Fundamentos de Aprendizaje Automático 2020/2021

PRÁCTICA Nº 2

1. Objetivo

En esta práctica continuaremos el estudio de los clasificadores implementando dos nuevos algoritmos: *vecinos próximos* y *regresión logística*. Siguiendo la filosofía de la anterior práctica se deben comparar sus resultados con los que proporciona la librería de *scikit-learn* (http://scikit-learn.org/stable/). También se continuará aplicando el análisis ROC para comparar clasificadores.

2. Tareas

La planificación temporal sugerida y las tareas a llevar a cabo son las siguientes:

- *1ª semana:* Implementar el algoritmo de *vecinos próximos* (clase **ClasificadorVecinosProximos**) para realizar una tarea de clasificación de los siguientes conjuntos de datos, probando con diferentes valores de vecindad (K=1, 3, 5, 11 y 21). Ambos conjuntos se encuentran en Moodle.

  - ✓ Conjunto de datos pima-indians-diabetes (https://www.kaggle.com/kumargh/pimaindiansdiabetescsv).
  - ✓ Conjunto de datos ***wdbc*** (http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/).

  En el uso del algoritmo de vecinos próximos es recomendable normalizar los atributos. La normalización es importante puesto que evita problemas de escala. Para llevar a cabo la normalización se añadirán nuevos métodos, tal y como se describe en el apartado 3. Aparte de la normalización y el valor de K, un aspecto importante en vecinos próximos es la métrica o distancia utilizada. En la implementación propia se debe trabajar con las siguientes distancias: Euclídea, Manhattan y Mahalanobis. Para calcular la última puedes aprovechar los distintos métodos que proporciona numpy, como *cov* o *Mahalanobis*.

- *2ª semana*: Implementar el algoritmo de *regresión logística* (clase **ClasificadorRegresionLogistica**), aplicando el método de maximización de la verosimilitud visto en las clases de teoría. Comparar los resultados obtenidos para los conjuntos de datos ***indians*** y ***wdbc*** con el algoritmo de *regresión logística* y con el de *vecinos próximos*.

- *3ª semana*: Analizar los dos conjuntos de datos anteriores con la librería de scikit-learn para vecinos próximos y con la correspondiente para regresión logística, según los detalles del apartado 4.

- *4ª semana*: Ahora que se han implementado tres clasificadores se puede extender el Análisis ROC que iniciamos en la primera práctica. Para usar los mismos datos, debes ejecutar Naive Bayes con los dos nuevos conjuntos: indians y wdbc. Posteriormente realiza el Análisis ROC para comparar los resultados en los conjuntos indians y wdbc con los tres clasificadores.

3. Normalización de datos

Para la normalización de los datos, se implementarán dos métodos nuevos en la clase que consideres más apropiada, justificando la elección:

- `calcularMediasDesv(self,datos,nominalAtributos)`: esta función calculará las medias y desviaciones típicas de uno o varios atributos continuos según lo que se pase en el argumento `datos`.
- `normalizarDatos(self,datos,nominalAtributos)`: esta función normalizará cada uno de los atributos continuos en la matriz `datos` utilizando las medias y desviaciones típicas obtenidas en `calcularMediasDesv`.

4. Scikit-learn

Vecinos Próximos

Scikit-learn implementa dos clasificadores basados en vecindad. Para los propósitos de esta práctica interesa ***KNeighborsClassifier***. Un argumento de este método es **metric**, que permite indicar la distancia a utilizar. Scikit-learn permite una gran variedad de métricas que puedes consultar en https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html#sklearn.neighbors.DistanceMetric. Las tres que interesan en esta práctica son las mismas que incluiremos en la implementación propia: Euclídea, Manhattan y Mahalanobis.

Regresión Logística

Scikit-learn proporciona una implementación de la *Regresión Logística* a través de los métodos ***LogisticRegression*** y ***SGDClassifier***. Revisa la información sobre estos métodos para comprender bien las diferencias entre la implementación propia y la que hace Scikit Learn de este algoritmo.

5. Fecha de entrega y entregables

Semana del 23 al 27 de Noviembre de 2020. La entrega debe realizarse antes del comienzo de la clase de prácticas correspondiente. Se deberá entregar un fichero comprimido .zip con nombre **`FAAP2_<grupo>_<pareja>.zip`** (ejemplo `FAAP2_1461_1.zip`) y el siguiente contenido:

1. **Ipython Notebook (.ipynb)** con las instrucciones necesarias para realizar las pruebas descritas en los apartados anteriores y el correspondiente análisis de resultados. El Notebook debe estructurarse para contener los siguientes apartados:

Apartado 1	Resultados de la clasificación mediante vecinos próximos (implementación original) para diferentes valores de vecindad y distancia en los conjuntos de datos propuestos. Obtener los resultados tanto para datos normalizados como sin normalizar, con el objetivo de justificar el rendimiento del algoritmo en base a estas características. Analizar el impacto de K y las diferentes distancias permitidas.
Apartado 2	Resultados de la clasificación mediante regresión logística en los conjuntos de datos propuestos. Probar con diferentes valores para la constante de aprendizaje y el número de pasos.
Apartado 3	Resultados de la clasificación utilizando los algoritmos de Scikit-Learn para vecinos próximos y regresión logística. Comparación con los resultados de la implementación propia. En los resultados utilizar los mismos valores de K y tipos de distancia que en la implementación propia.
Apartado 4	Aplicar el análisis ROC para comparar los clasificadores Naive Bayes, Vecinos Próximos y Regresión Logística en los conjuntos indians y wdbc. Comentar los resultados

2. **Ipython Notebook exportado como html.**

3. Código Python (**ficheros .py**) necesario para la correcta ejecución del Notebook

6. Puntuación de cada apartado

La valoración de cada apartado será la siguiente:

- ✓ **Vecinos próximos**: 3,5 puntos
- ✓ **Regresión logística**: 3,5 puntos
- ✓ **Scikit-Learn**: 1,5 puntos
- ✓ **Análisis ROC**: 1,5 puntos