PAC 3: Clasificación

Presentación

En esta prueba de evaluación estudiaremos cómo hacer un clasificador para detectar la quiebra de proyectos empresariales.

Competencias

En este enunciado se trabajan en un determinado grado las siguientes competencias general de máster:

- Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería en 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 en informática.
- Capacidad para la aplicación de los conocimientos adquiridos y para 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 continuado, autodirigido y autónomo.
- Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- Capacidad para asegurar, gestionar, auditar y certificar la calidad de los desarrollos, procesos, sistemas, servicios, aplicaciones y productos informáticos.

Las competencias específicas de esta asignatura que es trabajan son:

- Entender que es el aprendizaje automático en el contexto de la Inteligencia Artificial.
- Distinguir entre los diferentes tipos y métodos de aprendizaje.
- Aplicar las técnicas estudiadas en un caso concreto.

Objetivos

El objetivo de esta prueba de evaluación es el estudio para la implementación en python de un detector de fallos en proyectos empresariales.

Descripción de la PEC a realizar

El objetivo de esta prueba de evaluación es clasificar los datos de los archivos adjuntos relacionados con la quiebra de proyectos empresariales. Queremos predecir la quiebra en función de sus propiedades.

Datos

El archivo de datos Qualitative_Bankruptcy.data.txt tiene un formato tipo tabla, donde cada fila corresponde a uno de los 175 ejemplos. La última columna es la clase y el resto corresponden a los atributos del ejemplo. El archivo adjunto Qualitative_Bankruptcy.info.txt contiene la descripción de estos atributos.

Estos archivos pertenecen al problema "Qualitative Bankruptcy" del repositorio de aprendizaje de la UCI :

http://archive.ics.uci.edu/ml/

Ejercicio 1

Implementar un programa python que aplique el Naïve Bayes, árboles de decisión y la Adaboost en el archivo de datos Qualitative_Bankruptcy.data.txt, utilizando la validación cruzada como protocolo de validación y la *accuracy* como medida de evaluación.

Solución

Hemos implementado la solución utilizada 4 archivos: classification.py, naiveBayes.py, decisionTrees.py y adaboost.py. El primero hace el tratamiento general de carga de los datos, tratamiento del protocolo de validación, creación de los conjuntos de entrenamiento y test, la llamada a los algoritmos de aprendizaje y muestra los resultados devueltos por los mismos (en forma de promedio y desviación estándar) .

Ejercicio 2

Lea la página 844 del capítulo:

H.B. Aradhye & C. Dorai . Multimodal Analysis in Multimedia Using Symbolic Kernels

del libro "Encyclopedia of Data Warehousing and Mining", editado por John Wang. Puede consultar la página en la dirección:

http://books.google.es/books?id=gdkY4QHy0XIC&pg=PA844&lpg=PA844

Implemente el kernel que considere apropiado para los datos de nuestro conjunto como precomputed kernel de las máquinas de vectores de soporte del sklearn y aplicarlo a el archivo de datos Qualitative_Bankruptcy.data.txt. Utilice la validación cruzada como protocolo de validación y la *accuracy* como medida de evaluación.

Solución

Después de leer el artículo adjunto, se pueden seleccionar cualquiera de los kernels 2 Nominal, 3 Discrete o 4 Stringlike. Nosotros hemos decidido implementar el número 3, debido a que los atributos del conjunto de datos son ordinales. Cualquiera de los otros también nos valdría.

La implementación del kernel queda así:

```
def valor(x):
 return {'N': 0, 'A': 1, 'P': 2}[x]

def kfCityBlock(a, b):
 return 2 * 6 - sum(map(lambda x, y:
            abs(valor(x) - valor(y)), a, b))
```

El uso de las SVMs con el sklearn con el kernel como precomputed matrix lo tiene en el archivo classification.py.

Ejercicio 3

Elabore una tabla con todos los resultados de los ejercicios anteriores. Realice una valoración global comparando los métodos (mirando si las diferencias son estadísticamente significativas) y redacte unas conclusiones globales sobre la aplicación de los métodos a este conjunto de datos. Los criterios de corrección de la PAC invalidan una A si todos los procesos no están bien justificados y comentados.

Solución

El fragmento de código siguiente muestra el funcionamiento de los componentes mencionados en el enunciado:

```
5-fold cross-validation
```

```
Naive Bayes
 Accuracy: 96.6 +- 2.1 %
Decision Trees
 Accuracy: 90.3 +- 4.6 %
AdaBoost
 Accuracy: 95.4 +- 3.4 %
Support Vector Machines (CityBlock)
 Accuracy: 94.3 +- 2.6 %
```

En este ejercicio se le pide que valore los resultados globales y dar unas conclusiones globales de su aplicación. ¿Qué habla de cuestiones como la precisión obtenida, coste computacional ... y que comente su utilidad práctica y

la de cada uno de los componentes que haya importado de la categorización de textos cuando los aplicamos a este problema.

Para terminar, cómo cree que puede mejorar el sistema? ¿Qué líneas futuras probaríais?

Recursos

Esta práctica requiere los siguientes recursos:

Básicos:

Para realizar esta PAC dispone de unos archivos adjuntos:

- Qualitative_Bankruptcy.data.txt
- Qualitative_Bankruptcy.info.txt
- Archivo del vídeo Clasificación + Naïve Bayes
- Código Árboles de decisión
- Código Adaboost
- Código del vídeo sklearn + precomputed
- Artículo

Complementarios: Manual de teoría de la asignatura

Criterios de valoración

Los ejercicios tendrán la siguiente valoración asociada:

Ejercicio 1: 3,5 puntos

Ejercicio 2: 3,5 puntos

Ejercicio 3: 3 puntos

Es necesario razonar las respuestas en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.

Formato y fecha de entrega

La PEC debe entregarse antes del **próximo 15 de mayo** (antes de las 24h).

La solución a entregar consiste en un informe en formato PDF usando la plantilla colgada en el tablón de la asignatura más los archivos de código (*. Py) que usó para resolver la prueba. Estos archivos deben comprimir en un archivo ZIP.

Adjuntar el fichero a un mensaje en el apartado de **Entrega y Registro de AC (RAC)**. El nombre del archivo debe ser ApellidosNombre_IA_PEC3 con la extensión. zip.

Para dudas y aclaraciones sobre el enunciado, diríjase al consultor responsable de su 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 Grado en Informática, siempre y 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.