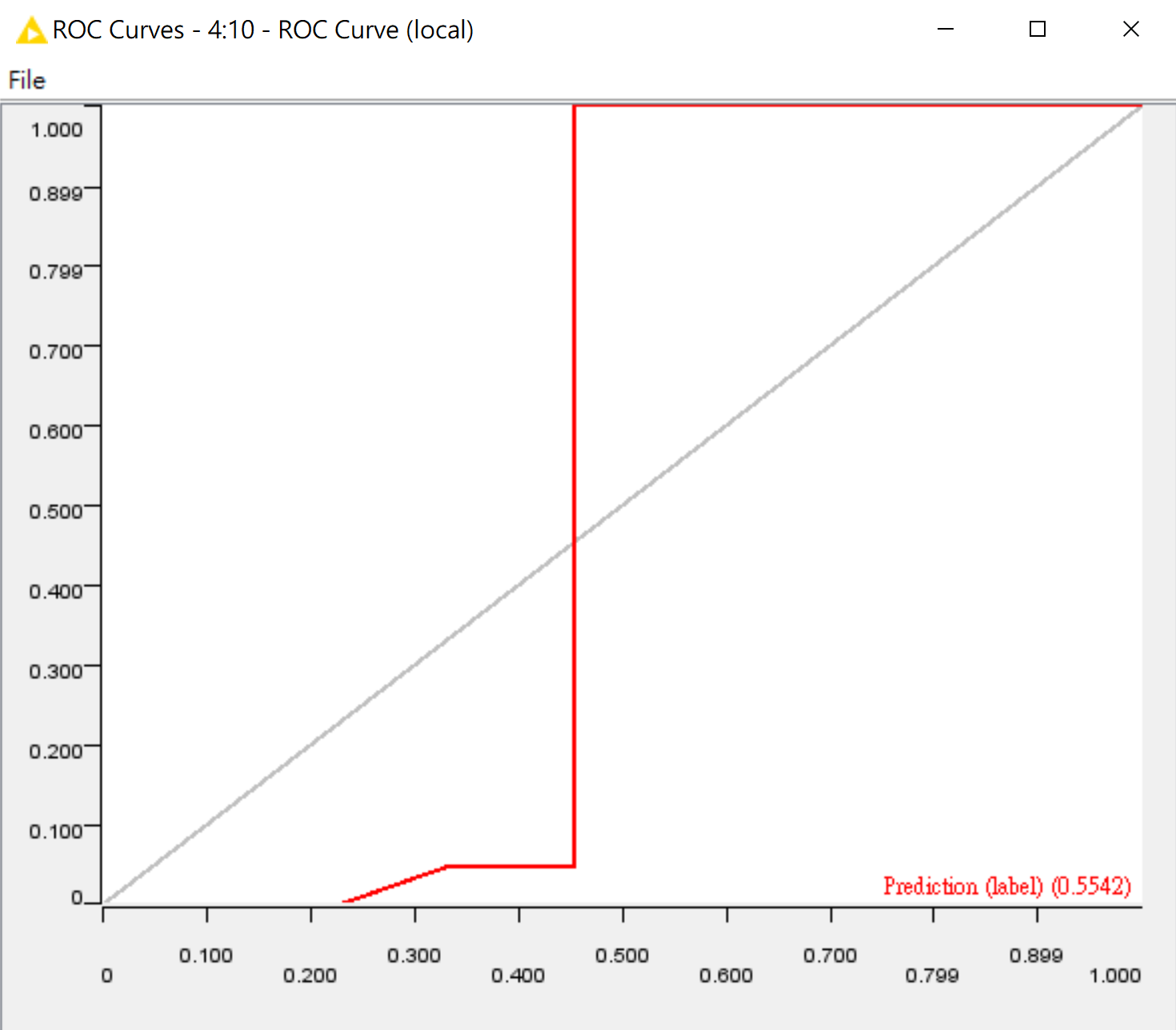
ROC cruve when positive class value is 2



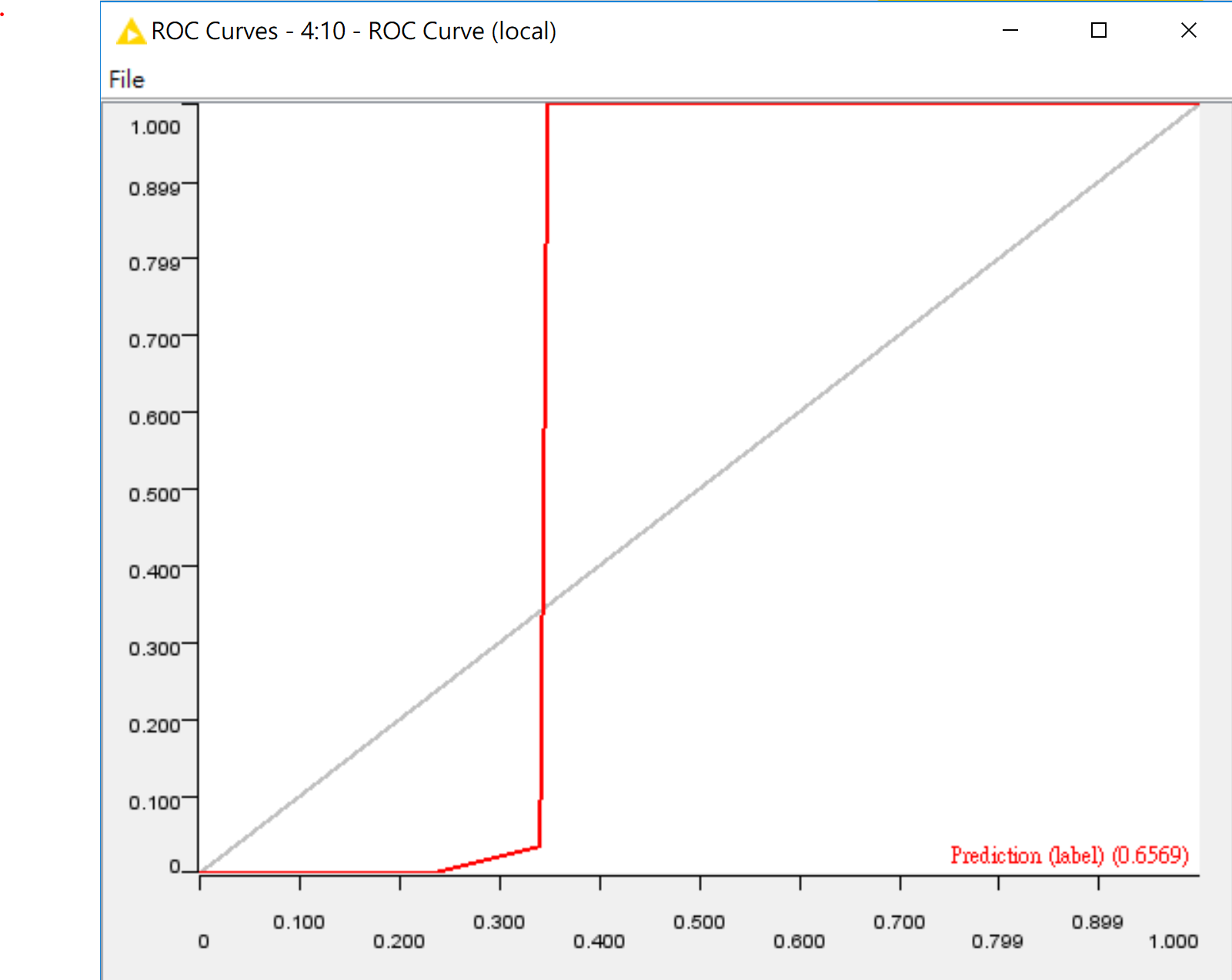
ROC cruve when positive class value is 3



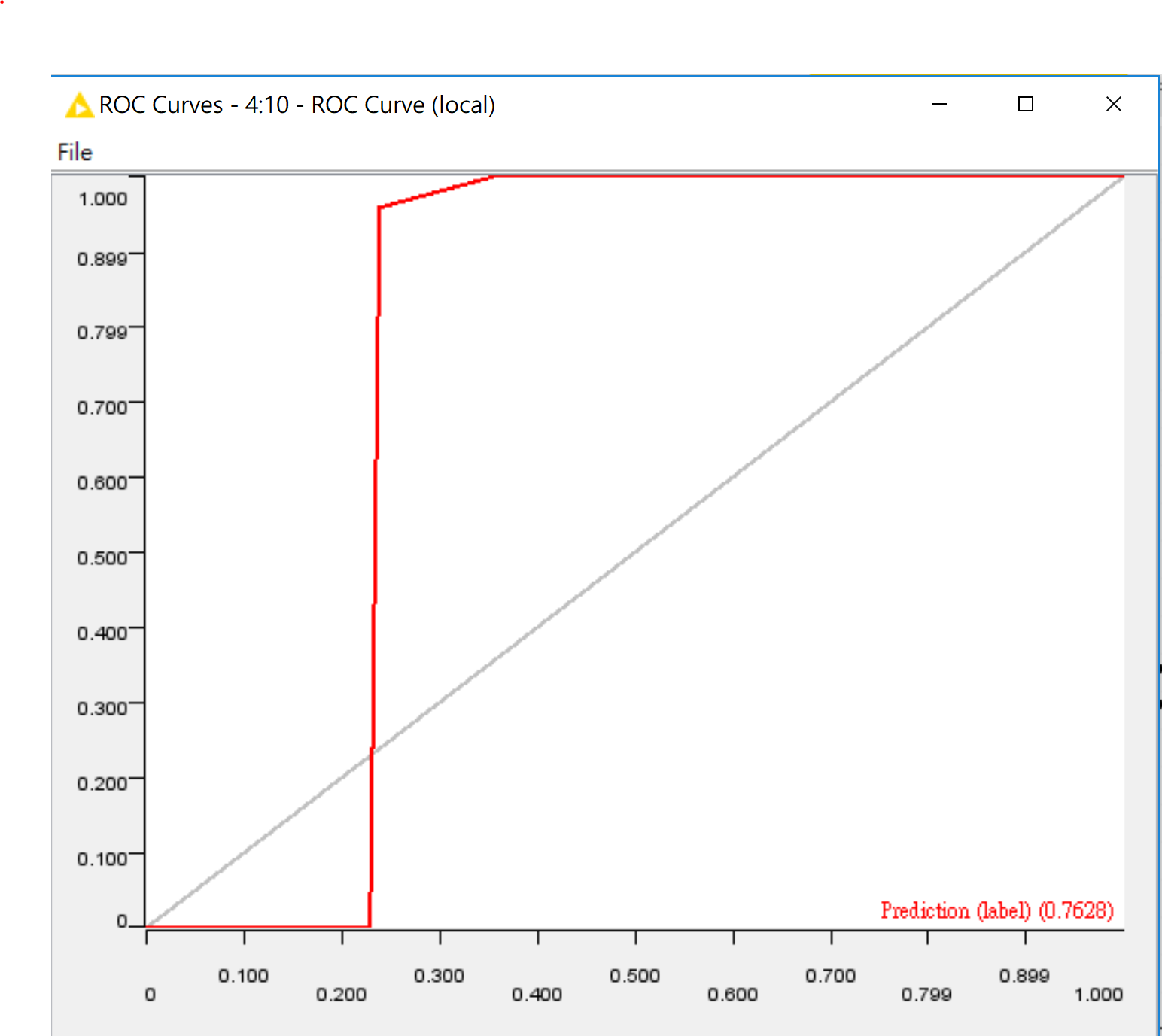
ROC cruve when positive class value is 4



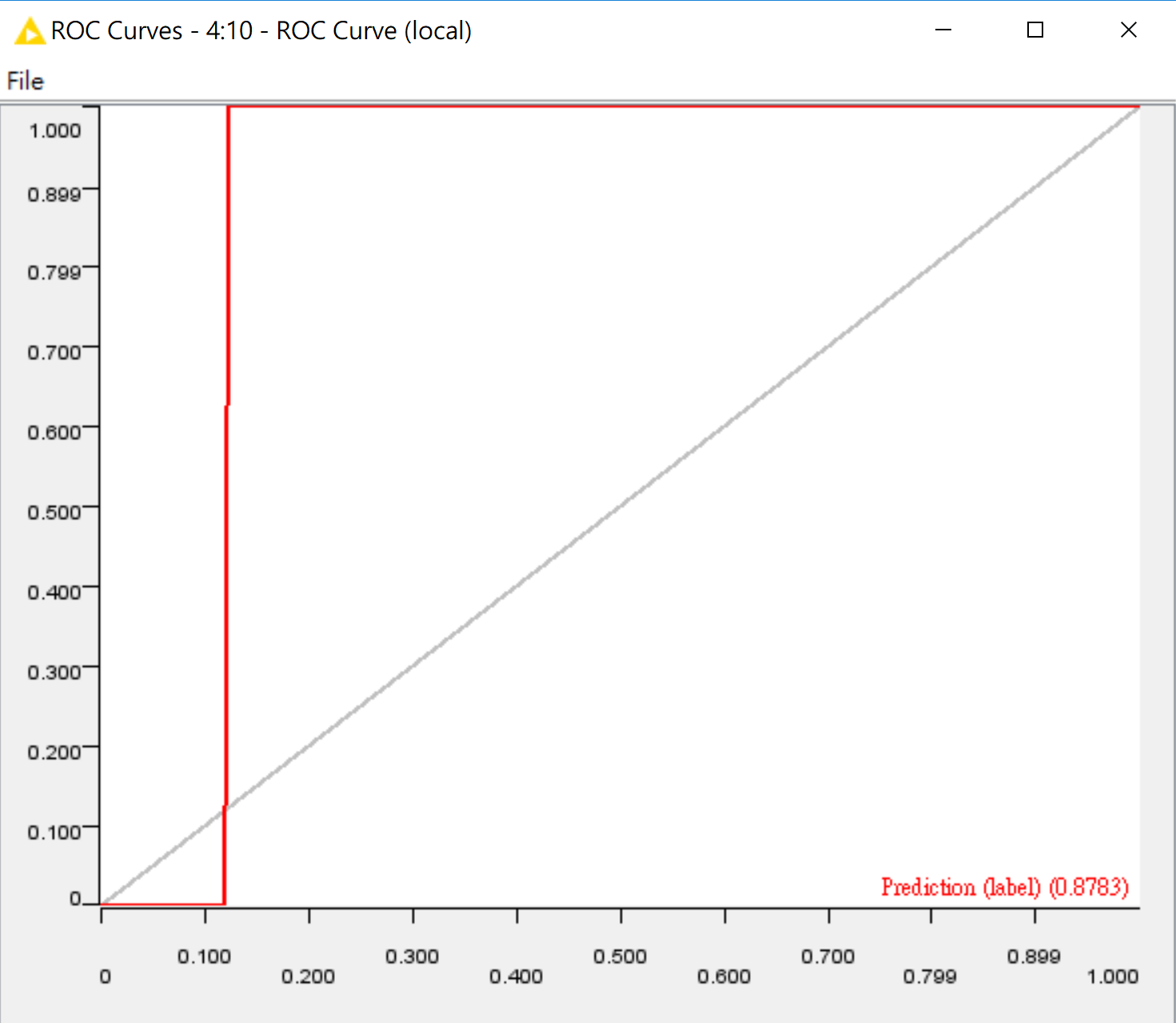
ROC cruve when positive class value is 5



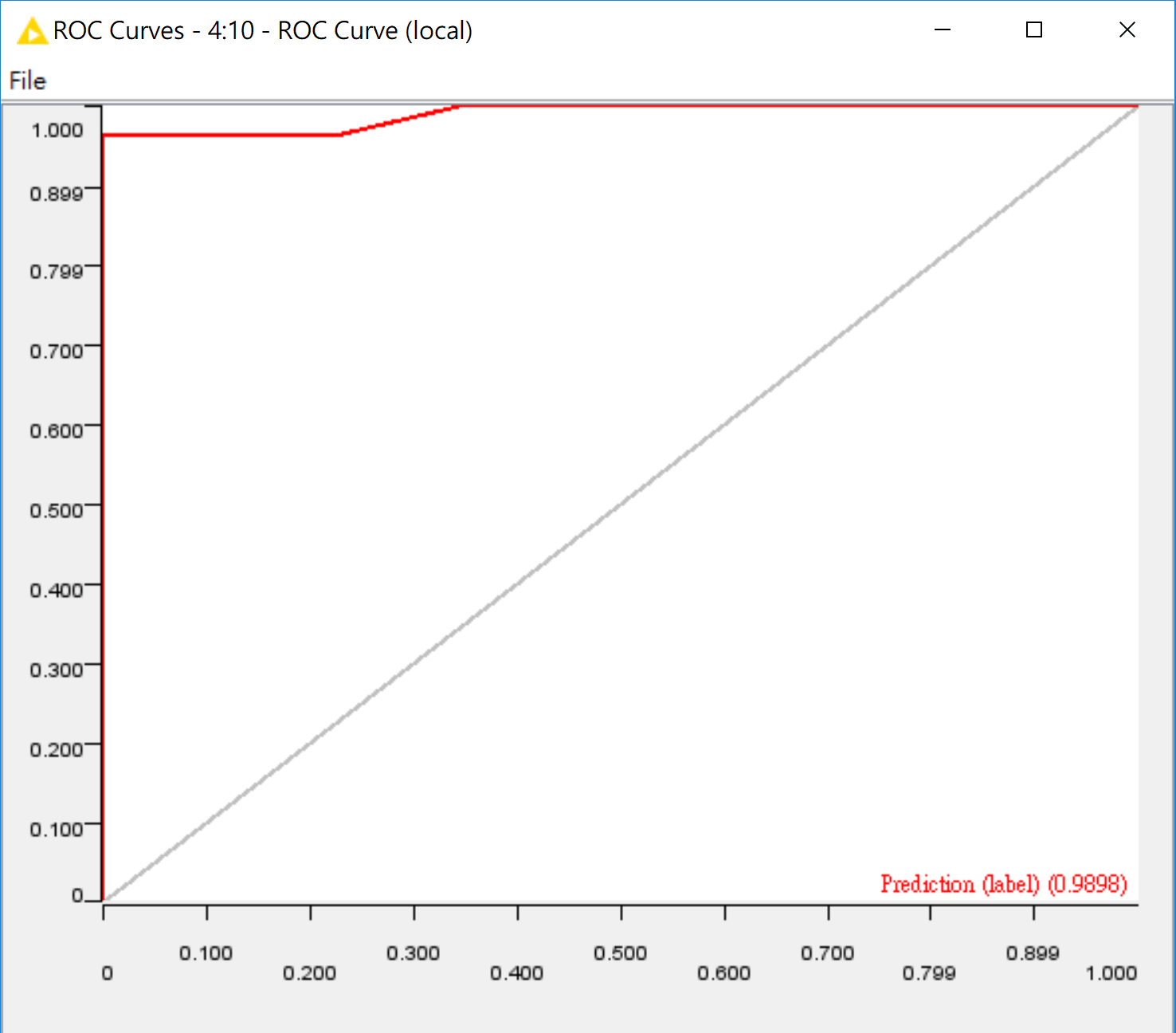
ROC cruve when positive class value is 6



ROC cruve when positive class value is 7



ROC cruve when positive class value is 8



ROC cruve when positive class value is 9

## 五、Discussion

本次使用的dataset總共有60000筆，但我們每次分析只用了1200筆，猜測準確率應該會降低些許，實驗結果也證實了準確率確實比Kannada Kaggle上的實驗結果低一點，經過這次專案又讓我多學會了一些分析技巧，像是交叉驗證方法(Cross Validation)、ROC curve、SVM的觀念等等......。

一開始在處理data的時候遇到了一些麻煩像是，label的值須從integer轉成string才能交由SVM進行處理。後來遇到ROC curve又需要將label的值須從string轉成integer。後來再驗證方法的時候遇到節點warnning也順利的解決掉了。

在ROC curve上，我們針對不同的positive class value來做圖，得到了不同的結果，當positive class value is 9時，其最接近ideal classfier。

理論上來說準確度會比kappa值高一點，我們也從實驗結果發現，我們分析得方法準確度都比kappa值高一些，也是相對合理的。

(github連結：<https://github.com/king87515/knime_MNIST>)