Jacobo Misael Tapia de la Rosa
A01336590

Aprendizaje automático

Árboles de decisión:

- Utilizando python en conjunto el módulo 'scikit-learn', podemos utilizar los árboles de decisión como nuestro método de aprendizaje supervisado mediante el uso de clasificación y regresión. Con este módulo podemos crear un modelo que nos permita predecir el valor de una variable objetivo mediante el aprendizaje de simples reglas de decisión inferidas de los atributos de la información.
- Con lo anterior, se genera a base de un conjunto de datos, una partición de manera aleatoria en dos conjuntos, abarcando 60% para entrenamiento y 40% para el testing. Generando una visualización del árbol en el archivo 'decision_tree.png' además de un archivo csv con el título "dt_results" en la cual se tienen 2 columnas, una que es el valor de desempeño de los datos de prueba y el la segunda la predicción realizada. Dicho .csv es leído nuevamente para hacer el cálculo del error cuadrático medio, el cual llegó a tomar valores desde -17 hasta 17. Que el error varía en cada iteración puede deberse a que cada que el programa se corre, el conjunto de entrenamiento y pruebas es totalmente distinto.

Redes neuronales artificiales:

- Con el fin de que en los tres modelos se ocupen los mismos conjuntos de entrenamiento y pruebas, la función read_data fue pasada a otro archivo.
- Debido a que en python, es difícil que las redes neuronales convergen antes de el número máximo de iteraciones permitidas; esto debido a que la información no está normalizada, se debe escalar la información. En este caso utilizaremos el módulo de scikit-learn llamado Standard Scaler, el cual estandariza los atributos removiendo la media y escalando a la varianza unitaria.
- Para entrenar el modelo, usaremos el modelo Multi-Layer Perceptron Classifier
- Posteriormente instanciamos nuestro modelo, en este caso, solo definiremos el atributo 'hidden_layers_sizes', en el cual el parámetro es una tupla que consiste en el número de neuronas, que se necesitan en cada capa, en el cual la enésima entrada en la tupla, consiste en el número de neuronas en la enésima capa del modelo MLP. Los valores que se escogieron para tres capas es el mismo número de

neuronas el cual es equivalente al número de atributos en nuestro conjunto de datos. Es decir {3,3,3} con un método de activación de 'relu'. La información puede encontrarse al correr el programa con la línea "MLP information", los warnings que aparecen ahí son por el uso de scikit de antiguas versiones de numpy.

- De igual forma que en el caso de los árboles de decisión, se genera un archivo .csv con una columna del valor de performance y el valor predicho por la red.

K vecinos:

Se repite un proceso muy similar a los anteriores, únicamente se añade un archivo csv por cada 'k' solicitado.

Al final, se arma un .csv con la tabla que se indica en el punto número 6, en un archivo titulado 'global_results.csv'

Conclusiones:

- ¿Cuál es la técnica de aprendizaje automático que aporta mejores resultados?: Depende de lo que quieras lograr; sin duda una técnica simple de entrenar es los k vecinos cercanos, sin embargo, difícilmente sus predicciones son correctas pues tienden a hacer overfitting, y es difícil que se ajusten a distintos comportamientos. Por otro lado las redes neuronales aunque son muy buenas por el concepto que manejan, en este caso tienden a ser ligeramente imprecisas y tardadas en entrenar, por lo que adquieren habilidades de razonamiento, se emplea mucho tiempo en lograr eso. Finalmente para este ejercicio los árboles de decisión arrojan buenos resultados, pues a pesar de que perdemos algo de rendimiento con ellos, tienen una gran capacidad predictiva y son precisos gracias a que están representando las relaciones no lineales para resolver el problema, además son fáciles de interpretar debido a sus reglas de decisión.
- ¿Qué pasa cuando se utiliza el promedio de las predicciones?: Se llega a un número que va acorde o se encuentra en un intermedio entre las distintas predicciones, sin embargo dicho número; en la mayoría de las veces se acerca mucho al valor original que se quería evaluar, en este caso, fueron datos muy cercanos al Performance dado en el set de datos inicial.
- ¿Qué puedes decir acerca del sobreajuste(overfitting)?: Debido a que en machine learning queremos encontrar patrones que podamos generalizar para poder realizar predicciones sobre aquella información que aún no se analiza, sin embargo estos

patrones pueden aparecer desde el entrenamiento debido a casualidades en los datos, llevando a no tener una generalización; a esto le llamamos sobreajuste. Este sobreajuste lo tienen distintos algoritmos, teniendo a ajustarse a diversas características de los datos de entrenamiento que no tienen relación con la función objetivo que deseamos generalizar. Un ejemplo es entrenar algo con solo respuestas correctas, cuando quiera predecir algo, su comportamiento será azaroso pues nunca logró llegar a un patrón que pudiera predecir. Para evitarlo hicimos una práctica en este ejercicio que fue dividir los datos en entrenamiento y pruebas, permitiendo realizar una evaluación de la efectividad del modelo, evitando así que los mismos datos para entrenar, funcionen para evaluar; dándonos a conocer cómo se va comportando un modelo en relación a su entrenamiento.

Código fuente:

- https://github.com/Jatapiaro/AprendizajeAutomatico_Actividad

.