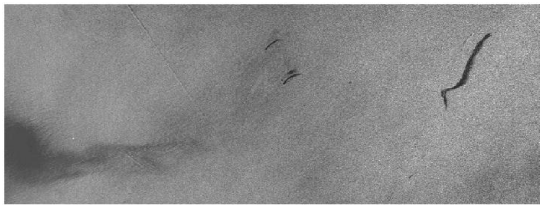
Ricardo Aler Mur

PRÁCTICA 3 (1.5 puntos). **CLASIFICACIÓN CON COSTE EN RAPIDMINER**

Vamos a trabajar con clasificación con coste en Rapidminer. Utilizaremos el fichero oil. El objetivo es la detección de escapes de petróleo. Abajo se puede ver una imagen con un escape. Oil tiene 41 instancias positivas y 896 negativas y es por tanto una muestra muy desbalanceada.



1. Introducción: En el problema *oil* una de las clases es bastante minoritaria. Como sabemos, en ocasiones esto implica que el clasificador aprenderá bien la clase mayoritaria a costa de ignorar la minoritaria. Comprobar si esto ocurre en *oil* usando el clasificador Decision Tree y validación cruzada. ¿Cuáles son los porcentajes de aciertos de cada una de las clases? ¿Hay alguna que se aprenda particularmente mal?

Rapidminer: Usar **template** "Crossvalidation (nominal, decisión tree)"

2. Evaluación teniendo en cuenta el coste: En este apartado se trata de evaluar modelos aprendidos utilizando matrices de coste en lugar de usar sólo el porcentaje de aciertos. Usaremos el **template** "Compare Learning Algorithms by Significance Test". Esta template está pensada para regresión, por lo que habrá que hacer cambios: a) poner decisión trees y algún otro algoritmo de clasificación y b) reemplazar los dos "performance" por "performance cost". Dentro de los dos "performance cost", introducir una matriz de costes en la que el coste de clasificar mal un dato de la clase mayoritaria es de 25 euros, mientras que el coste de clasificar mal uno de la clase minoritaria es de 1000 euros. ¿Cuál es el mejor algoritmo en términos de coste? ¿Es la diferencia estadísticamente significativa?

3. Aprendizaje teniendo en cuenta el coste. Metacost: En el apartado anterior vimos que uno de los algoritmos es mejor que los demás respecto al coste. Sin embargo, el entrenamiento de cada uno de los algoritmos intenta minimizar el error de clasificación, no el coste. Simplemente por casualidad uno de ellos es mejor en el coste. Vamos ahora a intentar minimizar directamente el coste usando el meta-clasificador *MetaCost*. En este meta-clasificador hay que rellenar el clasificador y la matriz de costes. Como clasificador usad el clasificador que obtuvo los mejores resultados en el apartado anterior. Usad también la matriz de costes del apartado anterior. ¿Se consigue mejorar el coste conseguido en el apartado previo?

4. Visualización de curvas ROC: Ahora visualizaremos las curvas ROC de varios clasificadores. Usaremos la **template** "Compare ROCs" y Decision Tree, Naive Bayes y Random Forests. ¿Podemos sacar alguna conclusión de estas curvas? ¿Hay algún algoritmo que sea mejor que los demás en general? ¿Por qué puede ser?

NOTA: Aunque no es necesario, ir algo más allá de los puntos de la práctica, cambiando algún parámetro, o usando algún clasificador o módulo de Rapidminer puede garantizar una buena nota.

NORMAS DE ENTREGA