A continuación, vamos a comentar brevemente cada uno de los modelos que hemos aplicado sobre nuestro dataset, centrandonos en la elección de los parámetros de cada uno de los modelos y comentando los resultados que obtenemos con cada uno de ellos aplicándolos a los dos datasets que hemos generado, el original y el limpio.

## SVM

### Configuration of the model. Selecting parameters

Existen varios tipos de kernels para aplicar a los modelos de SVM. En concreto, para realizar pruebas y comprobar cuál de ellos se ajusta major a nuestros datos, vamos a utilizar tres de ellos:

* Lineal
* Polinómico
* Radial

Todos ellos los aplicaremos sobre los dos datasets que hemos generado, el original, con todos los datos, outliers y clases desbalanceadas; y el limpio, sin outliers, eliminando la columna de “SKINTHICKNESS” y con las clases balanceadas. Esto lo haremos para comprobar si al limpiar el dataset obtenemos mejores resultados con este modelo.

Sabemos que las clases de nuestro dataset no son linealmente separables, aún así, vamos a aplicar el modelo con el kernel lineal, para comprobar si es capaz de separar las clases utilizando márgenes suficientes para ayudarle a clasificar.

* *TEST WITH ORIGINAL DATASET* 
  + Kerne lineal

En primer lugar, vamos a aplicar un modelo SVM con kernel lineal. Para ello, en primer lugar definimos las variables de entrada y de salida de nuestro modelo. Para esta primera prueba, vamos a utilizar el dataset original con todas sus columnas.

INPUTS\_ALL = ['PREGNANT','GLUCOSE','BLOODPRESS','SKINTHICKNESS','INSULIN','BODYMASSINDEX','PEDIGREEFUNC','AGE']

OUTPUT = 'DIABETES'

Una vez elegidas las variables de entrada y el dataset que vamos a utilizar, vamos a definir los parámetros de nuestro modelo. Utilizaremos GridSearch para aplicar varios parámetros en nuestro modelo y quedarnos con el que mejores resultados nos aporte.

En este caso, vamos a probar con los siguientes valores de coste:

|  |
| --- |
| [0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000] |

Con estos valores especificamos el numero de puntos que pueden estar mal dentro de nuestras predicciones. Cuando el valor de nuestro coste es pequeño, le estamos indicando al modelo que hay menos vectores soporte que pueden ser utilizados para determinar el hiperplano. Cuando el valor es más grande, estamos incluyendo más puntos para determinar dicho hiperplano.

Comprobando el accuracy de este modelo en train, vemos que obtenemos un valor de 0.76708.

Una vez entranado el modelo, podemos obtener los coeficientes de las variables de nuestro dataset, para ver a cuál de ellas le ha dado más importancia:

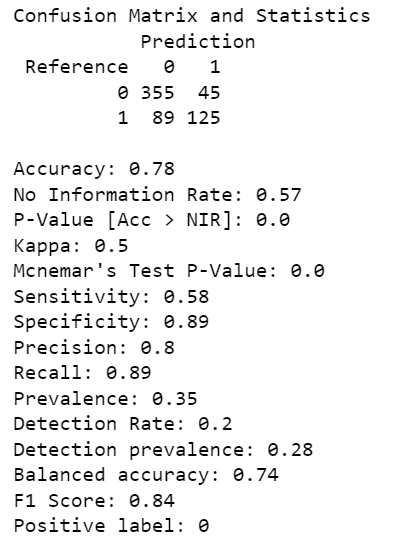
Gráfico

Descripción generada automáticamente con confianza media

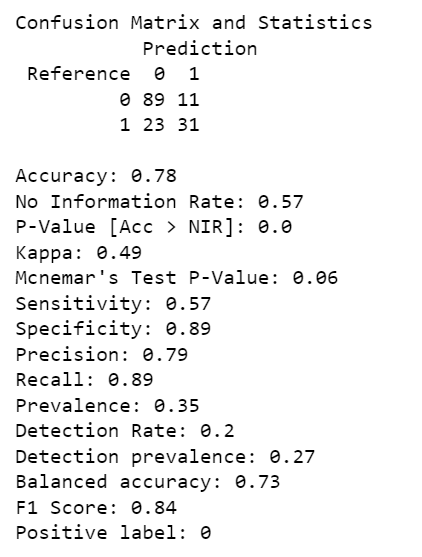
Como Podemos ver en la gráfica, las variables que más importancia tienen para este modelo son ‘GLUCOSE’, ‘BMI’, ‘PREGNANT’ y ‘PEDIGREEFUNC’.

Una vez averiguadas estas variables, vamos a generar las matrices de confusion tanto de train como de test, para ver cuánto accuracy tiene nuestro modelo.

La matriz de confusion de train:



La matriz de confusion de test:



Como podemos ver, tenemos un accuracy de un 0.78 en test.

To compare these two models we will look at the calibration plot and the ROC curve.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* + Kernel Polinómico

Aplicando este tipo de kernel al modelo, Podemos llegar a separar clases que no sean linealmente separables con una mayor precision.

Al igual que con el kernel anterior, utilizamos las mismas variables y el mismo dataset para entrenar nuestro modelo.

A su vez, en este modelo tenemos que indicar una serie de hiperparámetros más, para que el modelo sea capaz de predecir correctamente.

En primer lugar, debemos indicar los grados de la función polinómica que queremos aplicar. EN nuestro caso, hemos elegido una función de grado 8, puesto que es el numero de variables que estamos tratando en nuestro dataset.

Otro hiperparámetro que debemos indicar es el gamma, que delimita el ancho de la función, nos permite ver efectos más locales. A su vez, al igual que en el kernel lineal, debemos indicar el coste de la función, para especificar el numero de vectores que necesitamos para definir nuestro hiperplano.

En resumen, los parámetros que hemos seleccionado para este modelo son:

|  |
| --- |
| C: [0.0001,0.001,0.01,0.1,1,10,100]  Gamma:[0.001,0.01,0.1,1,10]  Degree: 8 |

Para este modelo con estos parámetros obtenemos un accuracy del 0.6922.

Una vez entrenado el modelo, podemos ver las gráficas de los errores, y cuál ha sido el modelo que ha elegido el método de cross validation como mejores parámetros:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico radial, Gráfico de líneas

Descripción generada automáticamente

Vemos que el major modelo es el que tiene coste 10 y un gamma de 0.1, parámetros lo suficientemente grandes como para abarcar un numero significativo de vectores que definan el hiperplano.

Una vez seleccionado el major modelo, Podemos generar las matrices de confusion, para comprobar cuál es el accuracy del modelo en los conjuntos de train y test.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Podemos ver que para nuestro modelo, obtenemos un 0.83 de accuracy en train y un 0.73 en test.

Pasamos ahora a pintar la curva roc de este modelo. Se puede observer que no aporta muy buenos resultados:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* + Kernel Radial

Por ultimo, vamos a configurar los parámetros del mismo modelo, entrenandolo con los mismos datos, pero esta vez usando un kernel lineal. En este caso, no es necesario indicar el numero de grados, puesto que lo que va a hacer este modelo es separar las clases mediante un hiperplano ‘circular’.

Los parámetros que hemos seleccionado para este modelo son:

|  |
| --- |
| C: [0.0001,0.001,0.01,0.1,1,10,100]  Gamma:[0.001,0.01,0.1,1,10] |

Al igual que en los modelos acteriores, realizaremos un grid search para hacer un cross validation y seleccionar los parámetros con los que major se ajusta nuestro modelo. Podemos ver que el modelo nos aporta un accuracy del 0.7622.

Una vez entrenado el modelo, podemos ver las gráficas de los errores, y cuál ha sido el modelo que ha elegido el método de cross validation como mejores parámetros.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos ver que el cross validation ha elegido como major modelo a aquél con uncoste de 100 y un gamma de 0.01, algo distinto de nuestro modelo polinómico

* *TEST WITH MODIFICATED DATASET*

Vamos a utilizar exactamente los mismos parámetros descritos antes para todos los modelos, lo único que cambiará ahora es que en lugar de utilizar el dataset original, vamos a aplicar los modelos sobre el dataset limpio, sin outliers, con clases balanceadas, y excluyendo la columna ‘SKINTHICKNESS’ que, como ya hemos comentado, no nos aporta buenos resultados ya que la mayoría de datos son erróneos.

Las variables que vamos a utilizar son :

INPUTS\_ALL = ['PREGNANT','GLUCOSE','BLOODPRESS' ,'INSULIN','BODYMASSINDEX','PEDIGREEFUNC','AGE']

OUTPUT = 'DIABETES'

* + Kernel lineal

Utilizando exactamente los mismos parámetros que con el dataset original, obtenemos los siguientes resultados:

* + - Accuracy de 0.7581.
    - Importancia de las variables: En el gráfico Podemos ver que ahora nuestro modelo da más importancia a todas las variables que tenemos en el dataset, al haber eliminado la columna ‘SKINTHICKNESS’, tenemos menos ruido en los datos y Podemos ser mas precisos con nuestro modelo.

Gráfico, Gráfico de barras

Descripción generada automáticamente

* + - Matrices de confusion:
    - The calibration plot and the ROC curve: Podemos ver que el área bajo la curva y los resultados que nos muestran en general las gráficas son mejores que

Gráfico, Gráfico de líneas

Descripción generada automáticamente Gráfico, Histograma

Descripción generada automáticamente Gráfico, Gráfico de líneas

Descripción generada automáticamente Gráfico, Gráfico de líneas

Descripción generada automáticamente

* + Kernel Polinomico

Utilizando exactamente los mismos parámetros que con el dataset original, unicamente cambiando el grado del polinomio de 8 a 7 (numero de variables que tiene ahora nuestro dataset al haber eliminado la columna), obtenemos los siguientes resultados:

* + - Accuracy de 0.7768.
    - Gráficos del error: Ejecutando un cross validation con 5 folds, obtenemos que el mejor modelo es el que utiliza un coste de 100 y un gamma de 0.1, parámetros bastante grandes para obtener un abanda más ancha.

Gráfico, Gráfico de líneas

Descripción generada automáticamente Gráfico

Descripción generada automáticamente

* + - Matrices de confusion:

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza media Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

* + Kernel radial

Utilizando exactamente los mismos parámetros que con el dataset original, obtenemos los siguientes resultados:

* + - Accuracy de 0.8062.
    - Gráfica de errores: Nos indica que el major modelo para este kernel es el de coste 10 y gamma 0.1

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* + - Matrices de confusion:

Una captura de pantalla de un celular

Descripción generada automáticamente Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

* + - The calibration plot and the ROC curve: Podemos ver que el modelo se ajusta bastante a los datos, y que el area bajo la curva roc es bastante grande. Este es el mejor modelo

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

### Results and conclussions

Comparando los modelos para los mismos datasets, podemos ver que, en general, el modelo polinómico no se comporta muy bien con este set de datos, mientras que el radial y el lineal modelan bastante bien los datos.

Los resultados de las puntuaciones y curvas roc para el dataset original son los siguientes:

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Los resultados de las puntuaciones y curvas roc para el dataset limpio son los siguientes:

## Multi-Layer Perceptron

### Configuration of the model. Selecting parameters

Para este modelo, al igual que en todos los anteriores, vamos a probar una serie de parámetros, en este caso, capas de las neuronas y penalización del momento (Alpha), para que, mediant el grid search con cross validation, seamos capaces de identificar los mejores parámetros para nuestro modelo.

En este caso, también hemos entrenado el modelo con los dos datasets que tenemos disponibles, el original y el limpio, para poder comparar los resultados de ambos y ver cómo se comportan los modelos con los diferentes datasets.

* *TEST WITH ORIGINAL DATASET*

Para entrenar el modelo con el dataset original, vamos a configurar las variables de entrada y salida como se indica en la tabla

INPUTS\_ALL = ['PREGNANT','GLUCOSE','BLOODPRESS','SKINTHICKNESS','INSULIN','BODYMASSINDEX','PEDIGREEFUNC','AGE']

OUTPUT = 'DIABETES'

Además, de cara a configurar el modelo, vamos a hacer pruebas con las siguientes estructuras de redes neuronales, así como diferentes penalizaciones de L2.

|  |
| --- |
| MLP\_alpha: [1e-9,1e-7,1e-5,0.001,0.01]# L2 regularization term  MLP\_hidden\_layer\_sizes':[(5,),(10,),(15,),(5,5),(10,10),(15,15) |

Con estos parámetros, el método de Grid search con cross validation ha elegido el major modelo con los parámetros alpha = y una sola capa occulta con 5 neuronas.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*Gráfico, Gráfico de líneas

Descripción generada automáticamente*

En las siguientes gráficas se muestra la sensibilidad del modelo. En la primera de ellas Podemos ver la importancia que el modelo le ha asignado a cada una de las variables de nuestro dataset. Como se puede ver, la variable ‘SKINTHICKNESS’ ha tomado un papel importante en el modelo, cuando sabemos que la mayoría de sus datos son erróneos. Esto a la hora de predecir nos puede llegar a penalizar.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Las matrices de confusion que hemos obtenido para el major modelo son:

Texto

Descripción generada automáticamente Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

Podemos ver que tenemos un 0.83 de accuracy en entrenamiento y un 0.81 en test.

A continuacion se muestran los gráficos de calibracion y la curva roc, para observar cómo de bueno es el modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamenteGráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos ver que el area bajo la curva roc es de 0.87, lo cual es bastante buen resultado para este modelo.

* *TEST WITH MODIFIED DATASET*

Para la misma configuración del modelo, tanto las capas ocultas como los terminos de regularización, vamos a aplicarlo al dataset limpio, con las clases balanceadas y sin outliers.

En este caso, el grid search con cross validation ha elegido como major modelo a la red neuronal con dos capas ocultas de 15 neuronas cada una, y el mismo alpha que en para el modelo anterior, es decir alpha = .

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

A continuación se muestran los resultados de la sensibilidad del modelo, indicando cuál de las variables ha sido más significative en la prediccion.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Podemos ver que PEDIGREEFUNC, PREGNANT y BLOODPRESSURE son las variables que más han ayudado al modelo.

Las matrices de confusion resultants de este modelo son las siguientes:

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

Podemos ver que en entrenamiento tenemos un 1 de accuracy, esto nos puede estar indicando que es un modelo sobreentrenado, aunque el cross validation lo haya elegido como major modelo. En test Podemos ver que no tenemos un accuracy demasiado alto, unicamente de 0.68, lo que nos hace sospechar que es un modelo sobreentrenado.

Viendo las curvas de calibración y la ROC, Podemos deducir que, aunque este modelo tenía las mismas opciones que el otro, el cross validation ha dado peores resultados.

Gráfico

Descripción generada automáticamente con confianza baja Gráfico, Histograma

Descripción generada automáticamente Gráfico, Gráfico de líneas

Descripción generada automáticamente

### Results and conclussions