

Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo

Esperanza Manrique Rojas¹

emanrique@uabc.edu.mx

¹ Facultad de Contaduría y Administración, Universidad Autónoma Baja California, Tijuana, Baja California, México.

Pages: 586–599

Resumen: En la industria de desarrollo de software, existe un gran número de lenguajes de programación cada uno con el objetivo de dar respuesta a la complejidad de manejo de datos para generar información oportuna y a la aportación de innovaciones tecnológicas. El lenguaje de elección se decide, algunas veces, por el nivel de comodidad y la experiencia previa del desarrollador. Machine Learning (ML) automatiza la construcción de modelos de análisis de datos, rama de la inteligencia artificial que basa su desarrollo en sistemas que aprenden de datos, identifican patrones y toman decisiones. Se ha identificado que más personas están se encuentran interesadas en aprender sobre ML y a su vez, los lenguajes de programación que se pueden utilizar para el desarrollo de estas aplicaciones. Seleccionar cual es el mejor o adecuado lenguaje de programación, es una tarea difícil. Este capítulo tiene como objetivo describir las características que distinguen cada uno de los lenguajes de programación y algunas herramientas de software que se utilizan en el desarrollo de aplicaciones ML.

Palabras-clave: Machine Learning, lenguajes de programación, Python, Julia, R, Matlab.

Machine Learning: analysis of programming languages and development tools

Abstract: The industry is flooded with innumerable programming languages each with the objective of solving the complexities of the business and bringing technological innovations. The language of choice is sometimes decided by the level of comfort and previous experience of the developer. Machine Learning (ML) automates the construction of data analysis models, a branch of artificial intelligence that bases its development on systems that learn from data, identify patterns and make decisions. For this reason, more and more people are interested in learning about ML and the programming languages they can use for development, but a question arises when it starts and you should select the one that should be the best programming language. This chapter aims to describe the characteristics that distinguish each of the programming languages and some software tools used in the development of ML applications.

Keywords: Machine Learning, programming languages, Python, Julia, R, Matlab.

1. Introducción

La popularidad de la Inteligencia Artificial, en especial la rama de Machine Learning (ML) incrementado en los últimos tiempos, la de desarrollar e implementar aplicaciones diversas para la resolución de problemas, ha crecido en de forma exponencial y donde se incluyen múltiples dominios industriales. Es por esto que hoy en día, muchos profesionistas del área de tecnologías y los que no son propiamente de esta área, han demostrado el interés en aprender sobre ML y, a su vez, los lenguajes de programación que pueden utilizar, pero surge una duda al comenzar el proceso de desarrollo de la aplicación, para esto debe seleccionar el mejor o adecuado lenguaje de programación. Existen un gran número de lenguajes de programación para resolver las complejidades del negocio y aportar innovaciones tecnológicas. Cada año surgen lenguajes de programación, marcos y tecnologías, pero pocos sobreviven al paso del tiempo.

Algunas veces el lenguaje que se selecciona para el desarrollo es por las experiencias y la comodidad del desarrollador. La ventaja de seleccionarlo de esta forma es, que ayuda a crear y desarrollar rápidamente prototipos, aunque a largo plazo puede no ser la mejor opción para desarrollar dicha aplicación, y una de las razones podría ser la cantidad de procesos de datos que se necesitara. Algunos de los lenguajes de programación utilizados en ML son R, Python, Julia y Matlab, entre muchos otros.

2. Machine Learning

ML resuelve situaciones por sí solo a partir de un análisis de datos y cuantos más datos tengan mejores resultados, además, para realizar el análisis se utilizan algoritmos que diseñan otros datos según las necesidades (Juárez, G. 2017). A través de los datos de entrada, ML ejecutar un algoritmo y como resultado, genera más información para el problema (Bishop, C., 2007). El objetivo de generar más datos se basa en las siguientes técnicas:

- Regresión lineal y polinómica.
- Árboles de decisión.
- Redes neuronales.
- Red bayesiana.
- Cadenas de Markov.

Estas técnicas permiten a ML reconocer patrones, extraer conocimiento, descubrir información y hacer predicciones.

Se considera que cada persona aprende de una manera particular, utiliza los sentidos, la experiencia y sus habilidades cognitivas, también puede confiar en estrategias personales y técnicas de aprendizaje, por ejemplo, tomar notas, resolver ejercicios, leer, memorizar, marcar libros. En el entorno informático, se pretende lograr que las computadoras alcancen la autonomía (Jordan, M. I., & Mitchell, T. M., 2015) y de esta forma aprendan de forma automática sus propias habilidades que se definen con los algoritmos para el aprendizaje y la gestión de datos.

El ML no es auto programación, sino auto aprendizaje de datos y experiencia para generar patrones y resolver nuevas tareas. Este aprendizaje es la combinación de

técnicas, datos, conceptualización de análisis de datos y algoritmos para generar nuevos patrones o modelos de predicción.

2.1. Aplicaciones comunes de Machine Learning

Las aplicaciones del ML se pueden identificar en aplicaciones de uso cotidiano como Twitter o Facebook, por ejemplo, cuando agrega un amigo o una persona a través de las famosas recomendaciones, o incluso en YouTube es muy común viajar a través de los videos recomendados, muchos del usuario de estas herramientas y otras, han experimentado estas recomendaciones (Ingeniero, 2015).

Para tener una mayor claridad sobre el ML, a continuación, se describen algunas áreas en las que este tipo de tecnología se usa comúnmente:

- Detección de correos electrónicos no deseados (spam): mediante la detección de texto en correos electrónicos recibidos, los algoritmos de ML los clasifica como no deseados (Lenin J., 2017) y (Ingeniero, 2015).
- Detección de patrones en imágenes: esta función se encuentra en la cámara fotográfica en la que se detectan las sonrisas de las personas que se identifican y una vez que están sonriendo, se realiza la toma automáticamente. El desafío aquí es identificar que la persona está sonriendo dadas todas las características faciales que una persona puede tener (Ingeniero, 2015).
- Otra Aplicación muy utilizada es AMAZON, la cual da recomendaciones de productos de acuerdo a los patrones de compra que identifica, es fascinante esta aplicación dentro del mundo del aprendizaje automático (Ingeniero, 2015) y Bishop, C. (2007).

2.2. Tipos de aprendizaje de Machine Learning

Para comprender esta actividad, es importante conocer la clasificación de los tipos de algoritmos de aprendizaje de ML que se describirán a continuación:

- Aprendizaje supervisado: se enseña al algoritmo cómo realizar su trabajo (Rojas, E. 2018), con un conjunto de datos clasificados bajo una cierta apreciación o idea para encontrar patrones que puedan aplicarse en un análisis (Mueller, J., & Massaron, L., 2016) y producir una salida que ya se conoce.
- Aprendizaje no supervisado: se define como un modelo predictivo entrenado de manera similar al aprendizaje supervisado, pero la diferencia es que la comprensión se da en datos no clasificados o etiquetados y descubre patrones de ejemplos similares entre grupos de datos (Bishop, C., 2007).
- Aprendizaje reforzado: es un tipo de aprendizaje automático en el que no hay capacitación con datos clasificados o no clasificados; el sistema aprende en un entorno donde no hay información sobre la posible salida (López, J., López, B., & Díaz, V., 2004), a través de acciones y los resultados obtenidos, como lo mencionan Mueller, J., & Massaron, L. (2016) además, el modelo se refuerza al resolver el problema de la mejor manera.

2.3. Pasos para construir un modelo de Machine Learning

La creación de un modelo de ML no se limita a usar un algoritmo de aprendizaje o una biblioteca de ML; es un proceso que generalmente implica al menos 6 pasos, que se muestran en la Figura 1.



Figure 1 – Machine learning model construction process

Lopez, R. (2015), describe cada uno de estos pasos como se describe a continuación:

1. Recolectar los datos: Los datos se pueden recolectar de fuentes tal como un sitio web, utilizando una API o una base de datos. Este paso es uno de los más complicados y requiere un tiempo determinado.
2. Preprocesamiento de los datos: Con los datos disponibles, se debe asegurar que todos tengan un formato correcto para alimentar el algoritmo de aprendizaje. Por lo general se tiene que realizar varias tareas de preprocesamiento antes de poder usar los datos.
3. Explore los datos: Se realiza un análisis previo para corregir los casos de valores faltantes o tratar de encontrar a primera vista cualquier patrón en ellos que facilite la construcción del modelo. En este punto, se deben detectar valores atípicos; o encuentre las características que tienen más influencia para hacer una predicción.
4. Entrena el algoritmo: los algoritmos de aprendizaje se alimentan con los datos que se procesaron en las etapas anteriores. La idea es que los algoritmos pueden extraer información útil de los datos iniciales y luego hacer las predicciones.
5. Evaluar el algoritmo. Se realizan las pruebas de la información que genera el conocimiento del entrenamiento previo que se obtuvo a través del algoritmo.

Se realiza una evaluación sobre la precisión del algoritmo en sus predicciones y, si no está satisfecho con su rendimiento, debe volver a la etapa anterior y continuar entrenando el algoritmo cambiando algunos parámetros hasta que se logre un rendimiento aceptable.

3. Lenguajes de desarrollo de Machine Learning

Existen varios lenguajes de programación que se utilizan para construir aplicaciones de IA y ML. Cada aplicación tiene sus propios requisitos y restricciones, y algunos lenguajes podrían ser mejores que otros según los problemas específicos. Los lenguajes de programación han evolucionado, y otros se han creado en función de los requisitos únicos de las aplicaciones de IA.

Este capítulo solo aborda los lenguajes de programación Python, R, que son los más comunes para este desarrollo, Julia por ser un nuevo lenguaje para IA y Matlab, por su orientación matemática y estadística (fig. 2).



Figure 2 – Programming languages for Machine Learning development

3.1. Lenguaje Python

Python es un lenguaje donde su código se ejecuta en el navegador al cargar la página, es independiente de la plataforma y orientado a objetos, está listo para realizar cualquier tipo de programa desde aplicaciones de Windows hasta servidores de red o incluso páginas web. Es un lenguaje interpretado, lo que ofrece ventajas como la velocidad de desarrollo e inconvenientes como una velocidad más baja al ser ejecutado (Álvarez, M., 2003).

En los últimos años, este lenguaje se ha vuelto muy popular y algunas de las razones son las siguientes:

- El número de bibliotecas que contiene, los tipos de datos y las funciones incorporadas en el lenguaje (Álvarez, M., 2003) y (Equipo de Expertos, 2019).
- Python es gratis, importante: incluso para fines comerciales.
- La simplicidad y velocidad con la que se crean los programas. Un programa en Python menos líneas de código que su equivalente en Java o C, (Álvarez, M., 2003).

- El número de plataformas en las que se puede desarrollar, como Unix, windows, OS/2, Mac y otras (Roman V, 2015).

3.1.1. Características de Python para el desarrollo de Machine Learning:

1. La asociación Python de ML: se ha visto favorecida por aplicaciones que van desde el desarrollo web hasta la automatización de scripts y procesos. (Roman V, 2015).
2. Amplia selección de bibliotecas y marcos: Uno de los aspectos que hace que Python sea una opción tan popular en general es su abundancia de bibliotecas y macros que facilitan la codificación y ahorran tiempo en el desarrollo.
3. Código legible y consiso: facilidad de uso y simplicidad, especialmente para los nuevos desarrolladores. El aprendizaje profundo se basan en algoritmos extremadamente complejos y flujos de trabajo de múltiples etapas (Lopez, R., 2015) y (Roman V, 2015).
4. Agilidad: La sintaxis simple de Python significa que también es más rápido en desarrollo que muchos lenguajes de programación y permite al desarrollador probar algoritmos rápidamente sin tener que implementarlos (Roman V, 2015).
5. Colaboración: fácil de leer es de gran valor para la codificación cooperativa, o cuando los proyectos de Python de Deep Learning o ML cambian de manos entre los equipos de desarrollo (Expertos, 2019) y (Roman V, 2015).
6. Python es un lenguaje de programación de código abierto y está respaldado por una gran cantidad de recursos y documentación de alta calidad.

3.1.2. Bibliotecas de Python para Machine Learning

Una de las grandes ventajas que ofrece Python sobre otros lenguajes de programación; es cuán grande y proliferante es la comunidad de desarrolladores que lo rodea; comunidad que ha contribuido con una gran variedad de librerías de primer nivel que amplían las funcionalidades del lenguaje (Roman V, 2015). En el caso de ML, las principales bibliotecas que se utilizan son las descritas en la Tabla 2.

Librería	Aplicación o Uso
Scikit-Learn	Se utiliza para clasificaciones, extracción de características, regresiones, agrupaciones, reducción de dimensiones, selección de modelos o preprocesamiento. (Pedregosa, Varoquaux, Gramfort, et. Alabama, 2011).
Statsmodels	Es otra gran biblioteca que se centra en modelos estadísticos y se utiliza principalmente para análisis predictivos y exploratorios. (Seabold, S. y Perktold, J., 2010).
PyMC	Implementa modelos estadísticos bayesianos, incluida la cadena Markov Monte Carlo (MCMC). Ofrece funcionalidades para hacer que el análisis Baiesiano sea lo más simple posible (Patil, A., Huard, D. y Fonnesbeck, C. J., 2010).
NLTK	Es la biblioteca líder para el procesamiento del lenguaje. Proporciona interfaces fáciles de usar para más de 50 cuerpos y recursos léxicos, como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, etiquetado, análisis y razonamiento semántico (Patil, A., Huard, D., y Fonnesbeck, CJ, 2010).

Tabla 1 – Librerías de Python

3.2. Lenguaje R

R es un lenguaje de programación especialmente orientado al análisis estadístico y a la representación gráfica de los resultados obtenidos. Es un proyecto de GNU por lo que los usuarios son libres de modificarlo y extenderlo. R se distribuye como software libre bajo la licencia GNU y es multiplataforma (hay versiones para plataformas Windows, Mac y Linux, y de hecho algunas distribuciones de Linux lo tienen incorporado), lo que también ha facilitado su adopción y la existencia de una comunidad muy activa su entorno, con constantes desarrollos de nuevas funcionalidades y versiones mejoradas de las existentes. Es un lenguaje basado en comandos, en lugar de hacer clic y arrastrar iconos o menús con el mouse, escribe comandos o instrucciones que se ejecutan. Una secuencia de instrucciones o comandos R que implementa un flujo de trabajo para realizar una tarea se denomina script o script R. (Perkins, J., 2014).

3.2.1. Características principales como lenguaje (*Lenguaje de Programación, SF*) son:

- Posibilidad de crear gráficos, basados en LaTeX.
- Gran cantidad de herramientas estadísticas:
 - Modelos lineales y no lineales.
 - Pruebas estadísticas.
 - Algoritmos de clasificación y agrupación.
- Posibilidad de crear sus propias funciones, así como objetos, ya que su programación es POO (orientada a objetos).
- Integración con diferentes bases de datos.
- Puede tener un uso matemático, como sustitución de MATLAB.

El verdadero potencial de R reside precisamente en aquellas bibliotecas (algunas ya incorporadas en el sistema base y otras para ser instaladas) que permiten que R se destaque de otros entornos e lenguajes para el análisis estadístico, principalmente en las funcionalidades relacionadas con la manipulación de datos, visualización de gráficos y disponibilidad de algoritmos de aprendizaje automático. (Industria 4.0, 2015).

3.2.2. Contribución del lenguaje R al análisis de datos

R tiene algunas características especiales que lo hacen versátil para el manejo de elementos estadísticos, específicamente para operaciones con matrices y vectores, lo que facilita la manipulación de bases de datos. Por lo tanto, R le permite manipular datos muy rápidamente. (Rochina P., 2016).

En cuanto al aprendizaje automático, R ha implementado una gran cantidad de algoritmos, como consecuencia de las diferentes líneas de investigación de grupos que llevaron a su creación, debido precisamente al hecho de que R nació en el campo académico (Rochina P., 2016).

Frente a R, es su curva de aprendizaje, que generalmente es más lenta y complicada en comparación con Python.

3.2.3. Bibliotecas para ciencia de datos

Algunas de las bibliotecas que se han vuelto extremadamente útiles para analizar y manipular datos con R son:

- Tidyverse que explora, ordena y analiza datos, además este paquete incluye las bibliotecas ggplot2, tibble, tidyr, readr, purrr y dplyr; que comparten una filosofía propia y están diseñados para trabajar naturalmente entre ellos.
- Caret: cuando se trata de simplificar el proceso de ML este paquete ofrece una serie de herramientas para la construcción de modelos de ML en R. Caret proporciona herramientas esenciales para: la etapa de preparación de datos, dividir el conjunto de datos, seleccionar los atributos principales y evaluar los Modelos.

Otras bibliotecas de lenguaje R si se trata de organizar grandes volúmenes de datos de forma intuitiva, data.table es el paquete indicado. En la Tabla 3, se describen las bibliotecas para el desarrollo de ML (Lopez, SF).

Librería	Aplicaciones o Uso
RandomForest	Para trabajar específicamente con modelos Random Forest, el paquete randomForest puede ser una buena opción; Este paquete permite crear este tipo de modelos de una manera muy simple.
Rpart	El paquete rpart es una buena alternativa para trabajar con árboles de clasificación. Implementa los principales algoritmos para trabajar con este tipo de modelos.
Igraph	Para analizar y visualizar redes y gráficos, el paquete igraph es la mejor opción. Este paquete proporciona una serie de rutinas altamente eficientes para visualizar y analizar las conexiones de las redes
Outliers	Si lo que necesita es encontrar valores atípicos. Esta biblioteca ofrece varias funciones y pruebas para identificar valores atípicos.
Survival	Es un paquete que facilita la realización de análisis de supervivencia.
Forecast	Proporciona métodos y herramientas para mostrar y analizar predicciones univariadas de series de tiempo, incluido el suavizado exponencial a través de modelos de espacio de estado y el modelado automático ARIMA
Nnet	Las redes neuronales han recibido mucha atención últimamente debido a sus habilidades para aprender las relaciones entre las variables. Representan una técnica innovadora para adaptar modelos que no se basa en los supuestos convencionales requeridos por el modelado estándar; y eso también puede manejar datos multivariados de manera muy efectiva. Un gran paquete para trabajar con redes neuronales de una manera muy simple es nnet.

Table 2 – Programming Language Library R.

3.3. Lenguaje matlab

La plataforma MATLAB está optimizada para resolver problemas científicos y de ingeniería. El lenguaje MATLAB, basado en matrices, es la forma más natural de expresar las matemáticas computacionales en el mundo. Los gráficos integrados facilitan la visualización de los datos y la obtención de información de ellos. Una amplia

biblioteca de herramientas integradas (Toolboxes) le permite comenzar a trabajar de inmediato con algoritmos esenciales para su dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y funciones de MATLAB son rigurosamente probadas y diseñadas para trabajar juntas (Lopez, SF).

Gracias al uso de MATLAB, los ingenieros y otros expertos han implementado miles de aplicaciones para mantenimiento predictivo, análisis de sensores, finanzas y electrónica de comunicaciones. MATLAB facilita las partes más difíciles del aprendizaje automático gracias a (MathWorks for Machine Learning, SF):

- Técnicas de procesamiento de señales y extracción de funciones avanzadas.
- Ajuste de hiperparámetros y selección de funciones automáticas para optimizar el rendimiento de los modelos.
- Posibilidad de usar el mismo código para escalar el procesamiento de big data y clusters.
- Generación automática de código C / C ++ para aplicaciones integradas y de alto rendimiento.
- Todos los algoritmos comunes de clasificación, regresión y agrupamiento para el aprendizaje supervisado y no supervisado.
- Ejecución más rápida que con código abierto en la mayoría de los cálculos estadísticos y el aprendizaje automático.

Con MATLAB, los ingenieros y analistas de datos tienen acceso inmediato a funciones prediseñadas, cajas de herramientas integrales y aplicaciones especializadas para clasificación, regresión y agrupamiento (MathWorks for Machine Learning, SF).

MATLAB le permite hacer lo siguiente (MathWorks for Machine Learning, SF):

- Compara enfoques como regresión logística, árboles de clasificación, máquinas de vectores de soporte, métodos combinados y aprendizaje profundo.
- Utiliza las técnicas de reducción y refinamiento del modelo para crear un modelo preciso que capture mejor el poder predictivo de sus datos.
- Integra modelos de aprendizaje automático en sistemas, clústeres y nubes empresariales, y dirige los modelos al hardware integrado en tiempo real.
- Genera código automáticamente para el análisis de sensores integrados.
- Emplea flujos de trabajo integrados desde el análisis de datos hasta la implementación.

Para el análisis de datos multidimensionales, Statistics and Machine Learning Toolbox proporciona selección de características, regresión gradual, análisis de componentes principales (PCA), regularización y otros métodos de reducción de dimensionalidad que permiten identificar variables o características que impactan su Modelo.

El kit de herramientas proporciona algoritmos de aprendizaje automático supervisados y no supervisados, que incluyen máquinas de vectores de soporte (MVS), árboles de decisión potenciados y en bolsas, vecinos k más cercanos, medios k, medoides k, agrupamiento jerárquico, modelos de mezcla gaussiana y modelos de Markov ocultos (MathWorks, Comparación entre MATLAB y Python, SF).

3.4. Lenguaje Julia

Julia es un lenguaje con tipos dinámicos y buen soporte para uso interactivo. La versión 1.0 de Julia finalmente se lanzó y los desarrolladores dicen que el buen rendimiento que han logrado se debe a que Julia fue diseñada desde el principio para un alto rendimiento. Los programas de Julia compilan código nativo de manera eficiente en múltiples plataformas gracias al esquema desarrollado por LLVM (Máquina virtual de bajo nivel (Analytics Line, 2019)). La sintaxis fue diseñada pensando que era útil en matemáticas.

Si se trata de utilizar a Julia como un lenguaje de propósito general, se proporciona una biblioteca estándar que proporciona entradas / salidas asíncronas, control de procesos, así como un administrador de paquetes, entre otras cosas. Los cambios desde la versión 0.6 comienzan con un nuevo administrador de paquetes que es más rápido que la versión anterior y que admite entornos de paquetes por proyecto (Analytics Line, 2019) y Lopez M. (2018).

La nueva versión también ofrece una nueva representación canónica de los valores faltantes para que los desarrolladores de Julia puedan representarlos y trabajar con información incompleta. Cualquier tipo de colección genérica admite estos datos perdidos simplemente permitiendo la inclusión de elementos con los valores predefinidos perdidos.

Otras mejoras incluyen cambios en el tipo de cadenas interconectadas, es decir, de cadenas de caracteres para poner datos de manera arbitraria. Todos los datos de la cadena de caracteres se conservan mientras se indican qué caracteres son válidos o no válidos. .

3.4.1. Julia and Machine Learning

A medida que los sistemas de aprendizaje automático se vuelven más complejos, los investigadores necesitan cada vez más lenguajes diferenciables en los que simplemente puedan escribir sus algoritmos como código. La sintaxis matemática de Julia lo convierte en una forma ideal de expresar algoritmos tal como están escritos en documentos, mientras que Flux convierte el código en modelos entrenables con diferenciación automática, aceleración de GPU y soporte para terabytes de datos a través de JuliaDB.

3.4.2. Ventajas de lenguaje Julia sobre Python

Estas son las principales ventajas que tiene Julia sobre el lenguaje Python (Analytics Line, 2019):

1. Velocidad: en su estado predeterminado, el lenguaje Julia es aún más rápido que Python. Esto es posible porque Julia usa tanto las declaraciones de tipo como la compilación JIT (Just in time). Las versiones no optimizadas de la programación de Python no pueden igualar la velocidad de Julia. Sin embargo, puede aumentar la velocidad de Python utilizando compiladores de terceros como PyPy y otras bibliotecas externas.
2. Gestión automática de la memoria: con Julia, no se sobrecargará con las tareas de liberar y asignar memoria. El lenguaje hace estas tareas por usted. El lenguaje

- proporciona medidas efectivas para la recolección de basura. Sin embargo, esto también es una característica del lenguaje Python.
3. Sintaxis orientada a las matemáticas: es de conocimiento común que el lenguaje Julia fue creado especialmente para cálculos científicos que se utilizan en un entorno matemático. La sintaxis de este lenguaje de programación es bastante similar a las fórmulas matemáticas que normalmente se usan para otras operaciones matemáticas que no son solo ciencias de la computación. Por esta razón, Julia puede ser fácilmente entendida por no programadores.
 4. Paralelismo: no podemos negar el hecho de que tanto Julia como Python usan el paralelismo para la gestión de recursos. Sin embargo, Julia es menos pesada en términos de los recursos que usa en comparación con Python.

El rendimiento se logra mediante el uso de un compilador JIT en lugar de un intérprete. Lo que significa que la primera ejecución del código es más lenta ya que el compilador tiene que analizar y compilar el código. Ser capaz de significar una desventaja en ciertas ocasiones con respecto a otros lenguajes. Los usuarios de Julia indican que es tan fácil de usar como Python, R o Matlab.

4. Estudio comparativo de lenguajes de programación ML

Conociendo algunos de los lenguajes de programación de Machine Learning, en la Tabla 4 enumera y describe los factores que afectan a cada uno de ellos (González L, 2018).

Factores	Descripción
Velocidad	Al elegir el mejor lenguaje de programación, la velocidad es esencial. R fue desarrollado básicamente como un lenguaje estadístico, esto significa que tiene un mayor análisis de datos y soporte estadístico. Por su parte, Python depende de los paquetes, por lo tanto, cuando se trata de tareas relacionadas con estadísticas, R tiene una ventaja en comparación con Python y es un poco más rápido.
Curva de aprendizaje	Cuando se trata de la perspectiva funcional, R es el lenguaje de programación, mientras que cuando se trata de estar orientado a objetos, Python es el lenguaje. Si pertenece al grupo de programadores funcionales, entonces aprender Python será mucho más fácil en comparación con R. Cuando llegue a Matlab y Julia, ambos son similares a escribir algunas ecuaciones matemáticas, y sí, son fáciles de aprender e implementar.
Costo	El único lenguaje que se paga y necesita una licencia para su uso es Matlab. Los otros tres lenguajes son de código abierto y es completamente gratuito. Por lo tanto, cuando tiene recursos gratuitos disponibles, ¿por qué alguien elegiría pagar? Esta es la razón por la cual Matlab se retrasa un poco en comparación con otros lenguajes.
Comunidad para soporte	Todos los lenguajes de programación son muy populares en el mercado y cuentan con un gran apoyo de la comunidad. Aunque debe mencionarse que Python es la que tiene la comunidad más grande en Internet que es bastante solidaria en el momento de un problema con los desarrollos.

Table 3 – Factores que afectan cada uno de los lenguajes de programación.

Cuando se trata de análisis estadístico, R es el lenguaje de programación apropiado. Si se trata de tareas relacionadas con la visión por computadora, Matlab es la opción

preferida. Si se trata de tareas relacionadas con la bioinformática o la biología, entonces Julia es el lenguaje de programación elegido. Pero si se trata de tareas generales como el procesamiento de datos y el procesamiento de resultados, entonces Python es el lenguaje de programación más apropiado

5. Conclusión

El uso de uno y otro lenguaje debe dar respuestas a las siguientes preguntas:

¿Qué tipo de problema quieres resolver? Uno u otro se seleccionará de acuerdo con el tipo de análisis de datos que se debe llevar a cabo, ya sea aprendizaje automático, minería de datos, análisis web, etc. Por lo tanto, R es una muy buena opción cuando el análisis de datos requiere un cálculo independiente o un análisis individual en los servidores, mientras que Python lo usará cuando el análisis de datos requiera integrarse con las aplicaciones web o si es necesario incorporar la estadística del código de análisis en una base de datos de producción.

¿Cuánto me costará aprender a usar el lenguaje de programación? Debe tenerse en cuenta, dependiendo del conocimiento previo en programación, estadísticas, etc., si R o Python serán más fáciles de aprender y poner en práctica, debe tenerse en cuenta que el objetivo es resolver el problema. Eso existe entre manos, y aprender a usar las herramientas para resolverlo no debería convertirse en el núcleo del problema en sí.

¿Con qué herramientas se integrarán al entorno a trabajar? Debe conocer el entorno en el que trabaja y qué programas para big data e inteligencia de negocios se manejarán, para elegir el lenguaje de programación que mejor se integra con las herramientas.

Responder estas preguntas puede guiar el uso de uno u otro lenguaje, cada lenguaje tiene sus propias características, que pueden evaluarse para adoptar el que mejor se adapte al problema a resolver.

Referencias

- Álvarez, M. (2003). Lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web. Retrieved from: <https://desarrolloweb.com/articulos/1325.php>
- Analytics Line, (2019), Accessed April 19 2019, from <https://www.analyticslane.com/2018/09/24/conoces-el-lenguaje-de-programacion-julia/>
- Bishop, C. (2007). Pattern recognition and machine learning. New York, NY: Springer.
- Contreras, F. (2016). Introduction al machine learning. Accessed November 10 2018, Retrieved from: https://www.zemsania.com/recursos-zemsania/whitepapers/DTS/Machine_learning.pdf
- Equipo de Expertos Universidad Internacional de Valencia. (2019). Machine Learning Python: el lenguaje de los negocios del futuro. Retrieved from: <https://www.universidadviu.com/machine-learning-python-el-lenguaje-de-los-negocios-del-futuro/>

- Gonzalez L. (2018). Lenguajes de programación para Machine Learning, Accessed May 11 2019, Retrieved from: <http://ligdigonzalez.com/lenguajes-de-programacion-para-machine-learning/>
- Industria 4.0, (2015). Big Data Analytics, emprendimiento digital y Nuevos modelos de negocio. El lenguaje de programación R para análisis Estadístico y machine learning. Accessed April 10 2019, from <http://www.mikelnino.com/2015/04/R-lenguaje-programacion-analisis-estadistico-machine-learning.html>
- Ingeniero Beta. (2015) Applications of Machine Learning . from <http://ingenierobeta.com/machine-learning-aplicaciones/>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects, Retrieved from: http://science.sciencemag.org/content/349/6245/255?casa_token=PngLtzsuefoAAAAA:w2Nq8oJ899bxwJ5nyLMEco1l9nJx6_O3ocobyne8JJfw28Q2k_FHMT6DHcoFvjsIa21Hn5Fa_mylOeEW
- Juárez, G. (2017). ¿Cómo funciona el aprendizaje automático (Machine Learning) Retrieved from: <http://www.nexolution.com/como-funciona-el-aprendizaje-automatico-machine-learning/>
- Lenguajes de Programación (SF). Lenguaje R. Accessed April 15 2019, from <https://lenguajesdeprogramacion.net/r/>
- Lenin, J. (2017). Pin G.1, Filtrado de SPAM en SMS mediante algoritmos de aprendizaje automático Maskana - Ciencias de la Computación, Retrieved from: <https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/download/>
- Lopez B.(SF) Libro de IAAR, Accessed April 14 2019, from <https://iaarbook.github.io/rlang/>
- López, J., López, B., & Díaz, V. (2004). ALGORITMO DE APRENDIZAJE POR REFUERZO CONTINUO PARA EL CONTROL DE UN SISTEMA DE SUSPENSIÓN SEMI-ACTIVA. Revista Iberoamericana de Ingeniería Mecánica, 9(2), 77–91., Retrieved from: <https://www.facebook.com/>
- Lopez, R. (2015). Powered by Pelican Machine Learning con Python [Blog] Retrieved from: <https://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/>
- Marr, B. (2018). Artificial Intelligence: What Is Reinforcement Learning - A Simple Explanation & Practical Examples., Retrieved from: <https://www.forbes.com/sites/bernardmarr/2018/09/28/artificial-Intelligence-what-is-reinforcement-learning-a-simple-explanation-practical-Examples/#e5720c9139ce>
- MathWorks for Machine Learning,(SF) Accessed April 19 2019, from <https://la.mathworks.com/solutions/machine-learning.html>
- MathWorks, Comparación entre MATLAB y Python: principales razones para elegir MATLAB (SF). Accessed April 30 2019, from <https://la.mathworks.com/products/matlab/matlab-vs-python.html>

- Mueller, J., & Massaron, L. (2016). Machine learning for dummies. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Patil, A., Huard, D., & Fonnesbeck, C. J. (2010). PyMC: Bayesian stochastic Modelling in Python. *Journal of statistical software*, 35(4), 1. Retrieved from: <https://www.jstatsoft.org/article/view/v035i04>
- Pedregosa, Varoquaux, Gramfort, et. al. (2011) Scikit-learn: Machine learning in Python. *Journal of machine learning Research*, 12(Oct), 2825-2830. Retrieved from: <http://www.jmlr.org/papers/v12/pedregosa11a.html>
- Perkins, J. (2014). Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd. © Copyright 2013, Francisco J. Romero-Campero. © Copyright 2013, Created Using Sphinx 1.1.3. Retrieved from: https://www.cs.us.es/~fran/curso_unia/introduccion_R.html
- Rochina P. (2016), Python vs R para el análisis de datos, Retrieved from: <https://revistadigital.inesem.es/informatica-y-tics/python-r-analisis-datos/>
- Rojas, E. (2018). Glosario de los seis términos básicos del Machine Learning, Retrieved from: <https://www.muycomputerpro.com/2018/02/07/glosario-terminos-basicos-machine-learning>.
- Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical python, Retrieved from: <https://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf>
- Sommer, C., & Gerlich, D. (2013). Machine learning in cell biology – teaching Computers to recognize phenotypes. *Journal of Cell Science*, 126(24), 5529-5539. doi: 10.1242/jcs.123604

© 2020. This work is published under
<https://creativecommons.org/licenses/by-nc-nd/4.0/>(the
“License”). Notwithstanding the ProQuest Terms and
Conditions, you may use this content in accordance with the
terms of the License.