**Práctica 6. Aprendizaje automático en la práctica**

**Datos de entrenamiento**

En esta práctica se utilizarán datos artificiales obtenidos en la siguiente función proporcionada en el enunciado de la práctica ***gen\_data()***:

A screen shot of a computer program

Description automatically generated

Si se muestra la gráfica:

A graph with blue dots and red lines

Description automatically generated

Como el método ***gen\_data()*** devuelve una lista unidimensional y las funciones de ***scikit-learn*** que se usarán necesitan arrays bidimensionales se deberá de añadir una nueva dimensión siempre que sea necesario.

**Sobreajuste a los ejemplos de entrenamiento**

A continuación, se muestra el código donde se realiza el sobreajuste a los ejemplos de entrenamiento junto con su respectiva explicación:

**A screen shot of a computer program

Description automatically generated**

Para poder generar el error del modelo tanto de los datos de entrenamiento como los datos de prueba se aplicará la siguiente fórmula:

A mathematical equation with numbers and symbols

Description automatically generated

Traducido a código será de la siguiente forma:

A black background with red text

Description automatically generated

Una vez realizado todo el proceso de sobreajuste se obtienen los siguientes valores de error, tanto para el entrenamiento como de prueba:

* Error on training set: 11855.0447622640992
* Error on test set: 48579.4289101656

A graph of a graph with blue and orange dots

Description automatically generated

**Elección del grado de polinomio usando un conjunto de validación**

Generando los datos de la misma forma que en el apartado anterior esta vez usaremos el 60% de los datos para entrenamiento, el 20% para validación y el restante 20% para prueba.

Entrenando modelos con transformaciones polinómicas de grado 1 a 10 deberíamos encontrar

que el de menor error genera predicciones.

Con todo esto la función en la que se obtiene el grado óptimo del polinomio es la siguiente:

A computer screen shot of a program code

Description automatically generated

El resultado de esto, uniéndolo con el método del apartado anterior es el siguiente:

* Best polynomial degree: 2
* Error on training set: 14877.96905472397
* Error on test set: 28209.12655628797

A graph of a function

Description automatically generated with medium confidence

**Elección del parámetro λ**

Se utilizará una transformación polinómica de grado 15 y evaluaremos sobre el conjunto de validación cruzada los valores de *λ*.

Una vez explicados los valores que tendrán el grado del polinomio y el parámetro de regularización se desarrolla el método:

A screen shot of a computer program

Description automatically generated

Proporcionando la mejor lambda: 1e-06

**Elección de hiper-parámetros**

A continuación se implementará una búsqueda exhaustiva de la mejor combinación de grado del polinomio y valor de λ para un conjunto de 750 valores, usando el 60% de los datos para entrenamiento, el 20% para validación y el restante 20% para prueba.

Probando con polinomios hasta grado 15 y valores de λ se desarollará el método ***hyperparameter\_tuning()***:

A computer screen shot of a program code

Description automatically generated

Proporcionando el siguiente resultado:

* Best polynomial degree: 4
* Best lambda: 1e-06
* Error 28288.38408286469

**Curvas de aprendizaje**

Entrenaremos un modelo con polinomios de grado 16 sin regularizar con conjuntos crecientes

de datos de entrenamiento, generando entre 50 y 1000 ejemplos y usando en cada caso el 60% de los datos para entrenamiento, el 20% para validación y calculando en cada caso el error de entrenamiento y el de validación para generar una gráfica:

A graph with blue lines and orange lines

Description automatically generated

El código será el siguiente:

A screen shot of a computer program

Description automatically generated