Facultat d'Informàtica de Barcelona

Cuatrimestre de otoño, 2022/2023

Práctica de Aprendizaje Automático: Seismic Bumps

Guillem González
(guillem.gonzalez.valdivia@estudiantat.upc.edu)
Carles Orriols
(carles.orriols@estudiantat.upc.edu)

Contents

1 Introducción

"La actividad minera ha estado y está siempre relacionada con la aparición de peligros que se denominan comúnmente peligros mineros. Un caso especial de este tipo de amenaza es el peligro sísmico que se produce con frecuencia en muchas minas subterráneas. El peligro sísmico es el más difícil de detectar y predecir de los peligros naturales y, en este sentido, es comparable a un terremoto. Los sistemas de control sísmico y sismoacústico cada vez más avanzados, y permiten una mejor comprensión de los procesos del macizo rocoso y la definición de métodos de predicción del riesgo sísmico.

Sin embargo, la precisión de los métodos creados hasta ahora dista mucho de ser perfecta. La complejidad de los procesos sísmicos y la gran desproporción entre el número de eventos sísmicos de baja energía y el número de fenómenos de alta energía hace que las técnicas estadísticas sean insuficientes para predecir peligro sísmico. Por lo tanto, es esencial buscar nuevas oportunidades de mejorar la predicción de la peligrosidad utilizando también métodos de aprendizaje automático."[1]

El dataset elegido se corresponde con un conjunto de datos relacionados con el sector minero, concretamente fueron obtenidos en una mina de carbón de Polonia. El dataset describe, por cada muestra, una situación en la que se ha dado un evento sísmico, describiendo factores que determinan la fuerza del mismo, como los diferentes niveles de energía por cada evento, y otros parámetros.

En estos casos, y según se especifica en el enunciado, la estadística es inefectiva para la predicción de eventos, por lo que se requiere del uso de técnicas más avanzadas. A partir de estos datos, se busca determinar, a partir de técnicas de aprendizaje, predecir futuras situaciones, para discernir si son situaciones de peligro o no peligro.

2 Objetivos

El objetivo de este estudio es determinar, a partir de los atributos de la población, el atributo class de futuras muestras, partiendo del aprendizaje de los datos de la tabla.

La variable class determina, de manera dicotómica, si el evento registrado supone un evento de riesgo o no. Los valores a 0 determinan que la situación no es de riesgo, mientras que los valores a 1 determinan que sí hay riesgo.

3 Dataset

La tabla de descripción del dataset que se presenta en la documentación es la siguiente:

Data Set Characteristics	Multivariate
Number of Instances	2584
Area	N/A
Attribute Characteristics	Real
Number of Attributes	19
Date Donated	2013-04-03
Associated Tasks	Classification
Missing Values?	N/A
Number of Web Hits	85975

Veamos las descripciones de cada uno de los atributos. Recuperamos estas descripciones de la página de la UCI:

- seismic: Resultado de la evaluación de la peligrosidad sísmica por turnos en la explotación minera obtenida por el método sísmico (a - ausencia de peligro, b - peligro bajo, c - peligro alto, d - estado de peligro)
- seismoacoustic: Resultado de la evaluación de la peligrosidad sísmica por turnos en la explotación minera obtenido por el método sismoacústico

- shift: Información sobre el tipo de turno (W - turno de obtención de carbón, N - turno de preparación)
- Genergy: Energía sísmica registrada en el turno anterior por el geófono más activo (GMax) de los geófonos que monitorizan el tajo largo
- gpuls: Número de pulsos registrados en el turno anterior por el GMax
- gdenergy: Desviación de la energía registrada en el turno anterior por el GMax con respecto a la energía media registrada durante ocho turnos anteriores
- gdpuls: Desviación del número de impulsos registrados en el turno anterior por GMax con respecto al número medio de impulsos registrado durante ocho turnos anteriores
- ghazard: Resultado de la evaluación de la peligrosidad sísmica en el turno de trabajo de la mina obtenido por el método sismoacústico basado en el registro procedente únicamente de GMax
- nbumps: Número de golpes sísmicos registrados en el turno anterior
- nbumps2: El número de golpes sísmicos (en el rango de energía $[10^{\mathbf{2}},10^{\mathbf{3}}]$) registrados en el turno anterior
- nbumps3: El número de golpes sísmicos (en el rango de energía $[10^{\mathbf{3}},10^{4})$) registrados dentro del turno anterior
- nbumps4: El número de golpes sísmicos (en el rango de energía $[10^{4},10^{5})$) registrados en el turno anterior
- nbumps5: El número de golpes sísmicos (en el rango de energía $[10^{5},10^{6})$) registrados en el último turno
- nbumps6: El número de golpes sísmicos (en el rango de energía $[10^{6},10^{7})$) registrados en el turno anterior
- nbumps7: El número de golpes sísmicos (en el rango de energía $[10^{7},10^{8})$) registrados en el turno anterior
- nbumps89: El número de golpes sísmicos (en el rango de energía $[10^{8},10^{10})$) registrados en el turno anterior
- energy: Energía total de los golpes sísmicos registrados en el turno anterior
- maxenergy: La energía máxima de las protuberancias sísmicas registradas en el turno anterior
- class: El atributo de decisión o atributo a predecir: "1" significa que se ha producido un golpe sísmico de alta energía en el turno siguiente ("estado peligroso"), "0" significa que no se produjeron golpes sísmicos de alta energía en el turno siguiente ("estado no peligroso")

El objetivo es predecir si la siguiente estimación tendrá valor class = 0 o class = 1 (no peligro/peligro).

Observamos que el dataset consta de 19 atributos y 2584 observaciones. Todos los atributos son numéricos menos seismic, seismoacustic y shift, que son categóricos. La variable objetivo es class, que es un atributo booleano que determina si existe peligro (=1) o no (=0).

Una vez estudiados los datos (véase las tablas estadísticas y gráficos en el archivo jupyter notebook) podemos mencionar lo siguiente:

Observamos que, a priori, ninguno de ellos consta de una distribución normal. Las variables categóricas (seismic, seismoacustic y shift) se corresponden con la peligrosidad de la actividad y el tipo de actividad (shift/coal-getting). Las variables numéricas se refieren a las mediciones registradas de energía, números de baches sísmicos registrados en rangos de energía, estadísticas de energía promedio y máximos de energía y, finalmente, resultado de peligrosidad / no peligrosidad.

Observamos también que los valores de los atributos no se reparten de forma homogenea, sino que se concentran en valores determinados, por eso encontramos picos tan altos en los histogramas.

En cuanto a la correlación de las variables, encontramos que la variable objetivo "class" no tiene una correlación buena con ninguna de las demás variables. Los valores de correlación más cercanos se encuentran entre el 0.2 y 0.3. Por otra parte, sí que observamos buenas correlaciones entre otras variables, siendo la correlación más fuerte entre maxenergy y energy, con un 1 de correlación; rbumps3 con rbumps, con un valor aproximado entre 0.8 y 0.9; y rbumps2 y rbumps, con un valor aproximado de 0.8.

4 Preprocesamiento de los datos

En este apartado vamos a hacer un preprocesamiento de los datos, que no es más que "limpiar" y dejar listos los datos antes de aplicar cualquier modelo, para conseguir mejores resultados en dicha aplicación.

4.1 Completitud de los datos

En primer lugar, vamos a ver si los datos son completos o de lo contrario hay algún dato faltante, ya sea por no tomar el dato en el momento oportuno o por perdida del mismo en algún momento desde su medición. En nuestro caso no falta ningún dato en ninguna de las variables, cosa que nos ahorra tener que substituir estos datos faltantes por valores calculados por nosotros y las alteraciones, por pequeñas que sean, que esto genera.

4.2 Valores anómalos o incorrectos

Una vez visto que todas las entradas son completas para todas las variables, queremos ver si hay valores anómalos o fuera de lo normal, ya sea por errores de medición, valores que por lo que sea han sido alterados en el momento de su medición o de lo contrario son datos totalmente válidos que son poco frecuentes. Para evitar alteraciones relativamente importantes del modelo por un solo dato, debemos ver si hay alguno de estos casos y analizar el porqué. En nuestro caso podemos ver datos poco frecuentes en la mayoría de las variables. Parece que no hay nada fuera de lo normal y todo corresponde a que los terremotos con más energía y más fuertes son menos frecuentes que los de menos intensidad. No vemos ningún valor fuera de lo normal.
Por lo tanto, estos datos que están lejos de la media no van a ser manipulados, ya que podríamos decir que representan la parte más importante del estudio, el caso en que tenemos terremotos importantes y queremos poder predecir para actuar previamente y evitar daños mayores.

4.3 Variables no continuas

Transformamos las variables categoricas o binarias en variables "dummies".

4.4 Variables irrelevantes

En el histograma que hemos usado para ver datos anómalos también hemos podido percibir lo inútil que son algunas variables, en concreto el id de la fila. El resto de variables a priori no podemos decir que sean inútiles, hay algunas que parecerían redundantes, pero no podemos asegurarlo y, por lo tanto, las dejamos, ya que podríamos perder información muy importante.

4.5 Variables nuevas

En esta sección deberíamos generar nuevas variables si vemos que nos faltan datos para seguir con el estudio. A priori, no sale la necesidad de crear variables nuevas, ya que tenemos muchos datos a primera vista relacionados con el objetivo del problema.

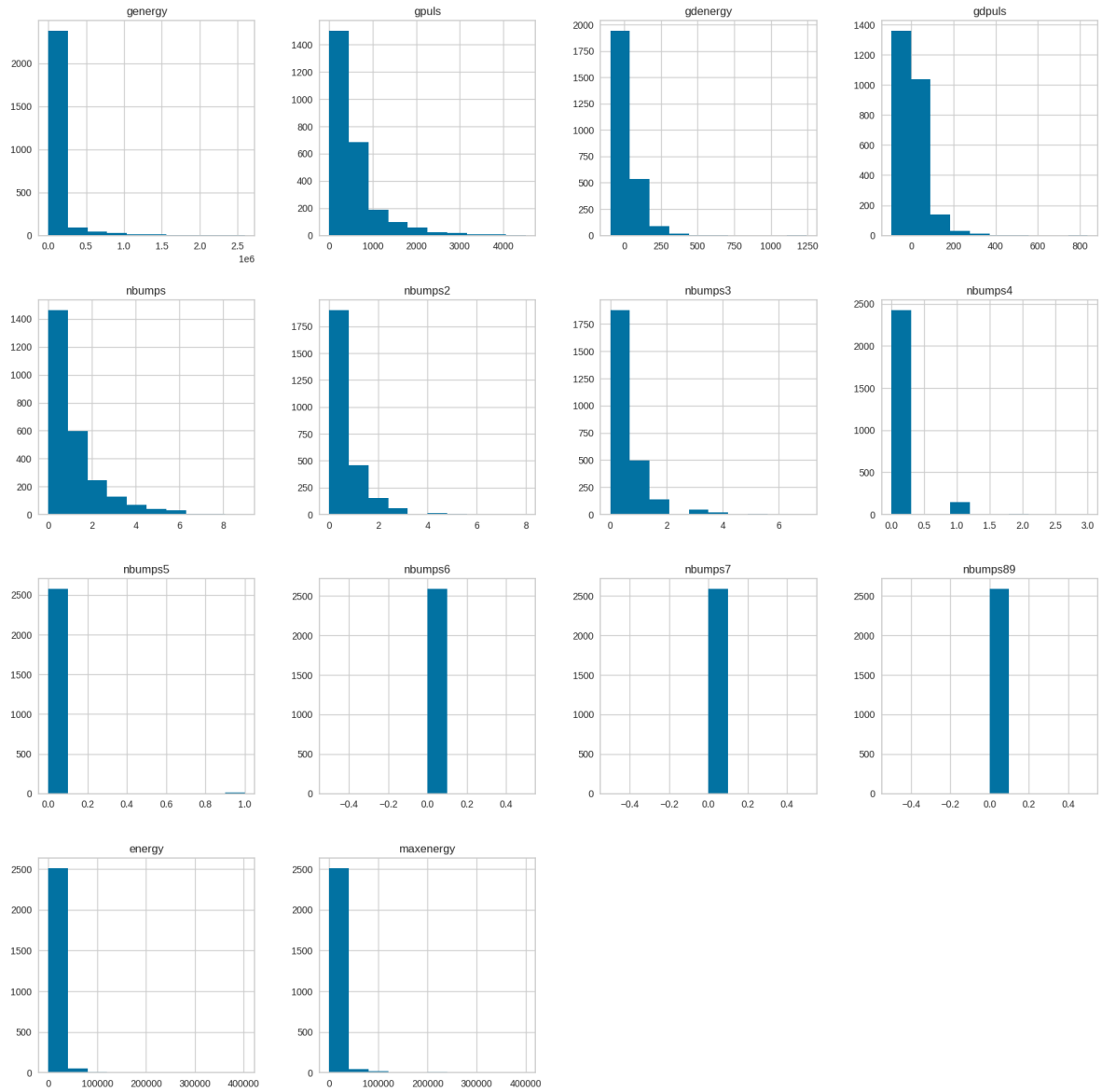
4.6 Normalización de variables

Vamos a usar estandarización las variables, excepto para usar en la regresión logística que nos daría problemas si lo hacemos con los valores estandarizados.

4.7 Transformación de variables

A priori no hay necesidad de transformar ninguna variable y por eso damos por buenas las variables tal y como las hemos dejado hasta el momento.

Aquí dejamos el histograma de los datos que nos ha parecido interesante para poder contrastar las conclusiones que nosotros hemos extraído de él en este apartado.



Hasta aquí hemos llegado en el preprocesamiento de los datos. Los datos están preparados para poder empezar a usar modelos y tener unos resultados óptimos.

5 Selección de modelos y estimación de rendimiento

5.1 Selección de modelos lineales/cuadráticos

En la documentación facilitada en el dataset escogido ya especifican que es un problema de clasificación. De este modo, hemos decidido analizar los siguientes 3 modelos, regresión logística, Naive Bayes y k-nearest neighbors, entre los mencionados en el enunciado de la pràctica. Una que, y compararlos para ver cuál de ellos puede dar mejores resultados.

5.2 Separación del dataset

Evidentemente, para hacer un buen análisis de cada uno de los modelos necesitamos separar el conjunto entre entrenamiento y test, como se haría en cualquier estudio decente. En ningún caso podemos usar el conjunto de test para entrenar el modelo, en ese caso estaríamos condicionando mucho el error y no es como trabajamos en el mundo real.
En este caso hacemos una partición del 70% para el conjunto de entrenamiento y el restante, 30%, para el test o validación.

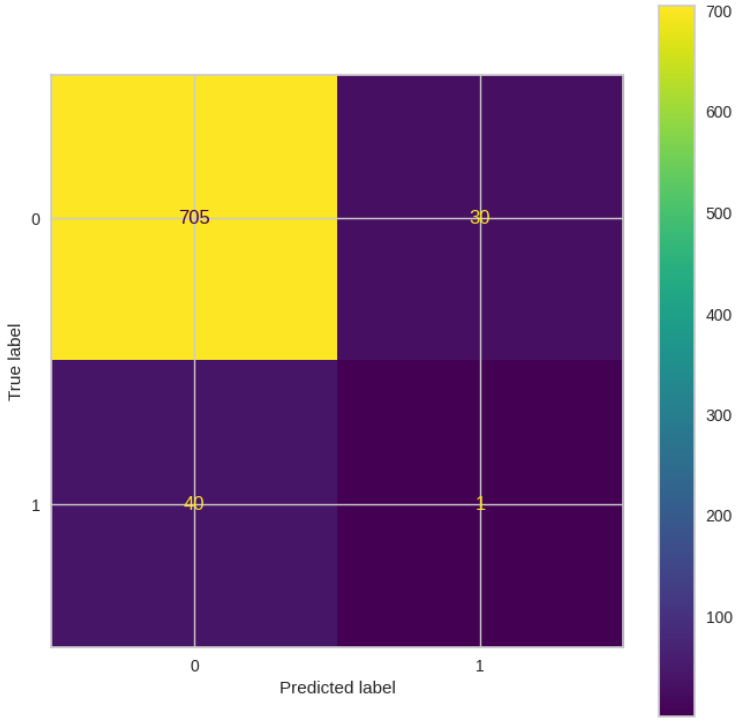
5.3 Regresión logística

El primer de los algoritmos que vamos a usar es la regresión logística, que estima la siguiente probabilidad $p(y = C_k|X = x)$ ahorrándose así el tener que hacer una distribución específica. La estimación mencionada corresponde a los ratios entre las probabilidades de las clases que se convierten en probabilidades a través de la función $p(C|X) = \sigma(w \cdot x)$

La cross-validation score nos da un resultado muy cercano al 0.90, que hace ver con buenos ojos este modelo y seguir con optimismo en el analisi de este metodo. Evidentemente hay que ver más en profundidad para hacernos una mejor idea de lo que es capaz este metodo. Seguimos haciendo un classification report con los datos de entrenamiento, con los siguientes resultados:

	precision	recall	f1-score	support
0	0.96	0.95	0.95	745
1	0.02	0.03	0.03	31
accuracy			0.91	776
macro avg	0.49	0.49	0.49	776
weighted avg	0.92	0.91	0.92	776

Seguidamente veremos la matriz de confusión



Al contrario de lo que pensamos en un principio, siendo bastante optimistas en este método, el resultado deja mucho que desear. La clase 1, que es la más importante en este problema, tiene muchísimas dificultades para ser pronosticada. Podemos ver que mientras para la precisión de la clase 0 supera 0.90 de sobra, la clase 0 no ha seleccionado mucho más de lo que haría el puro azar.

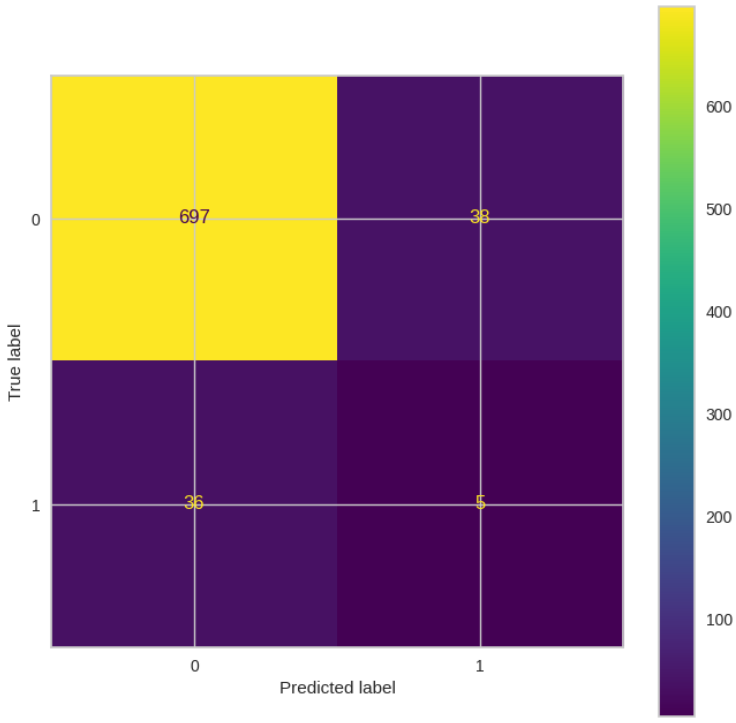
5.4 Naive Bayes

Como segundo método tenemos el Naive Bayes, con el que hemos hecho un análisis parecido al anterior para ver si lo mejoramos.

En un principio parece que ha empeorado un poco, però vamos a ver la predicción de valores:

	precision	recall	f1-score	support
0	0.95	0.95	0.95	733
1	0.12	0.12	0.12	43
accuracy			0.90	776
macro avg	0.54	0.53	0.53	776
weighted avg	0.90	0.90	0.90	776

Seguidamente veremos la matriz de confusión:



Pensábamos que podía tener resultados peores, pero de lo contrario hemos multiplicado por 5 la predicción de la clase 1, sin perder apenas en la clase 0. Esto hace ver muy claramente que este método para nuestro problema responde mejor. Aunque hemos afirmado que mejora mucho, hay que ser conscientes de que seguimos con unas predicciones muy bajas, no podemos permitirnos predecir 1 de cada 8 terremotos y además hacer tantas falsas alarmas como terremotos no predecidos, la población dejaría de confiar en nuestros avisos.

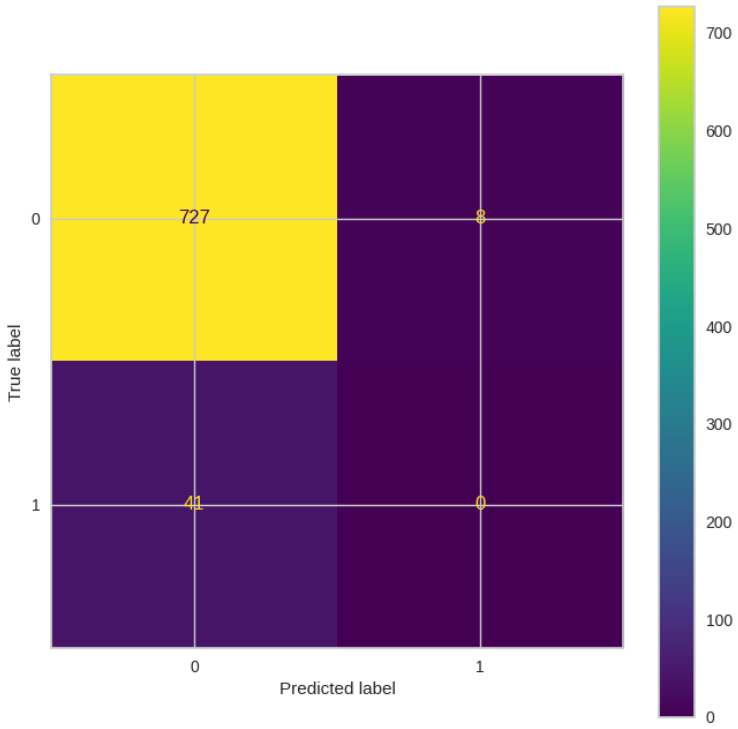
5.5 K-Nearest Neighbors

Este método, tal y como dice su nombre, trata de agrupar los vecinos más cercanos entre ellos en clústeres. Es un método de clasificación supervisada. Este es el último de los métodos que aplicamos.

La cross-validation score es la más alta de las tres, pero como hemos visto, tampoco podemos fiarnos demasiado, sobre todo en los problemas que la clase importante es tan minoritaria dentro del dataset. Así que vamos a ver como classifica cada valor.

	precision	recall	f1-score	support
0	0.99	0.95	0.97	768
1	0.00	0.00	0.00	8
accuracy			0.94	776
macro avg	0.49	0.47	0.48	776
weighted avg	0.98	0.94	0.96	776

Seguidamente veremos la matriz de confusión:



No predice ni un terremoto de nivel alto bien, pero que en el caso de la regresión logística. Y eso que hemos buscado los mejores valores para los parametros que usa el KNN. Claramente reconfirma que la cross validation score hay que cogerla con pinzas.

5.6 Selección de modelos no lineales

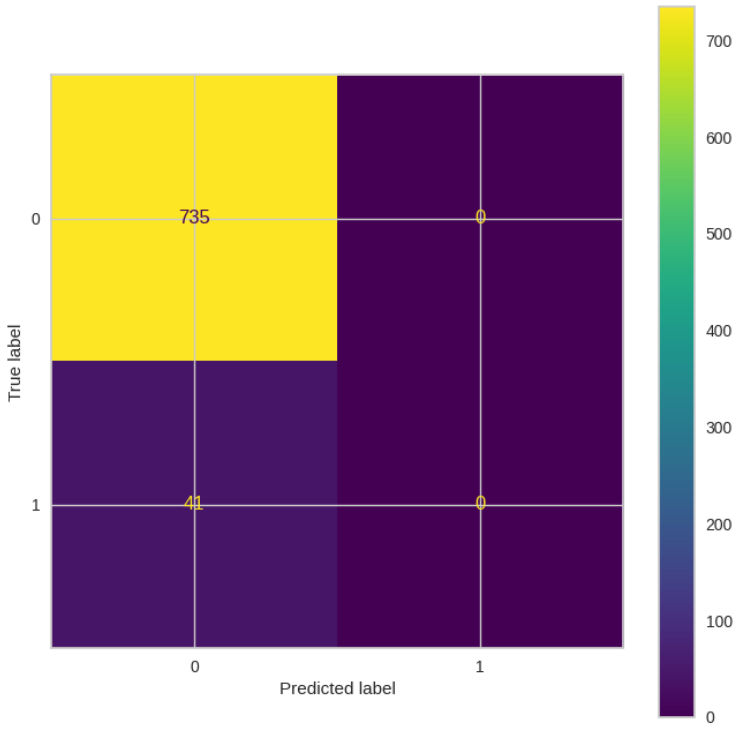
Una vez hechos los metodos lineales y ver lo mal que han ido es hora de ver como se comportan los metodos no lineales. Para esto hemos selecionado tres de ellos, que son los siguientes: Multi-Layer Perceptron, Random Forest i Gradient Boosting

5.7 Multi-Layer Perceptron

Para utilizar MLP debemos normalizar los datos. Como hemos visto en laboratorio, la estandarización proporciona una mayor convergencia, por lo que el proceso es más rápido, ya que tenemos que hacer un análisis de que valores son mejores para los parámetros que usa MLP. Pero vemos que este modelo parece fallar en los mismos puntos que los lineales, que predicen mal la clase 1. Podemos ver la precisión de clasificación aquí:

	precision	recall	f1-score	support
0	1.00	0.95	0.97	776
1	0.00	0.00	0.00	0
accuracy			0.95	776
macro avg	0.50	0.47	0.49	776
weighted avg	1.00	0.95	0.97	776

Seguidamente veremos la matriz de confusión:



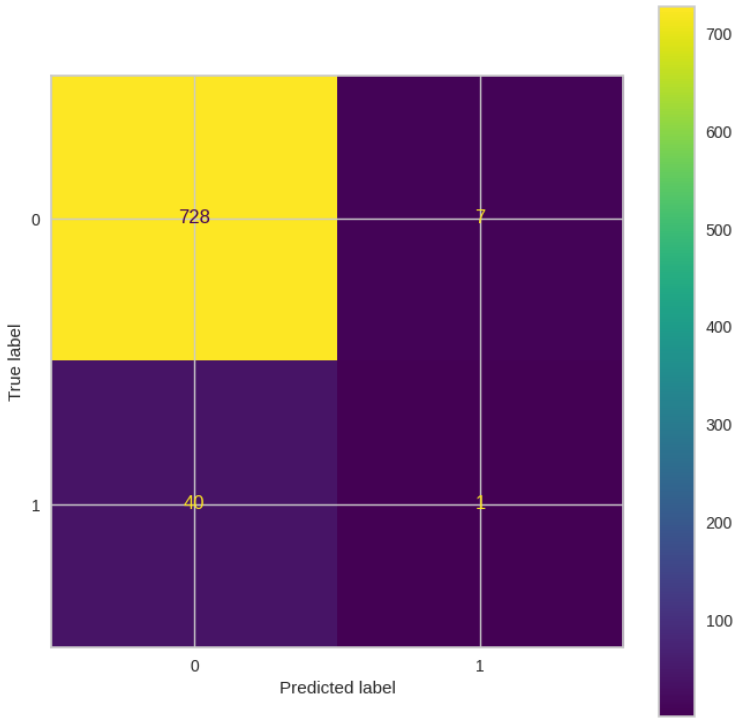
En este caso podemos ver claramente que el modelo predice todas las clases como 0, aparentemente parece que no tenga ningún criterio para decidir, aunque si lo tiene.

5.8 Random Forest

Con este método no lineal también vamos a entrenarlo para ver si hay forma de conseguir mejores resultados. Usaremos tanto el método con las clases balanceadas como sin. Primero vamos a ver sin balancear y a posterior el balanceado.

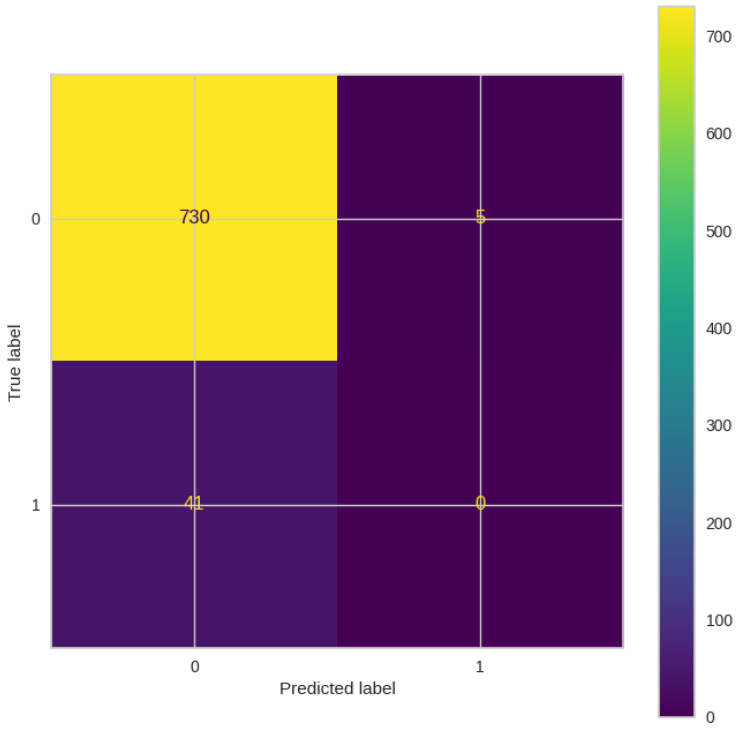
	precision	recall	f1-score	support
0	0.99	0.95	0.97	768
1	0.02	0.12	0.04	8
accuracy			0.94	776
macro avg	0.51	0.54	0.50	776
weighted avg	0.98	0.94	0.96	776

Seguidamente veremos la matriz de confusión:



Los resultados son muy malos, a continuación vamos a ver la versión balanceada.

	precision	recall	f1-score	support
0	0.99	0.95	0.97	771
1	0.00	0.00	0.00	5
accuracy			0.94	776
macro avg	0.50	0.47	0.48	776
weighted avg	0.99	0.94	0.96	776



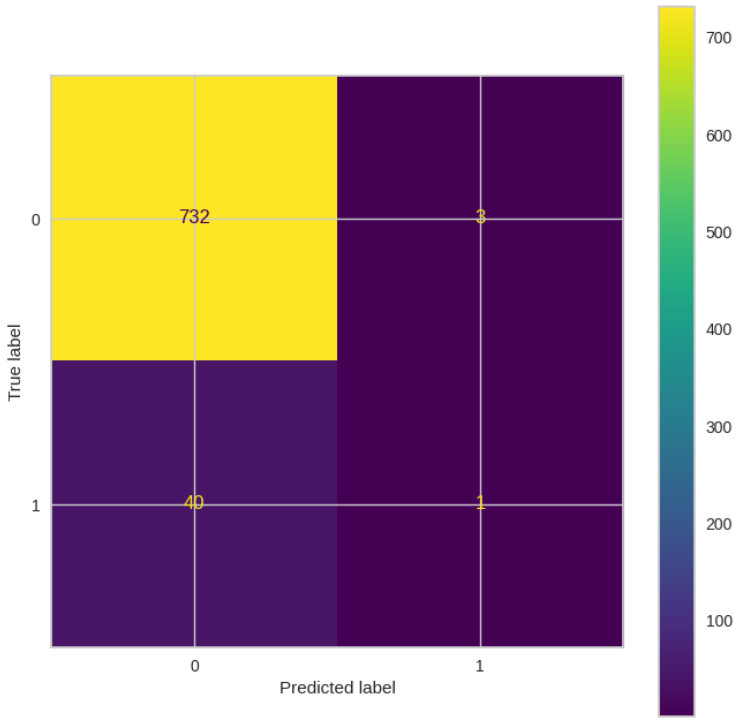
Aún obtenemos peores resultados. Parece que con este dataset será difícil de obtener buenos resultados. Veremos si el último método de la práctica nos da una agradable sorpresa.

5.9 Gradient Boosting

Gradient Boosting es un modelo que se construye aditivamente, cada modelo se especializa en el residuo del árbol anterior. Teniendo en cuenta esto, no tiene sentido manipular los pesos de las clases como en el apartado anterior, esto se hace ya automáticamente, si hay una clase más difícil el siguiente modelo le dará más peso.

	precision	recall	f1-score	support
0	1.00	0.95	0.97	772
1	0.02	0.25	0.04	4
accuracy			0.94	776
macro avg	0.51	0.60	0.51	776
weighted avg	0.99	0.94	0.97	776

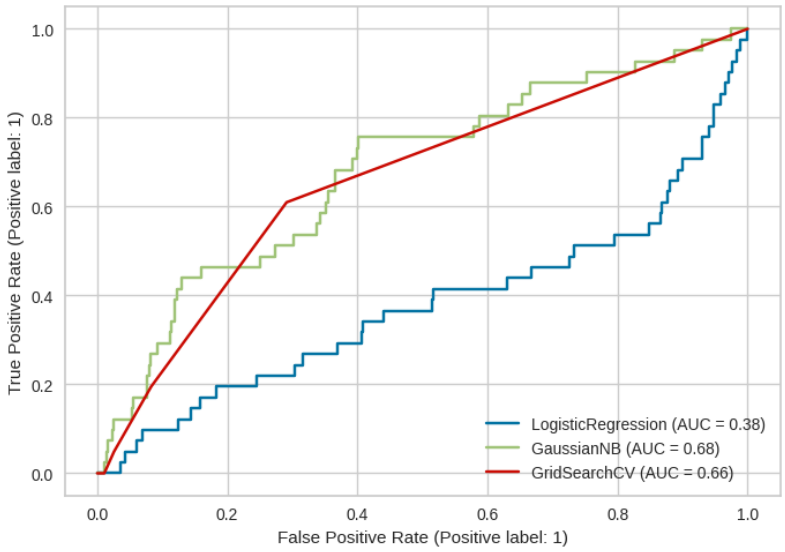
Seguidamente veremos la matriz de confusión:



Desgraciadamente, no mejoramos los resultados y seguimos en la misma línea que en todos los modelos, pocas predicciones correctas para la clase 1.

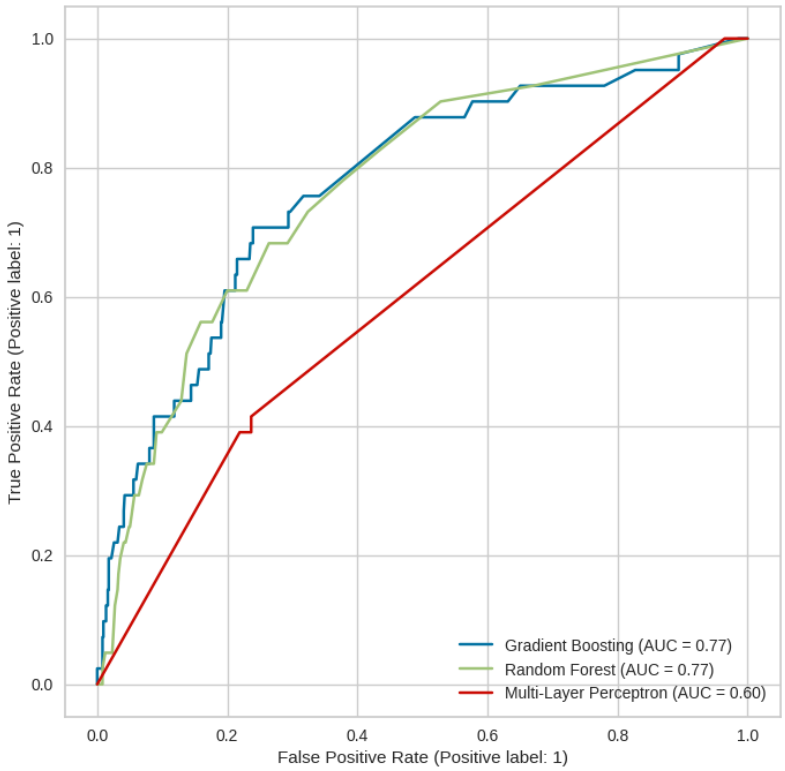
6 Interpretación de los modelos y elección

A priori podemos decir que el Naive Bayes parece que es el único que acierta terremotos sin ser casos anecdóticos y que podría ser el mejor de todos ellos, pero antes vamos a ver las curvas ROC, empezando por los métodos lineales.



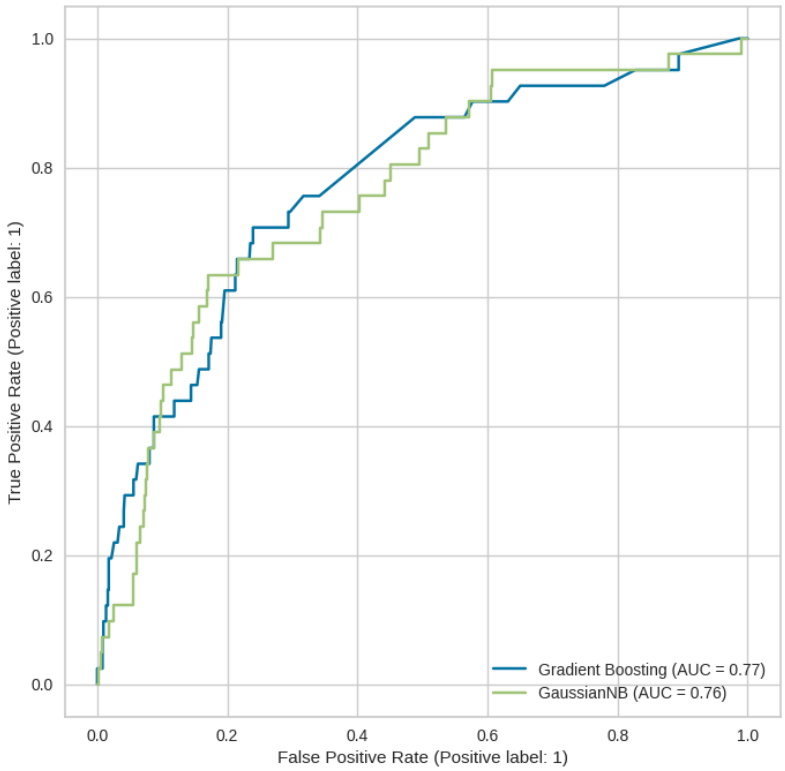
En esta comparativa podemos ver que regresión logística queda apartada, ya que está por debajo de la diagonal, y compiten Naive Bayes y KNN, pero viendo la predicción de la clase 1 y que mayoritariamente tenemos su curva por encima, pensamos que el mejor de los lineales sería el modelo Naive Bayes.

Así vamos a hacer lo mismo con los métodos no lineales.



Como en los modelos anteriores podemos ver que hay uno que claramente està descartado, como es el Multi-Layer Perceptron. Asi, aunque parecen muy parecidos creemos que Gradient Boosting mejora sustancialmente al resto de modelos.

Así entramos a dar el veredicto final, entre Naive Bayes y Gradient Boosting.



Las curvas son muy parecidas, en según qué punto está una u otra por encima, pero viendo el reporte de clasificación nos decantamos por el Naive Bayes. No es que sea buena opción, pero sí que es la mejor de las analizadas.

7 Conclusiones

El objetivo de esta práctica era hacer un estudio sobre predecir los terremotos a partir de sismos previos al mismo. Eso no es nada fácil, si no no habría las catástrofes que hay de vez en cuando. Para ello hemos intentado aplicar tres métodos lineales o cuadráticos y tres de no lineales, ninguno de ellos nos ha parecido que tenga una fiabilidad aceptable por el nivel del problema al que tenemos enfrente. Pero hemos probado diferentes métodos y ninguno de ellos no ha satisfecho lo suficiente. Al final hemos sacado la conclusión de que el mejor modelo lo ha generado Naive Bayes, pero sin dejarnos contentos.

Como comentábamos, la predicción no es buena, y no hemos sabido mejorar estos resultados que no nos terminan de gustar. Pensando en que puede fallar, se nos viene a la cabeza diferentes opciones. La primera es que no hay suficientes datos en el dataset con la clase 1. Por otro lado, nos surge la duda de si con más datos realmente conseguiríamos mejorar o los modelos seguirían con los mismos problemas, ya que durante problemas previos a esta práctica hemos visto datasets con características aparentemente similares y con mejores predicciones. Y por último nos surge la duda de si faltarían atributos que cogieran importancia en la creación del modelo.

Personalmente, los dos coincidimos en que este dataset nos ha dado mucha guerra y como consecuencia hemos aprendido mucho, ya que hemos hecho un gran esfuerzo en intentar mejorar los resultados, a diferencia en otros casos que el dataset te facilita el trabajo.

8 Referencias y estudio bibliográfico

Utilizamos citación en formato APA para esta práctica:

1. UCI Machine Learning Repository: seismic-bumps Data Set. (s. f.). Recuperado 24 de octubre de 2022, de http://archive.ics.uci.edu/ml/datasets/seismic-bumps
2. Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., & Mueller, A. (2015). Scikit-learn: Machine learning without learning the machinery. GetMobile Mobile Computing and Communications, 19(1), 29–33.
3. Bilogur, A. (s/f). missingno: Missing data visualization module for Python.
4. Seaborn: Statistical data visualization — seaborn 0.12.1 documentation. (s/f). Pydata.org. Recuperado el 5 de diciembre de 2022, de https://seaborn.pydata.org/
5. Pandas. (s/f). Pydata.org. Recuperado el 5 de diciembre de 2022, de https://pandas.pydata.org/
6. Introduction — statsmodels. (s/f). Statsmodels.org. Recuperado el 5 de diciembre de 2022, de https://www.statsmodels.org/stable/index.html
7. Bisong, E. (2019). NumPy. En Building Machine Learning and Deep Learning Models on Google Cloud Platform (pp. 91–113). Apress.
8. Matplotlib — visualization with python. (s/f). Matplotlib.org. Recuperado el 5 de diciembre de 2022, de https://matplotlib.org/
9. Yellowbrick: Machine Learning Visualization — Yellowbrick v1.5 documentation. (s/f). Scikit-yb.org. Recuperado el 5 de diciembre de 2022, de https://www.scikit-yb.org/en/latest/
10. Surhone, L. M., Timpledon, M. T., & Marseken, S. F. (Eds.). (2010). Scipy. Betascript Publishing.
11. haloboy777. (n.d.). Haloboy777/arfftocsv: Arff to CSV converter (python). GitHub. Recuperado 27 de diciembre de 2022, de https://github.com/haloboy777/arfftocsv