|  |
| --- |
|  |
| **Introducción a los Métodos de Aprendizaje Automático** |
| **Ejercicios domiciliarios**  **UT1 – PD1** |
| **Autor:** Gerardo Fernández - CI: 2858230-7  **Docente:** Ernesto Ocampo |
| **27/08/2021** |
|  |

**Ejercicio 1**

*Def 1:*

Machine Learning es un método de análisis de datos que automatiza la construcción de un modelo analítico. Es una rama de la inteligencia artificial, basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana.

*Def 2:*

Es el estudio de algoritmos que pueden mejorar automáticamente a través de la experiencia y del uso de los datos. Los algoritmos construyen un modelo en base a datos de muestra, conocidos como training data, para hacer predicciones o tomar decisiones sin tener que ser programados específicamente para eso.

*Def 3:*

Es un área de conocimiento dentro de la Inteligencia Artificial donde las computadoras aplican técnicas de aprendizaje estadístico con el objetivo de identificar automáticamente patrones en los [datos](https://www.iebschool.com/blog/analista-datos-big-data/).

La IA se enfoca en crear máquinas inteligentes que pueden imitar el razonamiento, capacidad y conducta humana, mientras que el machine learning es una rama de la IA que permite a las máquinas aprender de los datos sin ser programadas explicitamente para hacerlo.

La diferencia entre ML y análisis estadístico es su propósito. ML está pensado para realizar las predicciones más acertadas posibles, mientras que la estadística busca relaciones entre las variables y el significado de esas variables.

Data Mining también busca patrones en los datos, pero el ML no solo se centra en lo que sucedió o lo que ocurre sino que predice lo que sucederá en base a información similar.

Además el DM requiere mayot intervención humana para la toma de decisiones, mientras que en ML el proceso de aprender y mejorar su performance es automático.

Las áreas de aplicación son muy diversas. Hay sectores donde puede resultar clave para la toma de decisiones como puede ser en medicina, donde decidir si realizar una operación o no a un paciente o en el mundo de los negocios conocer en qué fechas es mejor bajar los precios o estimar ventas.

Otras aplicaciones son:

* Motores de búsqueda,
* Reconocimiento del habla, del lenguaje:
* Detección de rostro. Lo vemos en nuestros móviles.
* Anti-spam. Mediante el uso de tags.
* Anti-virus. Detectando software malicioso.
* Genética. En la clasificación de secuencias de ADN.
* Forecast. De clima o tráfico.
* Comprensión de textos.
* Vehículos autónomos y robots.
* Análisis de imágenes de alta calidad.
* Análisis de comportamiento de consumo y productividad. Para la identificación de clientes potenciales y en general en sectores de telco, banca, seguros…

**Ejercicio 2**

**Machine learning platforms**

*AI Platform y Datasets en Google Cloud*

<https://cloud.google.com/ai-platform>

The fundamental problem of any ML model is that you need a correct dataset to train it. They are expensive to make and take lots of time. [Google Cloud Public Datasets](https://cloud.google.com/public-datasets) are datasets curated by Google that are regularly updated. The formats are very different: from images to audio, video, and texts. The data is intended for a wide range of researchers with different use cases.

In addition, Google offers other [useful services](https://cloud.google.com/ai-platform) that you could find interesting:

* AI platform for training and managing ML models;
* Natural language processing services;
* Vision AI (models for computer vision);
* Speech synthesis software in more than 30 languages etc.

*Amazon Web Services*

<https://aws.amazon.com/machine-learning/>

[AWS](https://aws.amazon.com/machine-learning/) is a platform that provides artificial intelligence and machine learning services to developers. It is possible to choose one of the pre-trained AI services to work with computer vision, language recognition, speech generation, build recommender system and prediction models.

Using [Amazon SageMaker](https://aws.amazon.com/sagemaker/), you can quickly create, train, and deploy scalable machine learning models, or create custom models that support all the popular open-source ML platforms.You can also use Amazon’s services to provide new functionality to existing business solutions. They can be easily integrated with different software, for example, to modernize the contact center and increase customer retention. AWS can help achieve higher customer satisfaction and expand the standard set of business tools

*Microsoft Azure*

<https://azure.microsoft.com/>

[Azure Machine Learning Studio](https://azure.microsoft.com/) allows developers who don’t have experience in machine learning to use the drag-and-drop functionality. This platform allows you to build solutions directly “on the cloud” and easily create BI applications regardless of the quality of the data.

Microsoft also offers Cortana Intelligence, a tool that allows you to fully manage big data and analytics and transform data into meaningful information and subsequent actions.

Overall, Azure can be used by teams and large organizations to work on ML solutions together in the cloud. It has a wide set of tools for different purposes, which makes it so loved by international corporations.

*RapidMiner*

<https://rapidminer.com/>

[RapidMiner](https://rapidminer.com/) is a platform for data science and machine learning. It has a convenient graphical interface and allows to process data from a variety of different formats, including .csv, .txt, .xls, .pdf. Due to this ease of use and respect for privacy, Rapid Miner is used by thousands of enterprises around the world.

This tool is good when you need to build automated models quickly. It will help you to automatically analyze data and identify common quality problems with correlations, missing values, and stability. However, in order to solve more complex research problems, it is better to use other tools.

Features:

* Through GUI, it helps in designing and implementing analytical workflows.
* It helps with data preparation.
* Result Visualization.
* Model validation and optimization.

Pros:

* Extensible through plugins.
* Easy to use.
* No programming skills are required.

Cons:

* The tool is costly.

*IBM Watson*

<https://www.ibm.com/watson>

If you’re looking for a fully-functional platform with a number of tools for both research teams and enterprises, check out the [Watson platform by IBM](https://www.ibm.com/watson).

Watson is an open-source API suite. Its users have access to sample codes, a starter toolkit, and can create cognitive search engines and virtual agents. Their tools can be used by any developer to create their own software in the cloud, and the prices are very customer-friendly, which makes it a good solution for small and medium-sized businesses.

In addition, Watson has a chatbot creation platform that can be used by machine learning beginners for faster bot training.

*Anaconda*

<https://www.anaconda.com/>

## [Anaconda](https://www.anaconda.com/) is an open-source ML platform for data analytics that works with Python and R. It can run on any supported operating systems for other platforms. It allows developers to use more than 1,500 Python and R data science packages, manage libraries and environments (including Dask, NumPy, and pandas).

## Anaconda has great visualization capabilities for reports and modeling. This tool is popular because it brings together many tools with just one install.

## Now, let us have a closer look at frameworks, libraries, and other tools for machine learning that you cannot miss out on.

**Machine learning frameworks and tools**

### *Python tools*

*TensorFlow*

<https://www.tensorflow.org/>

[TensorFlow](https://www.tensorflow.org/) is a set of open-source deep learning software libraries by Google. Using TensorFlow tools, ML specialists can create highly accurate and feature-rich machine learning models.

This software simplifies the process of building and deploying complex neural networks. TensorFlow offers APIs for Python and C/C ++ languages ​​that allow exploring its possibilities for research purposes. Moreover, enterprises all around the world get powerful tools for working with their own data and processing it in a cheap cloud environment.

TensorFlow libraries significantly simplify the integration of self-learning elements for applications designed to solve high complexity problems like speech recognition, computer vision, or natural language processing.

Features:

* Helps in training and building your models.
* You can run your existing models with the help of TensorFlow.js which is a model converter.
* It helps in the neural network.

Pros:

* You can use it in two ways, i.e. by script tags or by installing through NPM.
* It can even help for human pose estimation.

Cons:

* It is difficult to learn.

*Weka*

<https://www.cs.waikato.ac.nz/ml/weka/>

These machine learning algorithms help in data mining.

Features:

* Data preparation
* Classification
* Regression
* Clustering
* Visualization and
* Association rules mining.

Pros:

* Provides online courses for training.
* Easy to understand algorithms.
* It is good for students as well.

Cons:

* Not much documentation and online support are available.

*Scikit-learn*

<https://scikit-learn.org/stable/>

[Scikit-learn](https://scikit-learn.org/stable/) simplifies the process of creating classification, regression, dimensionality reduction algorithms, and helps with predictive data analytics. This library is open-source and can be used for both research and commercial purposes. Sklearn is built on [NumPy, SciPy, pandas, and matplotlib](https://www.youtube.com/watch?v=oYTs9HwFGbY), which are indispensable tools for ML programming in Python.

Features:

* It helps in data mining and data analysis.
* It provides models and algorithms for Classification, Regression, Clustering, Dimensional reduction, Model selection, and Pre-processing.

Pros:

* Easily understandable documentation is provided.
* Parameters for any specific algorithm can be changed while calling objects.
* Free

*Jupyter Notebook*

<https://jupyter.org/>

[Jupyter Notebook](https://jupyter.org/) is a command shell for interactive computing. This tool can be used not only with Python, but also with other programming languages: Julia, R, Haskell, and Ruby. It is often used for data analytics, statistical modeling, and machine learning.

Basically, Jupyter Notebook helps with interactive representations of projects in the field of data science. It allows to create beautiful analytics reports and to store and share code, visualizations, and comments.

*Colab*

<https://colab.research.google.com/>

Another handy tool you might want to have if you’re working with Python is Colab. Colaboratory, or simply Colab, allows you to write and execute Python in the browser. It requires zero configuration, gives you access to GPU power, and the results are easy to share.

*PyTorch*

<https://pytorch.org/>

[PyTorch](https://pytorch.org/) is a Python-based open-source framework for deep learning based on Torch. It does GPU-accelerated tensor computing like NumPy. On top of this, PyTorch offers a large library of APIs for programming neural network applications.

PyTorch differs from other machine learning services. Unlike TensorFlow or Caffe2, it doesn’t use static graphs. On the contrary, graphs in PyTorch are [dynamic and calculated on the go](https://datascience.stackexchange.com/questions/45019/static-graphs-v-s-dynamic-graphs). Working with dynamic graphs makes PyTorch easier to work with for some people and allows even beginners to apply deep learning in their projects.

Features:

* It helps in building neural networks through Autograd Module.
* It provides a variety of optimization algorithms for building neural networks.
* PyTorch can be used on cloud platforms.
* It provides distributed training, various tools, and libraries.

Pros:

* It helps in creating computational graphs.
* Ease of use because of the hybrid front-end.

*Keras*

<https://keras.io/>

[Keras](https://keras.io/) is a neural network API that provides a deep learning library for Python. Keras is the most widely-chosen deep learning framework among winning teams on [Kaggle](https://www.kaggle.com/). This is one of the best tools for those who start their career as a machine learning specialist. Compared to other libraries, Keras is much easier to understand. Also, it is more high-level, therefore, it is easier to conceptualize the big picture using Keras. Popular Python frameworks such as TensorFlow, CNTK, or Theano can work with it as well

Features:

* Data preparation
* Classification
* Regression
* Clustering
* Visualization and
* Association rules mining.

Pros:

* Provides online courses for training.
* Easy to understand algorithms.
* It is good for students as well.

Cons:

* Not much documentation and online support are available.

*Knime*

<https://www.knime.com/knime-open-source-story>

You will need [Knime](https://www.knime.com/knime-open-source-story) to work with data analytics and form reports. This open-source machine learning tool integrates numerous components for machine learning and data mining through its modular data pipelining concept. This software has regular releases and excellent support.

One of the big advantages of this tool is that It can integrate the code of various programming languages like C, C++, R, Python, Java, and JavaScript. It can easily be adopted by a team with different programming skills.

Features:

* It can integrate the code of programming languages like C, C++, R, Python, Java, JavaScript etc.
* It can be used for business intelligence, financial data analysis, and CRM.

Pros:

* It can work as a SAS alternative.
* It is easy to deploy and install.
* Easy to learn.

Cons:

* Difficult to build complicated models.
* Limited visualization and exporting capabilities.

*Apache Spark MLlib*

<http://spark.apache.org/mllib/>

[Apache Spark MLlib](http://spark.apache.org/mllib/) is a data processing framework that has an expansive database of algorithms. MlLib is a library that uses Spark, the cluster computing framework. It distributes computing between computers, and that is its main advantage. Among other things, Apache Spark allows you to solve problems connected with classification, clustering, and collaborative filtering.

nside the Apache ecosystem, there is also an open-source framework called [Singa](https://en.wikipedia.org/wiki/Apache_SINGA), which is a software tool for scalable distributed training of deep neural networks.

Apache Mahout

<https://mahout.apache.org/>

[Apache Mahout](https://mahout.apache.org/) is an open-source cross-platform framework for professionals who want to develop machine learning applications that scale. Mahout gives developers the ability to use pre-built algorithms for Apache Spark, H20, and Apache Flink.

At the moment, Apache Mahout algorithms are usually used to build recommender systems (collaborative filtration), for clustering and classification tasks. However, in theory, it can be used to solve any kind of ML-related problem where scalability and performance are important.

Features:

* It provides algorithms for Pre-processors, Regression, Clustering, Recommenders, and Distributed Linear Algebra.
* Java libraries are included for common math operations.
* It follows Distributed linear algebra framework.

Pros:

* It works for large data sets.
* Simple
* Extensible

Cons:

* Needs more helpful documentation.
* Some algorithms are missing.

[*Accord.NET*](http://accord.net/)

<http://accord.net/>

[Accord.NET](http://accord.net/) is a .NET-based machine learning framework written in C#. The platform consists of several libraries covering a wide range of tasks, such as static data processing, machine learning, and pattern recognition. It allows to implement and test a great variety of machine learning algorithms and is well-documented.

Features:

It provides algorithms for:

* Numerical linear algebra.
* Numerical optimization
* Statistics
* Artificial Neural networks.
* Image, audio, & signal processing.
* It also provides support for graph plotting & visualization libraries.

Pros:

* Libraries are made available from the source code and also through executable installer & NuGet package manager.

Cons:

* It supports only. Net supported languages.

Shogun

<https://www.shogun-toolbox.org/>

[Shogun](https://www.shogun-toolbox.org/) is an open-source machine learning solution that focuses on Support Vector Machines (SVM). It is written in C++. This framework offers a wide range of unified machine learning methods based on reliable and understandable algorithms. Shogun can be used by regular programmers for a wide range of standard and cutting-edge tasks. Scientists can apply it for quick prototyping and flexible embedding in workflows. The tool supports many languages (Python, R, Java/Scala, C#, Ruby) and platforms (Linux/Unix, macOS, and Windows) and easily integrates with scientific computing environments.

Features:

* It provides support vector machines for regression and classification.
* It helps in implementing Hidden Markov models.
* It offers support for many languages like – Python, Octave, R, Ruby, Java, Scala, and Lua.

Pros:

* It can process large data-sets.
* Easy to use.
* Provides good customer support.
* Offers good features and functionalities.

**Ejercicio 3**

CRISP-DM (CRoss Industry Standard Process for Data Mining) es un modelo de proceso de 6 fases que describen el ciclo de vida de la ciencia de datos. Sirve para planificar, organizar e implementar un proyecto de ciencia de datos.

El método propuesto por CRISP-DM incluye, en el nivel jerárquico superior, seis fases, a saber:

* *Business Understanding*: Se trata de entender los objetivos del proyecto y los requisitos desde un punto de vista de negocio y convertir esas necesidades en un plan de data mining.
* *Data Understanding*: Se realiza una recolección inicial de datos con la intención de familiarizarse con ellos, detectar problemas de calidad del dato, obtener ya alguna conclusión o ‘insight‘ preliminar y, eventualmente, descubrir subconjuntos de datos interesantes.
* *Data Preparation:* Construye, a partir de los datos en bruto, los conjuntos de datos (‘datasets‘) que alimentarán finalmente los modelos. Incluye la selección de datos, su limpieza, transformación, etc
* *Modeling*: Quizá la fase central (aunque no necesariamente la que más tiempo ocupa) y que consiste en la selección y aplicación de diferentes modelos de machine learning / data mining. Con frecuencia, y durante esta fase, se plantea la necesidad de retomar la fase anterior para adaptar o refinar datos.
* *Evaluation*: Se trata de valorar si los modelos construidos satisfacen las necesidades de negocio. Al final de esta fase se decide si los resultados obtenidos consiguen los objetivos perseguidos y si, por tanto, se puede pasar a desplegarlos.
* *Deployment*: Una fase más técnica y de software que consiste en desplegar los modelos definidos para que puedan ser usados ya en los procesos de análisis y decisión de la compañía.

Otras metodologías similares son KDD, Semma y TDSP

KDD y SEMMA son casi identicos, cada fase de KDD corresponde directamente con una fase de SEMMA. El proceso CRISP-DM combina las etapas Selection-Preprocessing (KDD) o Sample-Explore (SEMMA) en Data Understanding

Una diferencia importante deCRISP-DM es que las transiciones entre etapas se pueden revertir

El proceso de ciencia de datos en equipo (TDSP) es una metodología de ciencia de datos ágil e iterativa para proporcionar soluciones de análisis predictivo y aplicaciones inteligentes de manera eficiente. TDSP ayuda a mejorar la colaboración y el aprendizaje en equipo al sugerir cómo los roles de equipo funcionan mejor juntos. TDSP incluye procedimientos recomendados y estructuras de Microsoft y otros líderes del sector para ayudar a implementar correctamente iniciativas de ciencia de datos. El objetivo es ayudar a las empresas a que se den cuenta de las ventajas de su programa de análisis.

Las etapas de TDSP son:

1. Business Understanding
2. Data Acquisition & Understanding
3. Modeling
4. Deployment
5. Customer Acceptance

**Ejercicio 5**

**Data Set**

Adult Data Set: Utilizado para predecir si el ingreso de un persona supera los 50K al año,

basado en la información de un censo.

**Descripción de atributos:**

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov,

Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th,

7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed,

Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty,

Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing,

Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany,

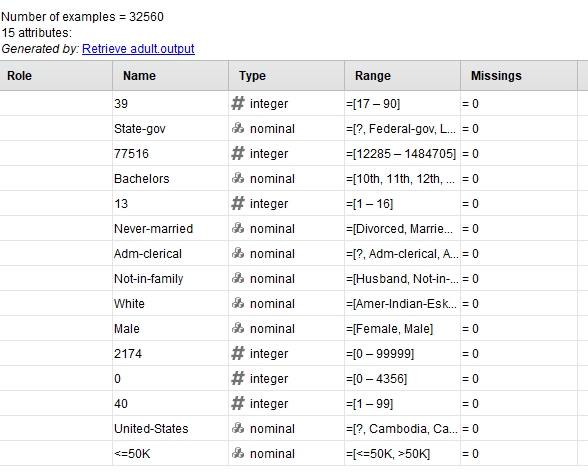
Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran

Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland,

France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary,

Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador,

Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

****

Los algoritmos que se pueden aplicar son Decision Trees y Support Vector Machines