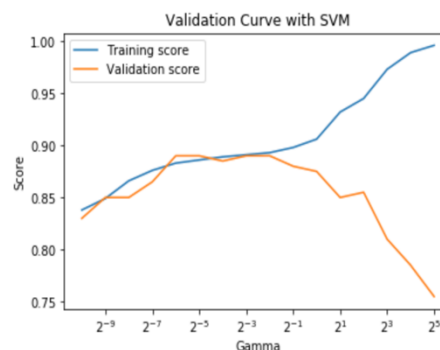


## Tarea 4 – Raimundo Herrera

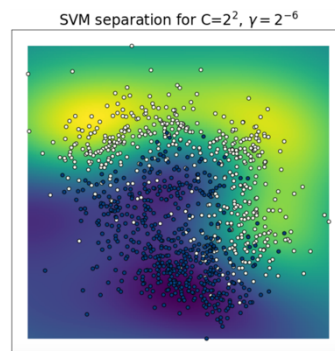
En esta tarea el objetivo es experimentar con dos clasificadores, *Support Vector Machines* (SVM), y Redes Neuronales. Lo que se pretende es, dado un *set* de datos que ya pasó por el proceso de transformación y selección de características, seleccionar los hiperparámetros y configuraciones que otorgan, para cada modelo, una mejor clasificación dadas las clases del problema en cuestión. La discriminación a realizar es de 2 clases con 2 características, y el conjunto de datos es el mismo para SVM y para redes neuronales.

Para ambos casos se utilizaron las implementaciones de los clasificadores presentes en la librería *sklearn*. En el caso de las redes neuronales, los datos se normalizaron en entrenamiento dado que las redes neuronales utilizadas (*Multi Layer Perceptron*) son sensibles al escalamiento de datos<sup>1</sup>.

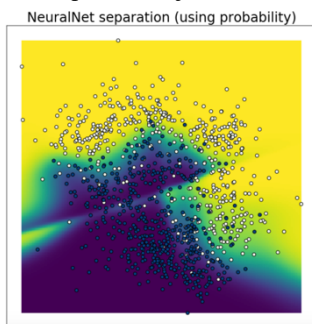
Asimismo, para los dos experimentos se utilizaron técnicas de búsqueda exhaustiva de parámetros. En primer lugar se visualizaron curvas de rendimiento entre los *sets* de validación y entrenamiento entregados, como la imagen de la derecha, para ver cuando el cambio los de parámetros mejoraba el rendimiento en ambos conjuntos. En dichas curvas es esperable elegir los parámetros que maximicen validación antes de sobreajustar a entrenamiento.



Para SVM los parámetros con los que se probaron distintos valores son C y Gamma. Para ellos se utilizaron dominios de potencias de 2 para su elección<sup>2</sup>. En la ilustración *Validation Curve with SVM* se observa como el parámetro Gamma se puede modificar hasta escoger el punto de mejor rendimiento en ambas curvas. Luego se experimentó con ambos parámetros al mismo tiempo, y se incluyeron en el *notebook* de la tarea visualizaciones adecuadas en 3 dimensiones. Los resultados para dicha sección fueron de un 87.3% en test. Dichos resultados se daban con los parámetros óptimos tras la búsqueda exhaustiva de ellos, y consiguen una separación ilustrada en la figura *SVM separation for C=2<sup>2</sup>, Gamma=2<sup>-6</sup>*, que muestra como está discriminando las etiquetas reales, que vienen dadas por los colores de los puntos, en la separación dada por el modelo, que viene mostrada por el mapa de colores de fondo.



Para redes neuronales, la cantidad de parámetros es mucho mayor, por lo que la búsqueda más lenta, y la visualización de curvas como la de validación anteriormente mostrada bastante menos ilustrativa. Sin embargo, la búsqueda arrojó resultados no mucho mejores que los resultados de la red sin modificar parámetros. Esto da



lucos de que para problemas con pocas características y pocas clases utilizar redes neuronales puede no ser necesario frente a modelos más simples como SVM. Para este caso, los parámetros óptimos encontrados consideran función de activación logística, capas de 90 neuronas, y el optimizador *lbfgs*. El puntaje obtenido es de un 87.1% en test. Lo anterior presenta una mejora leve respecto a los hiperparámetros que se usaron en un primer intento, pero porcentajes leves pueden ser significativos en este campo de estudio. Por otro lado, la separación que se observa en la imagen *NeuralNet separation (using probability)*<sup>3</sup>, da lucos de que es distinto lo que se está realizando a lo que hace SVM. Se puede observar una alta no linealidad y poca suavidad, y también interpretar un cierto sobreajuste, pero mitigado por el uso de conjuntos de *test* y validación.

Finalmente es importante destacar de los experimentos que si bien los resultados son satisfactorios tras búsquedas exhaustivas de hiperparámetros, en la mayoría de los casos las mejoras no fueron significativas, lo que da lucos de que para problemas simples, los modelos por sí solos otorgan resultados buenos, sin necesidad de ajustar de sobremanera los hiperparámetros. También se muestra como los resultados dependen altamente de los datos, lo que se ve en resultados similares en modelos distintos, donde separaciones mejores no son realmente posibles sin un sobreajuste excesivo.

<sup>1</sup> Según lo comentado en: [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)

<sup>2</sup> Según lo mostrado en: <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

<sup>3</sup> Una explicación más detallada de este gráfico y el anterior se encuentra en el *Python notebook*.