PEC2: Reducción de la dimensionalidad y Clasificación

Presentación

En esta prueba de evaluación estudiaremos cómo aplicar técnicas de reducción de la dimensionalidad a datos de un estudio médico sobre patologías dermatológicas.

Competencias

En este enunciado se trabajan las siguientes competencias generales de máster:

- Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería informática.
- Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la ingeniería informática.
- Capacidad para aplicar los conocimientos adquiridos y solucionar problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos.
- Poseer habilidades para el aprendizaje continuo, autodirigido y autónomo.
- Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.

Las competencias específicas de esta asignatura que se trabajan en esta prueba son:

- Entender qué es el aprendizaje automático en el contexto de la inteligencia artificial
- Distinguir entre los diferentes tipos y métodos de aprendizaje
- Aplicar las técnicas estudiadas a un caso real

Objetivos

El objetivo de esta prueba de evaluación es la aplicación de técnicas de reducción de la dimensionalidad y de clasificación en datos reales.

Descripción de la PEC

En esta PEC aplicarán técnicas de reducción de la dimensionalidad y de clasificación en datos reales de señales de radar reflejadas en la ionosfera y recopiladas por Space Physics Group en el Applied Physics Laboratory de Johns Hopkins University.

Concretamente, nos basaremos en los datos que os facilitamos, fruto de este estudio publicado:

Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, (1989): 262-266.

El fichero de datos “ionosphere.data” incluye señales de radar reflejadas en la ionosfera. Estas señales deben ser clasificadas para decidir si tienen que ser objeto de un análisis posterior o si tienen que ser descartadas (ausencia de estructura identificable, interferencia,…). Por lo tanto, se trata de un problema binario. Cada línea corresponde a una señal y contiene 34 atributos i su clase. La clase es el último valor de cada muestra y se representa como ‘g’ (*good*) y ‘b’ (*bad*), señal aceptable y rechazado, respectivamente.

La descripción completa de este conjunto de datos la podéis encontrar en:

https://archive.ics.uci.edu/ml/datasets/Ionosphere

El objetivo de esta prueba es familiarizaros y hacer uso de diferentes métodos de reducción de la dimensionalidad y llevar a cabo una comparativa entre diferentes técnicas de clasificación automática.

Para ello, os deberéis basar en la librería *open source* **scikit-learn** de Python, que incorpora una gran variedad de algoritmos de aprendizaje automático, preprocesamiento, validación y visualización. Por tanto, es muy importante que contéis con la documentación oficial disponible en **scikit-learn.org.**

Requisitos:

- Matplotlib

https://matplotlib.org/users/installing.html

```
python –mpip install –U pip
```

python –mpip install –U matplotlib

- Scikit-Learn

http://scikit-learn.org/stable/install.html

```
pip install -U scikit-learn
```

- Scipy

https://scipy.org/install.html

```
pip install -U scipy
```

Ejercicio 1

Aplicad un análisis PCA a los datos de la PEC (recordad que los datos incluyen valores no válidos):

a) ¿Cuántos componentes principales son necesarios para representar el 95% de la varianza de los datos originales?

b) Reconstruid el conjunto de datos a partir de los 18 componentes principales (mediante el método *inverse_transform*), y calculad la pérdida de información respecto al conjunto original. Para hacerlo, podéis calcular el promedio de las diferencias elevadas al cuadrado entre cada elemento del conjunto reconstruido y del original. ¿Qué relación tiene este valor respecto a las varianzas acumuladas calculadas en el apartado anterior?

c) Visualizar los datos originales (2 primeros atributos) y los datos transformados según sus dos componentes principales utilizando un color diferente para cada clase. Comentad el resultado.

Ejercicio 2

Aplicad el método “multidimensional scaling” (MDS) a los datos. Repetid este proceso 3 veces y mostrad una gráfica en dos dimensiones con colores diferentes para cada clase. Comentad los resultados. ¿Por qué cada gráfica es diferente?

Ejercicio 3

Con los clasificadores de scikit-learn i los parámetros indicados a continuación, obtener el *score*, el *training time* i el *prediction time* cuando se aplican a los datos proporcionados utilizando K fold cross-validation (K=5):

- k Nearest Neighbors (módulo KneighborsClassifier de sklearn.neighbors): con 3, 4 y 5 vecinos (primer parámetro).
- Linear SVM (módulo SVC de sklearn.svm): *kernel="linear"*, *C=0.025*, el resto de parámetros, valor por defecto.
- Decision Tree (módulo DecisionTreeClassifier de sklearn.tree): *max_depth=5*, el resto de parámetros, valor por defecto.
- AdaBoost (módulo AdaBoostClassifier de sklearn.ensemble): parámetros por defecto.
- Gaussian Naive Bayes (módulo GaussianNB de sklearn.naive_bayes): parámetros por defecto.

Ejercicio 4

a) Repetid el ejercicio anterior (*score*, *training time*, *prediction time, cross-validation K=5*) pero, en este caso, comparando diferentes configuraciones del clasificador SVM (módulo SVC de sklearn.svm):

   - SVM: *kernel="linear"*, *C=0.025* el resto de parámetros, valor por defecto.
   - SVM: *kernel="linear"*, *C=100*, el resto de parámetros, valor por defecto.
   - SVM: *kernel="rbf"*, *C=0.025* el resto de parámetros, valor por defecto.
   - SVM: *kernel="rbf"*, *C=100* el resto de parámetros, valor por defecto.

b) Comentad el significado del parámetro 'kernel'. ¿Qué otras opciones predeterminadas proporciona Scikit-Learn?

c) Comentad el significado del parámetro C. ¿Qué valores puede tomar?

Recordad consultar la documentación de scikit-learn en su web (www.scikit-learn.org) con tal de descubrir los métodos y funcionalidades ya implementados que os pueden facilitar el desarrollo de vuestras soluciones.

Recursos

Esta PEC requiere los recursos siguientes:

Básicos:

Para realizar esta PEC disponéis del fichero adjunto:

- *ionosphere.data*
- *ionosphere.names*
- *pac2_2020_template.py*

Complementarios: manual de teoría de la asignatura, vídeos de la asignatura, web de scikit-learn.

Criterios de valoración

Los ejercicios tendrán asociada la valoración siguiente:

Ejercicio 1: 2.5 puntos

Ejercicio 2: 2.5 puntos

Ejercicio 3: 2.5 puntos

Ejercicio 4: 2.5 puntos

Todo el código, tanto para la lectura y preprocesado de los datos como para la solución de los ejercicios, tiene que estar implementado en la plantilla proporcionada y mostrar claramente las respuestas solicitadas sin interacción del usuario.

Se han de razonar las respuestas de todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.

Formato y fecha de entrega

La PEC se ha de entregar antes del **próximo 28 de abril** (incluido).

La solución tiene que consistir en un archivo zip que contenga un informe en formato PDF y el archivo pac2_2020_template.py donde se debe implementar la solución a los ejercicios.

Adjuntad el fichero en el apartado de **Entrega y registro de EC (REC)**. El nombre del fichero debe ser ApellidosNombre_IAA_PEC2.zip.

Para dudas y aclaraciones sobre el enunciado, diríjase al consultor responsable del aula.

Nota: **Propiedad intelectual**

A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por tanto comprensible hacerlo en el marco de una práctica de los estudios del Máster de Ingeniería Informática, siempre que esto se documente claramente y no suponga plagio en la práctica.

Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se presentará junto con ella un documento en el que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y el su estatus legal: si la obra está protegida por copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante deberá asegurarse de que la licencia que sea no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente deberá asumir que la obra está protegida por copyright.

Deberán, además, adjuntar los archivos originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.