UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

Ensembles de clasificadores multi-label en Scikit-Learn

Presentado por Eduardo Tubilleja Calvo
en Universidad de Burgos — 24 de enero de 2018
Tutor: Dr. Álvar Arnaiz González
y Dr. Juan José Rodríguez Díez

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Álvar Arnaiz González y D. Juan José Rodríguez Díez, profesores del departamento de nombre departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Eduardo Tubilleja Calvo, con DNI 71298897R, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «Ensembles de clasificadores multi-label en Scikit-Learn».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 24 de enero de 2018

V^{o}. B^{o}. del Tutor:

D. Álvar Arnaiz González

V^{o}. B^{o}. del Tutor:

D. Juan José Rodríguez Díez

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

La minería de datos es un campo de la estadística y las ciencias de la computación, que consisten en el análisis de grandes cantidades de datos para descubrir patrones. Para ello se utiliza el aprendizaje automático, éste pertenece a un subcampo de las ciencias de computación y de la rama de inteligencia artificial, el objetivo de éste es desarrollar unas técnicas que permitan que las máquinas aprendan [20]. Dentro de este aprendizaje se encuentra el aprendizaje supervisado, en él normalmente los conjuntos de datos suelen tener solo una variable a predecir, conocido como single-label, pero aparecido el Multi-Label, este hace referencia a los conjuntos de datos en lo que cada elemento de la base de datos puede pertenecer a más de una clase, como por ejemplo en el etiquetado de imágenes: en el que una imagen puede tener a la vez las etiquetas «árbol», «montaña» y «mar». En este proyecto vamos a tratar sobre implementar diversos algoritmos de clasificadores (ensembles), para Multi-Label sobre la librería Scikit Learn de Python. Se ha seguido la guía de estilo de Python (PeP [4]) y Sklearn. Para que se entienda mejor y sea más gráfico, se han dibujado árboles y gráficas, mostrando los resultados al ejecutar dichos algoritmos sobre un conjunto de datos. Los algoritmos en los que nos vamos a centrar son Disturbing Neighbors [8], Random Oracles [13] y Rotation Forest [14].

Vamos a tratar a lo largo del documento de los ensembles, para que estos sean precisos necesitan que los clasificadores base predigan correctamente la clase de las mismas instancias. Tienen que ser diferentes para complementarse entre ellos, por ello la diversidad es importante. ¿Cómo puede un ensemble de clasificadores base que han sido generados por el mismo algoritmo tener distintas salidas? Una de las estrategias que podemos usar para ello son los ensembles homogéneos, es decir, mismo algoritmo entrenador con distinto conjunto de datos.

Objetivos del proyecto

En este apartado, se explica los objetivos que se quieren conseguir al final, la meta que pretendemos lograr y los motivos que me han llevado a realizar este proyecto.

2.1. Objetivos

A continuación se muestra el esquema con todos los puntos a tratar en este proyecto.

- Implementar los algoritmos Disturbing Neighbors [8], Random Oracles [13] y Rotation Forest [14] en Scikit-learn:
  - Que sirva para datos Single-Label o Multi-Label
  - Crear el método `fit` para entrenar un conjunto de datos.
  - Crear el método `predict` para predecir según el entrenamiento de unos datos.
  - Crear el método `predict_proba` para predecir probabilidades según el entrenamiento de unos datos.
  - o Evaluar el correcto funcionamiento de la clases.
- Crear diversos notebooks para mostrar el funcionamiento de los algoritmos:
  - Posibilidad de seleccionar el algoritmo.
  - Permitir utilizar conjuntos reales.
  - Mostrar su funcionamiento mediante el dibujado del árbol de decisión.
  - Mostrar el funcionamiento mediante gráficas en dos dimensiones con conjuntos de datos de «juguete».

- Mostrar resultados al usar validación cruzada.

Conceptos teóricos

Este apartado explica los conceptos teóricos necesarios para poder entender el proyecto.

3.1. Minería de Datos

Es conocida la frase «los datos en bruto raramente son beneficiosos directamente». Puede tener valor, ya que podemos extraer información útil para la toma de decisiones o exploración, y también para la compresión del fenómeno dominante en el conjunto de datos

> mineria.
>
> La finalidad de esto es descubrir unos patrones, una similitud o una propensión que expliquen el comportamiento de los datos. Para hacer esto utiliza los métodos de la inteligencia artificial, estadística y redes neuronales. El objetivo del proceso de minería de datos consiste en extraer información de un conjunto de datos, luego se interpreta esta información para un uso posterior [20].
>
> La minería de datos está basada en el aprendizaje automático, para ello se considera un conjunto con n muestras y se intenta predecir las propiedades de los datos desconocidos. Podemos separar los problemas de aprendizaje principalmente en dos [3]:
>
> - Aprendizaje supervisado: Consiste en que a partir de un conjunto de datos, hacer predicciones basadas en el comportamiento o las características de dichos datos. Nos permite buscar patrones en datos históricos. Dos de las tareas más comunes del aprendizaje supervisado son la clasificación y la regresión:

  - Clasificación: El programa debe aprender a predecir en que categoría o clase irán los nuevos datos, según las nuevas observaciones, por ejemplo, predecir si el precio de una acción bajará o subirá.
  - Regresión: El programa debe predecir el valor de una variable de respuesta continua, por ejemplo, predecir las ventas de un nuevo producto.
- Aprendizaje no supervisado: Usa datos históricos que no están etiquetados. El objetivo es explorarlos para encontrar alguna forma de organizarlos.

La minería de datos es conocida como sinónimo de KDD (Knowledge Discovery in Databases, Descubrimiento del conocimiento en bases de datos) Este proceso tiene una secuencia de pasos [5]:

1. Limpieza de datos. Eliminación del ruido y la inconsistencia de los datos.
2. Integración de los datos. Múltiples fuentes de datos son combinadas.
3. Selección de datos. Los datos relevantes para la tarea de análisis se recuperan de la base de datos.
4. Transformación de datos. Los datos son transformados y se consolidad como apropiados para la minería mediante la realización de operaciones simétricas o de agregación.
5. Minería de datos. Es un proceso esencial donde los métodos de inteligencia son aplicados a la extracción de patrones.
6. Evaluación de patrones. Identificar los patrones que de verdad son interesantes, que representan el conocimiento basado en medidas de interés.
7. Presentación del conocimiento. La visualización y el conocimiento representan técnicas que son utilizadas para presentar la minería del conocimiento a los usuarios.

Aunque el proceso de la minería de datos consta de más etapas nosotros nos centraremos en 4 etapas [16]:

- Determinación de los objetivos: Se tratan los objetivos que quiere conseguir el cliente bajo un asesor especialista en minería de datos.
- Preprocesamiento de los datos: Es la etapa que más tiempo se tarda en realizar el proceso. Se seleccionan, limpian, enriquecen, reducen y transforman las bases de datos.

- Determinación del modelo: Se lleva a cabo un estudio estadístico de los datos, más tarde se hace una visualización gráfica para una primera aproximación. Según los objetivos que se habían propuesto se pueden usar diferentes algoritmos de la Inteligencia Artificial.
- Análisis de los resultados: Se comprueban si los datos obtenidos tienen coherencia, después se comparan con los obtenidos en los estudios estadísticos y la visualización gráfica. El cliente es el que ve si los datos le aportan nuevo conocimiento que le permita considerar sus decisiones.

3.2. Multi-Label

Una investigación en aprendizaje supervisado trata con el análisis de datos de Single-Label donde los ejemplos de entrenamiento son asociados con un Single-Label de un conjunto de disjuntos labels. Sin embargo, los ejemplos de entrenamiento en varios dominios de un aplicación, son asociado un conjunto de labels, a los que llamaremos Multi-Label [17].

La clasificación Multi-Label es una técnica de minería de datos, la cual nos permite que de un conjunto de instancias de entrenamiento, podamos determinar a partir de unos atributos esenciales de dichas instancias para crear unas reglas que posteriormente se usarán para clasificar nuevas instancias [11]. Como por ejemplo en el etiquetado de imágenes: en el que una imagen puede tener a la vez las etiquetas «árbol», «montaña» y «mar».

3.3. Ensemble

Un ensemble es un esquema de combinación de predicciones individuales llamados clasificadores base. El éxito de un ensemble requiere tanto exactitud como diversidad de sus clasificadores base. La diversidad representa como diferente son las predicciones de los clasificadores base. Si los clasificadores base siempre están de acuerdo podría no haber diferencia entre usar sólo un clasificador base o varios combinado por un método de ensemble. Entonces el poder de usar un conjunto de clasificadores base consiste en la posibilidad que algunos de ellos pueden corregir una predicción incorrecta de otros.

Es normal obtener estos clasificadores base en un ensemble usando el mismo algoritmo, así que en esta situación el proceso de entrenamiento realizado por el ensemble es la principal fuente de

diversidad. La diversidad de Bagging proviene de elegir al azar diferentes instancias para entrenar a cada clasificador de base. El método Random Subspaces elige diferentes subconjuntos de atributos para entrenar a cada clasificador base. Boosting entrena de forma iterativa el conjunto de clasificadores base, modificando los pesos de las instancias para entrenar al clasificador actual. Estos nuevos pesos se calculan a partir del error de entrenamiento en el clasificador base anterior, por lo que cada nuevo clasificador de base llega más especializado en instancias que han sido mal clasificados antes. A veces los clasificadores base son muy estables y el algoritmo de entrenamiento del ensemble no es suficiente para proporcionar el nivel deseado de diversidad [8].

Los métodos de ensembles combinan las predicciones de unos estimadores base, que están construidos mediante un algoritmo de aprendizaje para mejorar la solidez de un solo estimador [15]. Se distinguen dos clases de métodos de ensembles:

- En los métodos de Bagging [1], se construyen varios estimadores de forma independiente y luego se calcula su promedio para las predicciones.
- En los métodos de Boosting [2], se construyen los estimadores secuencialmente y se trata de reducir el sesgo del estimador combinado.

3.4. Disturbing Neighbors

Introducción

Los Disturbing Neighbors (DN) o vecinos molestones se han utilizado con éxito para mejorar la diversidad en los bosques. DN usa un clasificador de 1-Nearest Neighbour (1-NN) para construir un conjunto de características adicionales que se agregan al conjunto de datos de entrenamiento de cada clasificador base. Este clasificador 1-NN es diferente para cada clasificador base. Las características compiladas son la predicción 1-NN más un conjunto booleano de características que indican cuál es el vecino más cercano. El conjunto de datos de entrenamiento original se transforma en un conjunto de datos aumentados, que es diferente para cada clasificador base, independientemente del esquema de conjunto en el que se va a utilizar.

Método

El método DN trabaja en cada clasificador base de la siguiente manera:

1. m instancias son seleccionadas aleatoriamente de el conjunto de datos de entrenamiento para construir un clasificador 1-NN. El valor m usa valores muy pequeños.
2. Dimensiones usadas para calcular distancia euclidea en el clasificador 1-NN son también seleccionadas aleatoriamente. Al menos el 50 % de los atributos son seleccionados.
3. Luego m+1 nuevas características son añadidas al conjunto de entrenamiento. Una de las características adicionales es la clase predecida por el clasificador 1-NN para cada instancia x, y la otra m son características booleanas, todos los conjuntos falsos excepto uno corresponden al vecino más cercano para esa instancia.
4. El clasificador base está entrenado usando las características originales mas las nuevas características de m+1.

Por lo tanto, el proceso normal de entrenamiento de los clasificadores básicos se altera añadiendo estas nuevas características del clasificador 1-NN. Es por eso que el método se llama vecinos molestones. La aleatoriedad aumenta la diversidad y se debe a:

- Los vecinos utilizados en cada clasificador 1-NN se seleccionan aleatoriamente. Por lo tanto, sus predicciones y las características booleanas son diferentes para cada clasificador base.
- Las dimensiones utilizadas para calcular las distancias euclidianas también se eligen de forma aleatoria, así que si dos clasificadores básicos tienen al menos los mismos m vecinos, las predicciones 1-NN y las características booleanas podrían ser diferentes.

Conclusión

Disturbing Neighbors es un método para alterar el proceso de entrenamiento normal de los clasificadores base en un ensemble, mejorando su diversidad y mejorando la precisión general del ensemble. Disturbing Neighbors crea nuevas características utilizando un clasificador 1-NN. Estas características son la salida 1-NN más un conjunto de atributos booleanos que indican cuál es el vecino más cercano. El clasificador 1-NN se crea utilizando un pequeño subconjunto de instancias de entrenamiento seleccionadas al azar del

conjunto de datos original. Las dimensiones utilizadas para calcular la distancia euclidiana también se seleccionan de forma aleatoria. Estas dos fuentes de aleatoriedad son las razones por las que las características creadas son diferentes cada vez, por lo que cuando estas nuevas características se usan para entrenar clasificadores base, la diversidad aumenta.

3.5. Random Oracles

Introducción

Random Oracles son mini-ensembles formados por dos modelos, pueden usarse como modelos base para otros métodos ensemble. El objetivo de usar Random Oracles es tener más diversidad entre los modelos base que forman un ensemble. Esta diversidad adicional puede mejorar la precisión de los ensembles.

Método

Un modelo Random Oracle es un mini-ensemble formado por un par de modelos y un Oracle aleatorio que elige entre ellos. Se puede considerar como una función discriminante aleatoria que divide los datos en dos subconjuntos sin tener en cuenta ninguna etiqueta de clase. Además, se puede usar un Oracle aleatorio como el modelo base de cualquier método ensemble. Dado un método base, el entrenamiento de un Random Oracle consiste en:

- Seleccionar aleatoriamente un Random Oracle.
- Dividir los datos de entrenamiento en dos subconjuntos usando el Random Oracle.
- Para cada subconjunto de datos de entrenamiento, se construye un modelo. El modelo Random Oracle está formado por un par de modelos y el propio oráculo.

La predicción del test de una instancia se realiza de la siguiente manera:

- Usa el Random Oracle para seleccionar uno de los dos modelos.
- Devuelve la predicción obtenida por el modelo seleccionado.

Si la complejidad computacional del oráculo es baja, tanto en el entrenamiento como en la predicción, la complejidad computacional de un modelo de Oracle aleatorio es muy similar a la complejidad del método base. En la fase de predicción, solo se usa uno de los

dos modelos. En la fase de entrenamiento, se construyen dos modelos. Sin embargo, están entrenados con una partición disjunta de los ejemplos de entrenamiento y el tiempo de entrenamiento de cualquier método depende, al menos linealmente, del número de ejemplos de entrenamiento.

Conclusión

Se pueden considerar diferentes tipos de Oracles. En este trabajo, se utiliza el Linear Random Oracle. Este oráculo divide el espacio en dos subespacios utilizando un hiperplano. Para construir el oráculo, dos objetos de entrenamiento diferentes se seleccionan aleatoriamente, cada objeto de entrenamiento restante se asigna al subespacio del objeto de entrenamiento seleccionado que está más cerca.

Las distancias se calculan según la distancia euclidiana, los atributos numéricos se escalan dentro de [0,1], para los atributos nominales consideramos que la distancia es 0 o 1 dependiendo de si los dos valores son diferentes o iguales.

3.6. Referencias

Las referencias se incluyen en el texto usando cite [19]. Para citar webs, artículos o libros [**?**].

3.7. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.1: Autómata para una expresión vacía

3.8. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Segundo item más información sobre el segundo item.

-

3.9. Tablas

Igualmente se pueden usar los comandos específicos de LATEXo bien usar alguno de los comandos de la plantilla.

Técnicas y herramientas

En este apartado de la memoria se presentan las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto.

4.1. GitHub

Es una plataforma para alojar proyectos y utiliza git como sistema de control de versiones. Se organiza por tareas (milestones e issues). Utiliza el framework Ruby on Rails [9].

Podemos acceder a él a través del siguiente enlace: https://github.com/

Ventajas:

- Es uno de los repositorios mas usados, por lo que es fácil encontrar información en Internet para resolver cualquier duda.
- El código es público por lo que cualquiera puede proponer cambios en el mismo, seguirte y ver el proyecto.
- Las distintas versiones del código están alojadas en la nube por lo que si perdemos el contenido de nuestro ordenador, podremos recuperarlo.

Desventajas:

- También puedes tener proyectos privados pero para ello tienes que utilizar una cuenta de pago, aunque los estudiantes e investigadores pueden obtener esto gratuitamente.

4.2. Python

Python es un lenguaje de programación interpretado se hace hincapié en que una sintaxis que favorezca un código legible [18]. Se trata de un lenguaje de programación multiparadigma (permite crear programas utilizando mas de un estilo de programación), ya que soporta orientación a objetos, programación imperativa y programación funcional. Es un lenguaje interpretado, usa tipado dinámico(una variable puede tomar valores de distinto tipo) y es multiplataforma [22].

Python es recomendable para programadores que empiezan por primera vez, o que vienen de ostros lenguajes, ya que hay mucha documentación para dar el primer paso en este lenguaje. La comunidad organiza conferencias y también colabora con el código.

Contiene miles de módulos de terceros, a parte de la biblioteca estándar de Python, por lo que tenemos infinitas posibilidades.

Por último, Python se desarrolla bajo una licencia de código abierto aprobada por OSI, por lo que es se puede utilizar libremente y distribuir. La licencia de Python es administrada por la Python Software Foundation.

Podemos acceder a él a través del siguiente enlace: `https://www.python.org/`

4.3. Spyder

Es un entorno de desarrollo interactivo para el lenguaje de Python, es de código abierto. Tiene funciones avanzadas de edición, pruebas interactivas, depuración e introspección. También es un entorno informático numérico y tiene diversas bibliotecas que podemos utilizar, como pueden ser numpy [12].

Podemos acceder a él a través del siguiente enlace: `http://pythonhosted.org/spyder/`

4.4. LaTeX

Se usa para la creación de documentos que necesiten una alta calidad tipográfica, como puede ser en artículos o libros científicos [7].

Podemos acceder a él a través del siguiente enlace: `https://www.latex-project.org/`

Ventajas:

- Es software libre, por lo que no requiere ningún coste.

- No te tienes preocupar por el diseño, ya que la herramienta se encarga de ello.

Desventajas:

- Si eres principiante necesitas un tiempo de aprendizaje para saber como funciona.

4.5. Jupyter Notebook

Es una aplicación web de código abierto, con él podemos crear y compartir documentos, que nos permiten visualizar los resultados al ejecutar nuestro código, ya sean imágenes, árboles...que otros entornos de desarrollo (como Spyder mencionado anteriormente) no nos permiten esto. Soporta más lenguajes, pero nosotros lo usaremos para el lenguaje de Python [6].

Podemos acceder a él a través del siguiente enlace: `http://jupyter.org/`

4.6. Scikit-learn

Es una librería de Python que contiene algoritmos de aprendizaje automático para problemas supervisados y no supervisados. Como esta basado en Python, puede integrarse fácilmente en aplicaciones que no suelen usarse para análisis de datos estdísticos. EL trabajo futuro incluye aprendizaje en línea para escalar a grandes conjuntos de datos [10].

Podemos acceder a él a través del siguiente enlace: `http://scikit-learn.org/stable/`

4.7. SonarQube

Es una plataforma que sirve para evaluar y analizar código. Es software libre, para llevar a cabo este análisis del código utiliza distintas herramientas como pueden ser Checkstyle, PMD o FindBugs, con dichas herramientas obtenemos métricas que nos ayudan a mejorar la calidad de nuestro código fuente [21].

Los aspectos que evalúa esta herramienta son:

- Technical Debt, esta parte nos indica los aspectos y métricas que no habíamos tenido en cuenta y también nos muestran la claridad del código. Una de las ventajas es que te indica donde

has cometido una falta de estilo o donde tienes demasiada complejidad.

- La complejidad, los cambios de flujo que sufre el código, es decir, las condiciones if, while for...
- Podemos ver las líneas de código que hemos escrito en cada fichero (sin contar los comentarios).
- Si queremos evaluar más aspectos, podemos instalar Plugins que nos lo permiten.

Podemos acceder a él a través del siguiente enlace: https://www.sonarqube.org/

4.8. Graphviz

Es un software de visualización gráfica de código abierto. Es una forma de representar nuestra información estructural como diagramas de gráficos y redes abstractas. En nuestro caso lo utilizamos para dibujar un árbol de nuestro conjunto de datos entrenado. Su arquitectura consiste en un lenguaje de descripción de gráficos llamado DOT.

Podemos acceder a él a través del siguiente enlace: https://www.graphviz.org/

4.9. Zenhub

Es un gestor de tareas es similiar a Trello https://trello.com/, tiene un modo pizarra en el que podemos ver los cambios. Una de las grandes ventajas es que podemos integrarlo desde GitHub, por lo que no es necesario el uso de una aplicación externa. Una pequeña desventaja podríamos decir que es que no se puede añadir código, pero como el repositorio de GitHub nos permite visualizar dicho código no es un gran problema.

Podemos acceder a él a través del siguiente enlace: https://www.zenhub.com/

Aspectos relevantes del desarrollo del proyecto

5.1. Formación

El proyecto requería unos conocimientos técnicos de los que desconocía en un principio. Entre ellos, conocimientos sobre Minería de datos, Scikit-Learn y documentar en LaTeX. Para aprender estas cosas se ha necesitado leer documentos científicos, descubriendo la utilidad de ellos, ya que nunca había hecho uso de ellos.

Fue necesario instruirse en cómo implementar algoritmos, para ello tomamos de ejemplo, los algoritmos ya implementados en Scikit-Learn.

Otro de los conocimientos adquiridos en la realización de ese proyecto ha sido la librería Scikit-Leran, porque nosotros queremos implementar algoritmos en dicha librería, porque vamos a tratar la minería de datos, y como queremos clasificar un conjunto de datos para después de entrar poder predecir unos resultados con la mayor precisión posible, Sickit-Learn es una librería perfecta para esto.

5.2. Entorno de desarrollo

Como entorno de desarrollo, hemos utilizado Spyder, para la programación de Python, y se ha utilizado Jupyter, ya que en sus notebooks interactivos puedes ejecutar código de Python como si fuera un interprete.

Ventajas:

- Spyder es un editor multilingüe que tiene características como coloreado de sintaxis, análisis de código, navegador de funciones...
- Tiene su propia consola con la que podemos interactuar y visualizar los datos. Como los comandos ingresados en la consola se ejecutan en un proceso separado, podemos detener cualquier proceso cuando queramos.
- También podemos depurar el código, para encontrar cualquier error.

Trabajos relacionados

En este apartado vamos hablar y comparar nuestro proyecto con otros relacionados. Lo dividiremos en 3 partes, una en la que hablaremos de otras librerías ensembles, otra parte sobre librerías de aprendizaje automático y una última de servicios de computación en la nube.

6.1. Librerías ensemble

A parte de de la librería que hemos usado de Scikit-Learn, existen librerías similares, hablaremos sobre alguna de ellas y las compararemos con Scikit-Learn.

TensorFlow

Es una librería de computación numérica que compute gradientes automáticamente. Es un sistema flexible y se puede utilizar para gran variedad de algoritmos, incluidos de entrenamiento e inferencia para modelos de redes neuronales. Se ha utilizado para realizar investigaciones y para implementar sistemas de aprendizaje automático en diferentes áreas [**?**].

Ha sido desarrollada por Google, y la utilizan empresas como Dropbox, Uber y Snapchat [**?**].

Pytorch

Es una biblioteca de aprendizaje de máquina de código abierto para Python que permite un crecimiento rápido de Deep Learning. Su mayor característica es que utiliza grafos computacionales dinámicos.

Ha sido desarrollado principalmente por el grupo de investigación de inteligencia artificial de Facebook, y por ejemplo el software "Pyro"de Uber para la programación probabilística se basa en él [**?**].

Keras

Es una API de redes neuronales de alto nivel, esta escrita en Python. Fue desarrollado con un enfoque en permitir la experimentación rápida.

EnsembleSVM

Es un aprendizaje automático de software libre. Esta librería nos ofrece la funcionalidad para llevar a cabo el aprendizaje de ensembles utilizando modelos base de la máquina de vectores de soporte(SVM). Permite entrenar eficientemente modelos para grandes conjuntos de datos [**?**].

6.2. Librerías de aprendizaje automático

Weka, Meka, Keel y Matlab

6.3. Servicios de computación en la nube

Google data proc, Azure, AWS

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[2] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[3] Andrés González. Conceptos básicos de machine learning, 2015. [Online].

[4] Nick Coghlan Guido van Rossum, Barry Warsaw. Style guide for python code, 2013. [Internet].

[5] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[6] jupyter. Jupyter-notebook, 2017. [Online].

[7] latex project. Latex, 2017. [Online].

[8] Jesús Maudes, Juan José Rodríguez, and César Ignacio García-Osorio. Disturbing neighbors ensembles for linear svm. In *MCS*, pages 191–200. Springer, 2009.

[9] Carlos Paramio. Github, 2011. [Online].

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] Julio Antonio Hernández Pérez, Raudel Hernández León, and Tercer C Autor. Clasificación multi-etiquetas basada en reglas de asociación de clases.

[12] pythonhosted. Spyder, 2017. [Online].

[13] Juan Rodríguez and Ludmila Kuncheva. Naïve bayes ensembles with a random oracle. *Multiple Classifier Systems*, pages 450–458, 2007.

[14] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.

[15] scikit learn. Ensemble, 2017. [Online].

[16] sinnexus. Minería de datos, 2016. [Online].

[17] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[18] Guido Van Rossum and Fred L Drake. *Python language reference manual*. Network Theory, 2003.

[19] Wikipedia. Latex — wikipedia, La enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].

[20] Wikipedia. Minería de datos — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 3-enero-2018].

[21] Wikipedia. Sonarqube — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 23-enero-2018].

[22] Wikipedia. Python — wikipedia, la enciclopedia libre, 2018. [https://es.wikipedia.org/w/index.php?title=Pythonoldid=105012532].