Código

El único fichero necesario para la generación de resultados es *main.py*. Este fichero cargará los modelos entrenados y datos precalculados y automáticamente construirá los dataset de predicción, generará las predicciones y las guardará en el fichero *.csv* de salida.

Los modelos y datos precalculados están guardados en la carpeta *./models/*.

La carpeta *./additional_files/* contiene los ficheros necesarios para crear un dataset de entrenamiento y entrenar los modelos. Pero estos ficheros no son necesarios para la ejecución de *main.py*.

Entorno de ejecución

El código enviado ha sido desarrollado y probado en un *Ubuntu 18.04* con *Python 3.6.8*. Para una ejecución sencilla, se ha exportado el entorno de *conda* utilizado, que contiene todas las librerías y versiones necesarias.

Para importar el entorno (llamado *aguathon_053*), ejecutar:

conda env create -f environament.py

Para ejecutar la predicción ejecutar:

```
source activate aguathon_053
python main.py
```

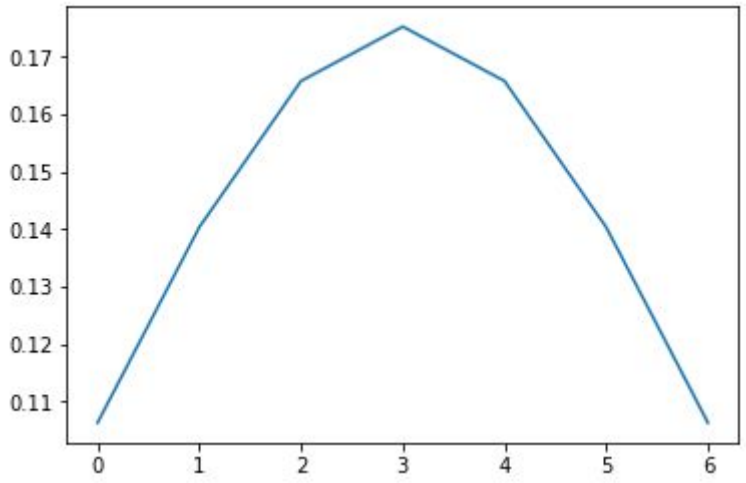
Construcción del dataset

Para la obtención de unos resultados óptimos, se ha construido un **dataset** formado por una serie de features que buscan facilitar la búsqueda de patrones en el flujo del agua entre los diferentes puntos de muestreo así como detectar cambios anómalos en lo que se consideraría un caudal normal dado un momento temporal.

Para detectar caudales fuera de lo normal, previamente se ha definido lo que se considera un caudal normal. Así, dado un instante temporal (día, mes y hora, no año), un caudal normal se define como la media ponderada de caudales registrados en ese instante junto con los instantes vecinos.

Estos instantes vecinos comprenden los caudales registrados en días cercanos al instante temporal formado por el día y el mes, y para cada uno de estos días, los caudales registrados en las horas contiguas.

Adicionalmente, ésta es una media ponderada por una Campana de Gauss, que busca que los elementos cercanos al instante temporal original tengan más importancia que los más lejanos.



Muestra de una función gaussiana

Para detectar el flujo de agua, el dataset incluye features formadas por la diferencia entre diferentes puntos de registro de caudal y diferentes momentos temporales, con el fin de detectar posibles crecidas bruscas del nivel del agua.

Modelo de predicción

Tras la ejecución de diferentes pruebas, el algoritmo de predicción elegido ha sido **LightGBM**, un modelo basado en *Gradient Boosting* y *decision trees*. Esta decisión ha sido determinada por las características de esta competición, en la que se valora que la obtención de resultados sea rápida.

Este algoritmo de reciente aparición, a diferencia de otros similares, construye los árboles de decisión siguiendo un crecimiento por hoja, en lugar de por nivel. Esto se traduce en modelos más pequeños, rápidos y con menos necesidad de memoria RAM, pero igual de competitivos en resultados como lo pueden ser XGBoost o Catboost.

Con el fin de obtener resultados óptimos, se ha entrenado un modelo que se especialice en cada tipo de predicción, 24h, 48h y 72h. Adicionalmente, el input de cada modelo es un dataset, como el descrito anteriormente, del que se han eliminado las variables que no aportan información a ese tipo de predicción. Esto se traduce en una obtención de resultados más rápida (tanto en entrenamiento como en predicción) y una mayor precisión del sistema, ya que el algoritmo sólo se centra en encontrar patrones entre las variables que realmente aportan información valiosa.

Técnicas de Machine Learning descartadas

Siguiendo la filosofía de la construcción un modelo robusto pero rápido, se han descartado otras técnicas de Machine Learning que mejoran ligeramente los resultados como *K-fold cross-validatio*n o *Model Stacking*. Ambas opciones han sido descartadas porque de una forma u otra, requieren del

uso de diferentes modelos para la obtención de resultados. Este significativo aumento del tiempo de cálculo no se ha considerado que justifique la diferencia de precisión en los resultados finales.