6. INTRODUCCION A MACHINE LEARNING

José J. Martínez P.

josejesusmp@gmail.com
Octubre 2022

6.1 Introducción

Machine Learning es un término que ya se venía utilizando desde el siglo pasado. Sin embargo, ahora se ha convertido en un término comodín que utiliza mucha gente sin conocer su significado real y dependiendo del ambiente y de la formación de la gente le dan diferentes significados, debdo a su utilización. Aunque, su florecimiento no es muy viejo, solo se logró hasta 2005, cuando G, Hinton, publicó un artículo en el que mostraba cómo al entrenar una red neuronal profunda, se podían reconocer dígitos escritos a mano, con una precisión del 98%. Como se ha mencionado, a historia de las redes neuronales ha sido una historia de amores y desamores y a finales de los 90s se había abandonado su investigación en estas redes. Con este artículo se demostró que con Deep Learning era posible lograr cosas que ninguna otra técnica de ML había hecho. El entusiasmo se extendió a muchas otras áreas de aplicación y digamos que estamos viviendo el auge de ML, que va a perdurar bastante.

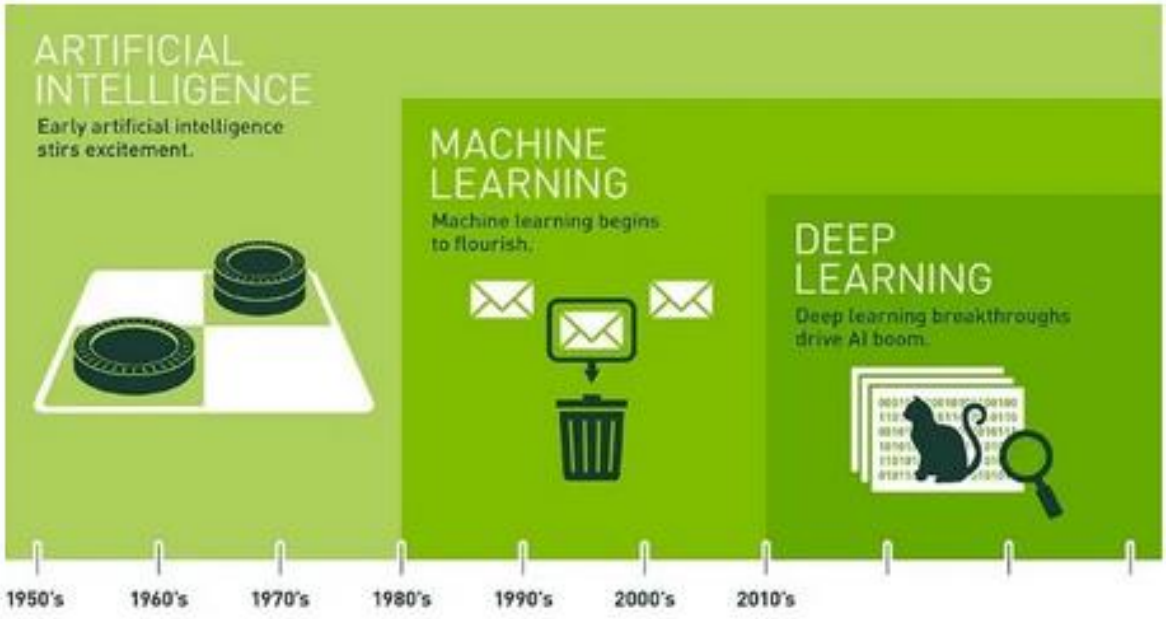


Figura 6.1. Una línea de tiempo de Inteligencia Artificial, Machine Learning y Deep Learnig.

La gráfica de la figura 6.1, refleja en el tiempo los desarrollos que marcan la situación actual de la IA. Desde los 80s se comenzó a trabajar en ML, especialmente enfocado hacia el reconocimiento de objetos o patrones, con base en el concepto de atributos o parámetros

(features) sin embargo, con Deep Learning de alguna manera se dispara su utilización, llevando al reconocimiento de los datos, como fuente de conocimiento, pues en ellos está la información real, valida, de la interacción de humanos y máquinas con el ambiente. Una vez se logran integrar estos conceptos, se encuentra que hay otras técnicas básicas para enfrentar los mismos y otros problemas, especialmente SVM (Support Vector Machine) y árboles de decisión.

Supongamos el siguiente problema: traducir un texto de un idioma a otro y solucionarlo con las técnicas actuales de programación. Para hacer un programa de computador que realice esta tarea, primero, hay que estudiar la gramática de ambos lenguajes, con el fin de codificar sus reglas; algo que no es tan simple. Luego incluir los diccionarios de ambos lenguajes, incluir también la codificación de las búsquedas y seleccionar las palabras más apropiadas en el lenguaje de destino; otra tarea de programación complicadas.

Pero hay más, ¿cómo trabajar con palabras homógrafas u homónimas? Por ejemplo, la palabra "can" en inglés, puede ser el verbo poder o puede ser simplemente una lata, en español. Todo depende del contexto. ¿Pero cómo codificamos apropiadamente ese contexto? Realmente la solución de este tipo de problemas, bajo ese concepto de programación es extremadamente difícil y por esto los traductores de lenguajes, antes de ML eran muy pobres.

Ahora las cosas han cambiado. Se encontró que la solución de muchos problemas complejos se encuentra en los datos. Usando una cantidad gigantesca de datos provenientes de diferentes fuentes en la web, Google tiene un modelo pre-entrenado que va mejorando sus resultados, de manera tan sorprendente, que en algunos casos superan el nivel de un traductor humano.

Ahora, se han desarrollado cientos de aplicaciones, que silenciosamente le dan poder a cientos de productos que usamos regularmente; desde mejores recomendaciones hasta búsquedas por voz.
Nos pueden surgir algunas preguntas:

- ¿Dónde comienza y termina ML?
- ¿Qué significa exactamente, que una máquina aprenda algo?
- Si descargo una copia de Wikipedia ¿aprendió algo mi computador?
- ¿Se volvió inteligente repentinamente el computador?

6.2 ¿Qué es ML?

Veamos varias definiciones de ML:

- "ML es el campo de estudio que le da a los computadores la habilidad para aprender sin que sean explícitamente programados". Arthur Samuel 1959.
- "ML es la ciencia o el arte de programar los computadores, de manera que puedan aprender a partir de los datos".
- Una simple: "Sistemas que mejoran su comportamiento con respecto a una tarea con más y más experiencia o datos".

- “Se dice que un programa de computador aprende de la experiencia **E**, con respecto a alguna tarea **T**, con alguna medida de comportamiento **P**, si su comportamiento sobre **T**, medido por **P**, mejora con la experiencia”. Mas orientada a la ingeniería. Tom Mitchel 1998.
- Otra forma de ver ML. En la gráfica 6.1, se puede observar, ML en comparación con la programación tradicional tomando como ejemplo encontrar la raíz cuadrada de un número. En el caso de la programación tradicional podemos programar el método de Newton y hallar la solución. En el caso de ML, tendríamos muchos ejemplos con resultados correctos de la raíz cuadrada y a través de regresión podríamos encontrar la solución.
-

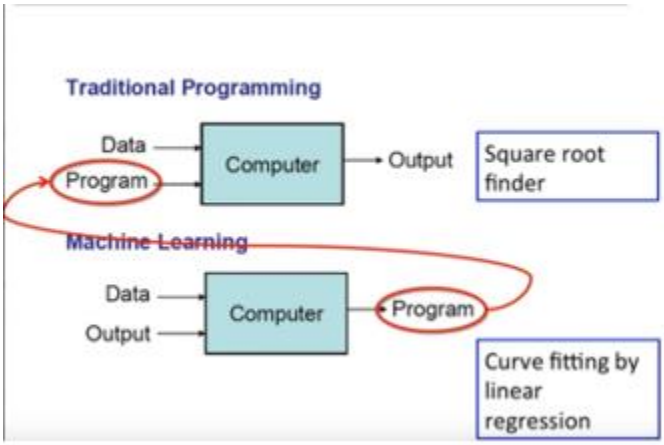


Figura 6.1. Programación tradicional versus ML.

Para tener una mayor claridad, vamos a trabajar alrededor de un ejemplo que es el filtro que tienen los sistemas de correo electrónico para detectar correos engañosos, esto se conoce como filtros SPAM.

El filtro spam que usa nuestro correo, es un programa ML, que, con base en instancias de emails marcados como spam, observa qué mensajes del correo marca o no como spam; y con que, instancias de correos normales, aprende cómo filtrar mejor el spam. Esas instancias de emails que usa el sistema para aprender se llaman el conjunto de entrenamiento, o dataset.

Volviendo a la última definición de ML, la tarea T del filtro Spam es marcar los correos que llegan como spam, correo no deseado, cuando esos correos presentan ciertas características, que el filtro Spam las tiene clasificadas como comunes a todos los correos spam. La experiencia E, es el conjunto, dataset, de correos buenos y malos que tiene el sistema y del cual aprende, que ha ido recabando el sistema de correo de la interacción permanente con sus usuarios en todo el mundo. Mientras que P, es la predicción que hace sobre un determinado correo de que es spam o no, también se conoce como el comportamiento del sistema, que depende de que tan bien hace la predicción y se le asigna una precisión que se usa mucho en problemas de clasificación. Para esto se usa la Matriz de confusión, figura 6.2.

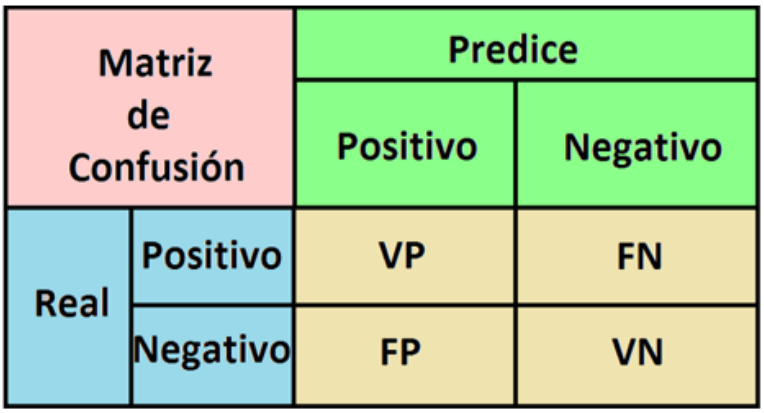


Figura 6.2. Matriz de Confusión, permite encontrar la precisión de un sistema de clasificación de ML.

6.3 Porque el uso de ML

Volvamos a nuestro ejemplo del filtro Spam y digamos que vamos a escribir este programa, el filtro spam usando técnicas de programación tradicional. Siguiendo unos pasos básicos de Ingeniería de Software, primero debemos enterarnos como son los correos no deseados, como es la forma de los encabezados, del tema y del contenido. Aparecen palabras como “gratis”, “regalo”, “tarjeta de crédito”, etcétera.

De manera que con base en toda esta información encuentro estas palabras y algunos patrones que casi siempre se presentan en los correos spam. De manera que entro a programar para que cada una de estas palabras y patrones, permitan que mi programa detecte los correos no deseados.

Puede ser que mi filtro Spam funcione correctamente un determinado tiempo. Esto lo pueden advertir quienes producen los correos Spam y van a buscar otras palabras y otros patrones, para burlar mi filtro. Al hacerlo, mi filtro deja de funcionar un tiempo, hasta cuando me dé cuenta de que está dejando pasar correos spam.

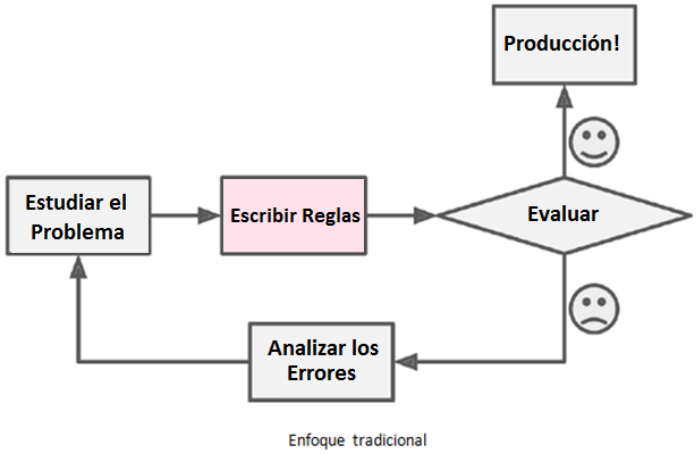


Figura 6.3. Enfoque tradicional de programación.

Ahora las palabras pueden ser “pandemia”, “soledad”, “ayuda”, “anticovid”. Entonces, después de haber logrado esta tarea, con su correspondiente gasto de tiempo, incluyo en mi programa estas nuevas palabras y filtros. Pero nuevamente quienes producen los spam buscan nuevas formas de que sus correos sean aceptados como correos buenos. De manera que después de un análisis, tendría que volver a actualizar mi programa filtro. Figura 6.3, el enfoque de programación tradicional. Llegamos a la conclusión que definitivamente esta no es la mejor manera de mantener un filtro actualizado y funcionando permanentemente.

En contraste, un filtro de spam basado en técnicas ML, aprende automáticamente qué palabras y frases son buenos predictores de spam, detectando patrones inusualmente frecuentes de las palabras en ejemplos spam, comparados con los ejemplos normales.
El programa es mucho más corto, más fácil de mantener y probablemente más preciso.
De manera que un filtro spam basado en ML se dará cuenta automáticamente que “pandemia”, “soledad” se han vuelto inusualmente frecuente en los emails marcados por los usuarios como spam y comenzará a marcarlos sin necesidad de reprogramarlo.

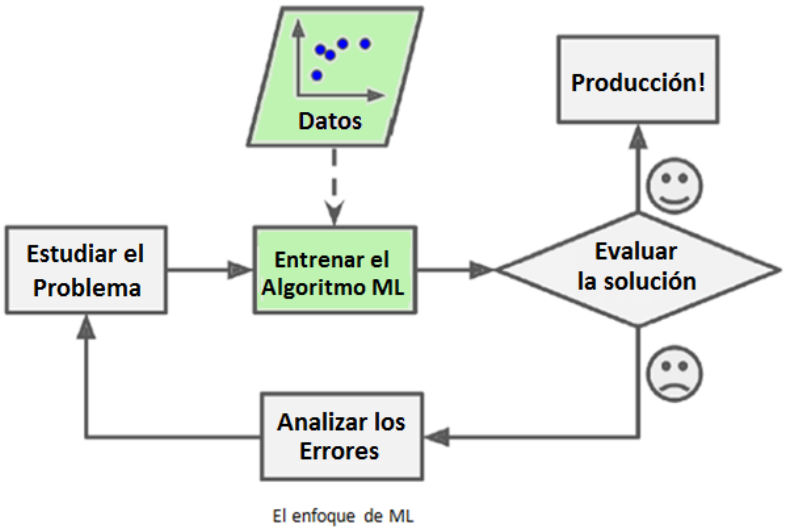


Figura 6.4. Enfoque de programación con ML.

En la figura 6.4, se puede observar cómo es la programación ML. El primer paso, lo mismo que en la programación tradicional se estudia el problema, pero luego se cambia y se entrena un algoritmo de ML, que es realmente un modelo, se evalúa su solución hasta cuando el modelo funcione ben y se lanza a producción. Sin embargo, en la medida en que los spams van cambiando, los usuarios van marcando que correos son no deseados, el modelo va aprendiendo y se va ajustando permanentemente a las nuevas formas de spam. Así, la actualización del filtro es automática.

Además, ML puede ayudar al aprendizaje humano. Los algoritmos de ML se pueden inspeccionar para ver lo que han aprendido, aunque para algunos algoritmos esto pueda ser algo complicado. Así, para el caso del filtro, cuando el modelo ya se ha entrenado con suficientes spams, es fácil inspeccionar y ver las listas de palabras y combinaciones de palabras que pueden ser los mejores predictores de spam.

Aplicar técnicas de ML para cavar en grandes cantidades de datos, puede ayudar a descubrir patrones que no eran inmediatamente evidentes, lo que se conoce como "*minería de datos*". Otra área donde sobresale ML es en problemas que o son muy complejos, para el enfoque tradicional, o simplemente no hay un algoritmo conocido.

En resumen, ML es ideal para:

- Solucionar problemas para los cuales existen soluciones que requieren mucho refinamiento o largas listas de reglas: un algoritmo ML generalmente puede simplificar el código y comportarse mejor que el enfoque tradicional.
- Solucionar problemas complejos: para los cuales se usan fundamentos de enfoques tradicionales que no son una buena solución.
- Ambientes fluctuantes: un sistema ML puede adaptarse a nuevos datos.
- Sacar ideas sobre problemas complejos con gran cantidad de datos.

6.4 Aplicaciones de ML

Son muchas las áreas de aplicación de ML, que dejan abierta a muchos nuevos desarrollos en diferentes campos. Según https://www.javatpoint.com/applications-of-machine-learning, hay 11 áreas, las que se presentan en la figura 6.5. En general la meta del ML es el agrupamiento o la predicción. La predicción se divide en dos categorías: la regresión y la clasificación.

La **regresión** se da cuando la variable que se predice es numérica, por ejemplo predecir el valor de una casa con base en atributos como zona, área del lote, metros cuadrados de construcción y año de construcción.

La **clasificación** se presenta cuando la variable que se va a predecir forma parte de alguna categoría definida como por ejemplo el estrilo de una casa, suponiendo una clasificación de

estilos de casas como: clásica, moderna, colonial, minimalista y rural. Una casa solo puede ser de un estilo.

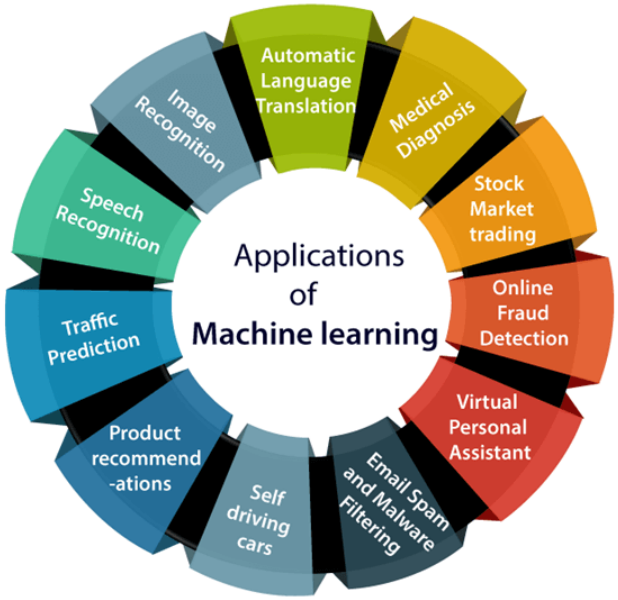


Figura 6.5. Áreas de aplicación de ML.

1. **Reconocimiento de imágenes**. Es una de las aplicaciones mas comunes de ML, por la cantidad de utilidad que tiene en campos que van desde la robótica, la seguridad hasta los sistemas de recomendación por imágenes.
2. **Reconocimiento de voz.** Incluye paso de texto a voz, en robótica para comandos por voz, identificación de voz, análisis de sentimientos, síntesis de voz y otras. Este componente es clave para muchas aplicaciones.
3. **Predicción de tráfico.** Las vías del mundo están cada vez mas congestionadas y los costos en tiempo y dinero de tener personas y mercancías varadas, es muy alto. De manera que cuando se quiere predecir una ruta hay sistemas confiables que lo hacen como “wase”, que funciona en tiempo real, con base en toras tecnologías y empresas que proveen estos servicios satelitales.
4. **Recomendación de productos.** O, Sistemas de Recomendación, se usan mucho en comercio electrónico como en Amazon, AliExpress, Netflix. Una vez uno ha seleccionado un producto las redes sociales lo invaden a uno de información referente al producto.
5. **Vehículos autónomos.** Los carros autónomos ya están en prueba en varios lugares del mundo, lo mismo que muchos robots móviles en empresas industriales. También en aviónica, drones, y para exploración espacial, roberts que en determinadas circunstancias funcionan de manera autónoma.
6. **Filtros spam y para programas malignos** (malware). Como lo vimos en nuestro ejemplo, la llegada de correos mal intencionados, pueden inducir a generar problemas al usuario, lo mismo puede suceder con cualquier otro tipo de información textual, como puede ser la publicidad engañosa. Por otro lado, poder

filtrar los programas engañosos o virus, son de mucha utilidad para el usuario normal.

7. **Asistente personal virtual.** Sistemas que ayudan a la toma de decisiones gerenciales leyendo información directamente de las bases de datos empresariales, haciendo promedios y resúmenes para adecuarlas al ambiente. También en herramientas de negociación.
8. **Detección en línea de fraudes**. Las transacciones en línea son permanentes y se utilizan en todo el mundo, aquí la utilización de ML hace estas transacciones confiables y seguras.
9. **Negocios en mercado de valores.** El uso de ML en mercado de valores esta muy extendido. Aquí la variable tiempo es fundamental, en algunos negocios de Fintech.
10. **Diagnóstico médico.** ML se comienza a utilizar en diagnóstico médico, especialmente apoyados en tomografías. Ayuda en diagnóstico y hallazgo de tumores de seno y cerebrales. Aquí es importante destacar la riqueza que hay en las historias clínicas, que se tiene de todos los usuarios de los servicios médicos, de cuya riqueza todavía no se tiene conciencia. Es posible que ya haya equipos para este tipo de diagnóstico que directamente incluyan hardware especializado con Deep Learning, para diagnóstico rápido.
11. **Traducción automática de lenguajes.** Hoy en día viajar a cualquier parte del mundo puede ser una experiencia más enriquecedora contando con equipos manuales de traducción automática en tiempo real a precios muy cómodos. Esta es una aplicación de ML, muy importante porque permite con la facilidad de comunicación entre los seres humanos.

Todas estas aplicaciones están soportadas por herramientas como RNA, Deep Learning y Redes Convolucionales. Otra herramienta muy utilizada es SVM, que se presenta en el apéndice.

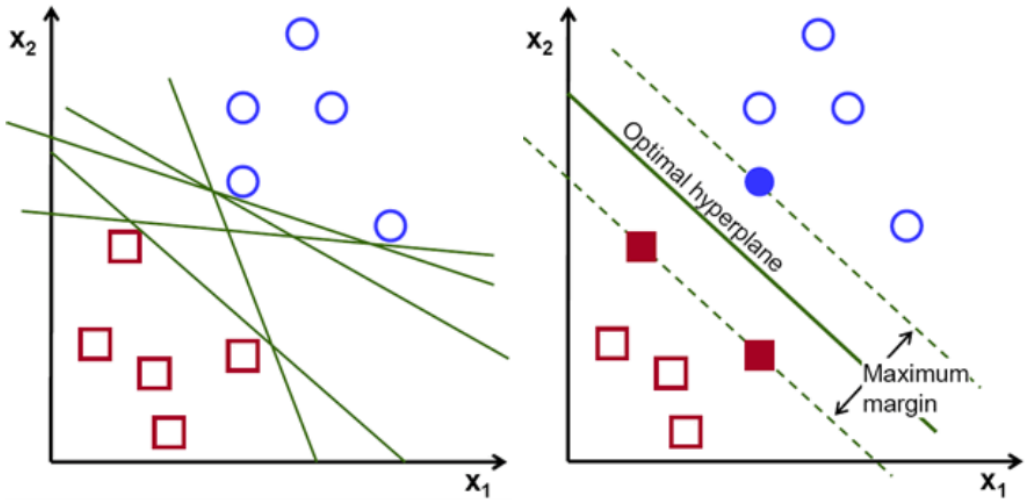
6.5 Algoritmos en ML

A continuación, se presentan algunos algoritmos básicos de ML de los que se considera importante tener cierta claridad sobre su funcionamiento.

6.5.1 VMS.

Una de la herramientas más conocidas y utilizadas en clasificación es el SVM (Support Vector Machine) fue desarrollada en los 90s para clasificación binaria y ahora su utilización se ha extendido para trabajar problemas de clasificación múltiple y regresión. Se utiliza por su precisión y bajas necesidades de poder de cómputo. Es una herramienta básica en ML.

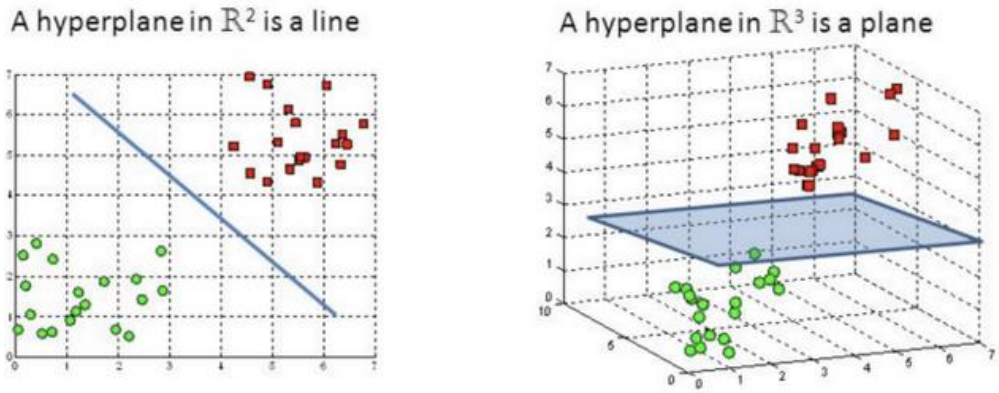
El objetivo del algoritmo VMS es encontrar un hiperplano en un espacio $n - dimensional$, donde n es el número de atributos o características, para que clasifique los puntos de datos.



Posibles hiperplanos

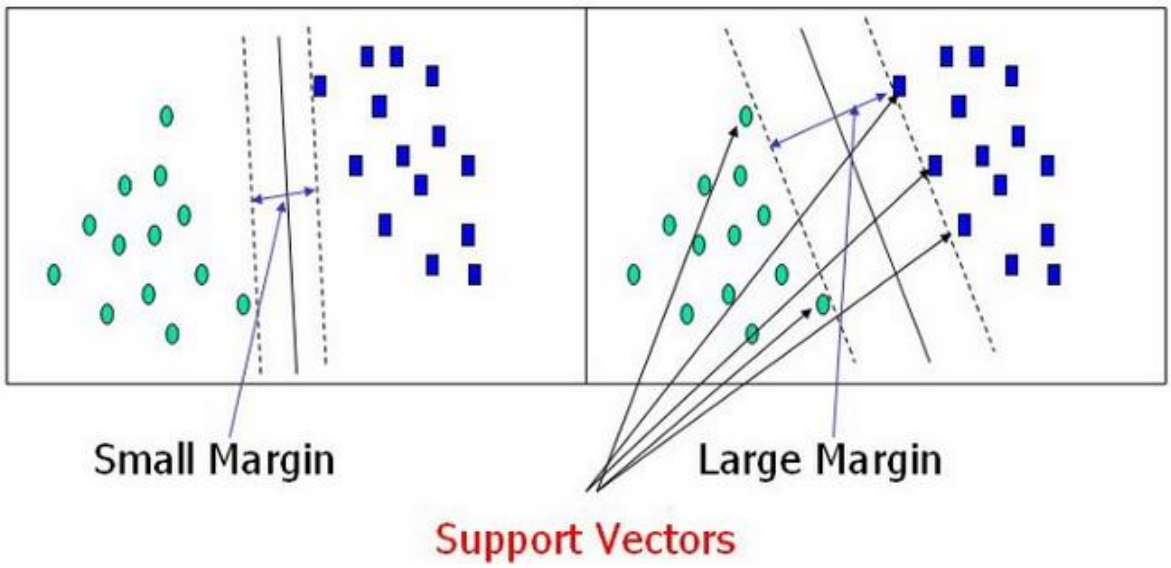
Para separar las dos clases de datos hay muchos posibles hiperplanos. La idea es encontrar un hiperplano que nos dé el máximo margen, es decir la mayor distancia entre los puntos de datos. Al maximizar la distancia del margen nos aseguramos de alguna manera, nos da confianza, para que los futuros puntos de datos queden bien clasificados.

Hiperplanos y vectores de soporte



Hiperplanos en el espacio de características 2D y 3D

Los hiperplanos son límites de decisión que nos ayudan a clasificar los puntos de datos. Los puntos de datos caen ya sea en uno u otro lado del hiperplano dependiendo del número de características. Si el número de características es 2, entonces el hiperplano es solo una recta. Si el número de características es 3, entonces el hiperplano se convierte en un plano bidimensional. Cuando se tienen más de 3 características es difícil imaginarse el hiperplano. Tomado de: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47



Vectores de soporte

Los vectores de soporte son los puntos más cercanos a los puntos de datos más cercanos al hiperplano que influyen en la posición y orientación del hiperplano. Utilizando estos vectores de soporte, maximizamos el margen del clasificador. Cuando se quitan los vectores de soporte se cambia la posición del hiperplano. Estos son los puntos que ayudan a construir la maquina VMS.

6.5.2 Vecinos más cércanos.

En estos algoritmos se busca encontrar las observaciones más cercanas, definiendo distancias, a la observación que queremos predecir. Están los algoritmos **Vecinos K-nearest**, el más común, donde se define el número de vecinos K; y **Radio de vecindad**, donde se define un radio de vecindad.

El algoritmo KNN es el más simple de los algoritmos de ML de aprendizaje supervisado, simplemente calcula la distancia d los nuevos puntos de datos a todos los otros puntos de daos de entrenamiento. Finalmente, asigna al nuevo punto de datos a la clase a la cual pertenecen la mayoría de los puntos de datos K. Se presenta un ejemplo en Notebook de Jupyter, tomado de https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/

6.5.2 SVM.

Vamos a presentar este tema en un Notebook de Jupyter. SVM1

6.6 Tipos de aprendizaje en ML

Ya hemos hablado algo de aprendizaje cuando estudiamos el aprendizaje supervisado para el algoritmo de retropropagación en RNAs. Por otra parte, el tipo de aprendizaje de permite alguna manera una clasificación de algoritmos de ML.

Hay cuatro categorías importantes de aprendizaje: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado y aprendizaje reforzado RL.

6.6.1 Aprendizaje supervisado

En el aprendizaje supervisado el conjunto de entrenamiento con el que se alimenta el algoritmo incluye las soluciones deseadas llamadas rótulos o labels.

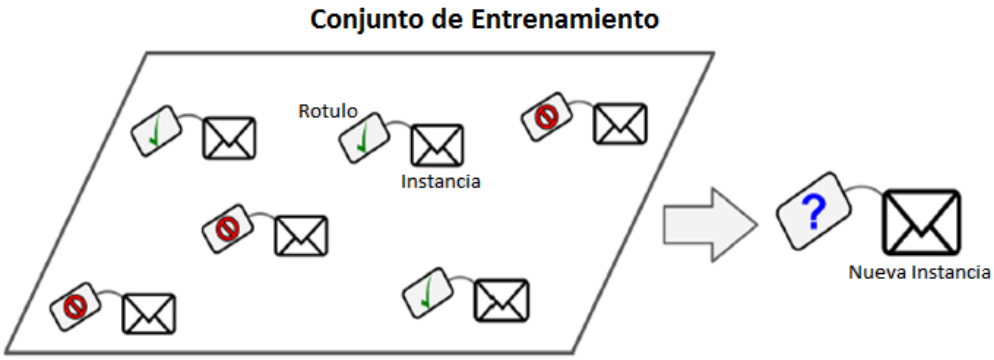


Figura 6.6. Un conjunto de entrenamiento rotulado para clasificación de spam

Los modelos se ajustan a los datos de entrenamiento que comprenden las entradas y salidas. Se usan para hacer perdiciones sobre conjuntos de prueba en los que solo se proveen las entradas; y las salidas del modelo se comparan con las variables rótulo almacenadas, lo que sirve para estimar la bondad del modelo.

Hay aprendizaje supervisado en la clasificación del spam. Este es un buen ejemplo ya que se entrena con muchos emails junto con su clase: spam o normal, así aprende cómo clasificar los nuevos emails. En la figura 6.6, el filtro se ha entrenado con un conjunto de entrenamiento, que incluye correos normales y correos no deseados, cuando llega un correo nuevo, el filtro lo clasifica.

Otra tarea es predecir un valor numérico objetivo, como el precio de un carro. Sí se tiene un conjunto de datos con características como: kilometraje, edad, marca, y otros, llamados predictores, se puede hacer un modelo de regresión para entrenar el sistema.

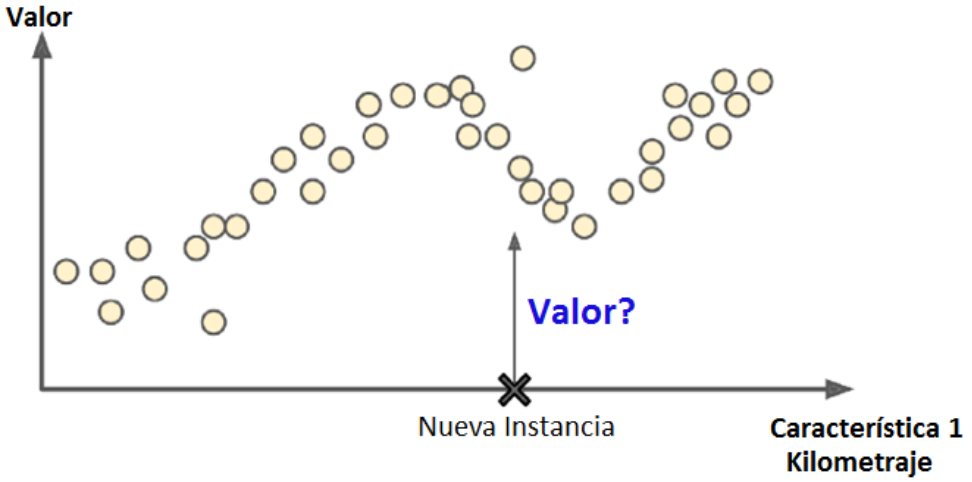


Figura 6.7. Un conjunto de entrenamiento rotulado para regresión.

En la figura 6.7, se tiene un modelo de regresión ajustado con el conjunto de entrenamiento, en este caso con sola una característica (feature), el kilometraje, para predecir el valor de un carro. En general hay varias características de entrada.

6.6.2 Aprendizaje no supervisado

Como se puede adivinar, un aprendizaje no supervisado se realiza con un conjunto de entrenamiento que no está rotulado, el sistema trata de aprender sin un profesor. En la figura 6.8, se puede apreciar un conjunto de datos sin rótulo.

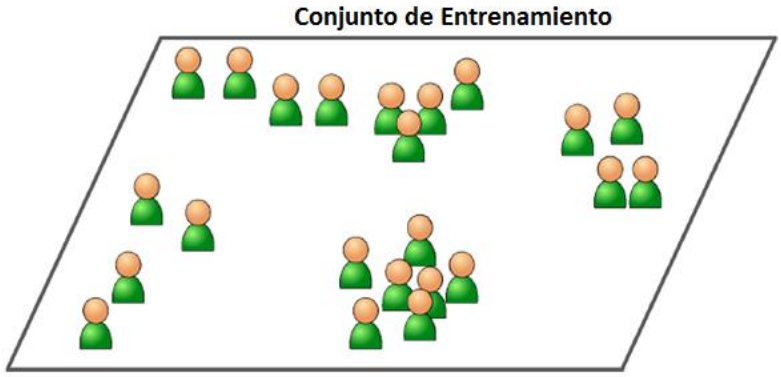


Figura 6.8. Un conjunto de entrenamiento rotulado para aprendizaje no supervisado.

Supongamos que usted tiene una cantidad de datos sobre visitantes a su blog y quiere correr un algoritmo de clustering para tratar de detectar grupos de visitantes similares. En el conjunto de datos no hay rótulos que le digan al algoritmo a qué grupo pertenece un visitante, él encuentra esas conexiones sin ayuda.

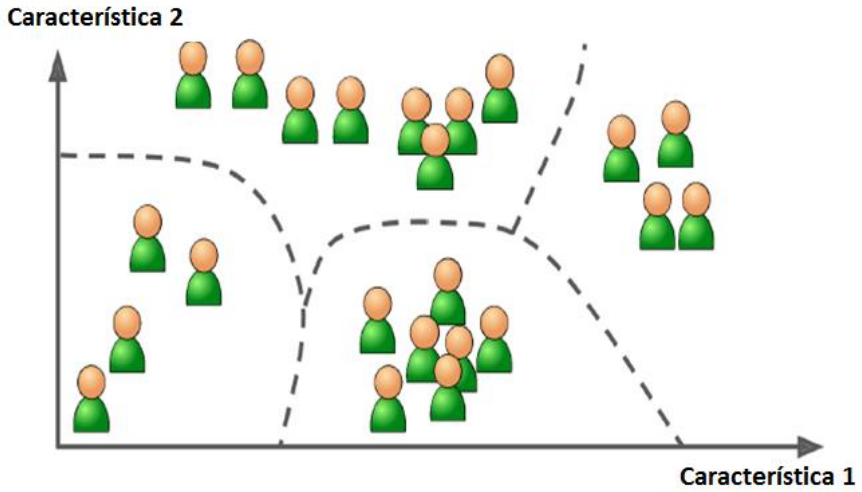


Figura 6.8. Agrupamiento, resultado de entrenamiento con aprendizaje no supervisado.

Por ejemplo, puede observar que en la tarde el 40% de los visitantes al blog son mujeres que aman los libros de cocina; mientras que el 20% son muchachos que les gusta la ciencia ficción, que visitan el blog durante los fines de semana. Este es un ejemplo de agrupamiento.

Los algoritmos de visualización también son buenos ejemplos de aprendizaje no supervisado. Al alimentarlos con datos complejos y sus rótulos, estos algoritmos sacan una representación 2D o 3D de sus datos, que se puede plotear fácilmente. Y tratan de preservar la estructura y la manera en qué se puede comprender cómo están orientados y organizados los datos y quizá encontrar patrones no sospechados.

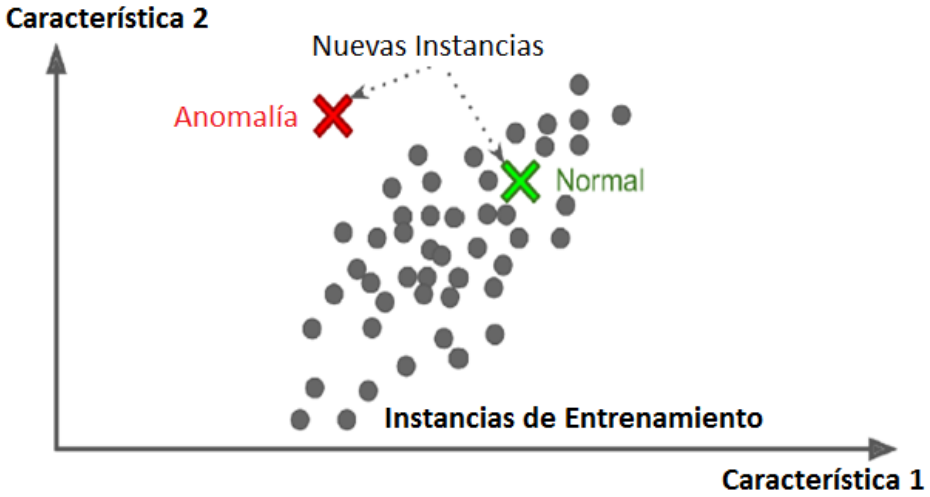


Figura 6.9. Detección de anomalías.

Otra tarea no supervisada es la detección de anomalías, por ejemplo, detectar transacciones inusuales en tarjetas de crédito para prevenir el fraude; encontrar defectos

de fábrica o eliminar valores atípicos en los datasets para alimentar otro algoritmo de ML. En la figura 6.9, se puede observar cómo al llegar una nueva instancia de una transacción comercial, el algoritmo puede detectar si la transacción es valida o se trata de una anomalía, es decir de un fraude.

6.6.3 Aprendizaje semi-supervisado

Es importante debido a que la rotulación de los datos generalmente consume tiempo y es costosa. Además, en muchos casos se tiene una cantidad apreciable de instancias no rotuladas y pocas instancias rotuladas. Para esto hay algoritmos que pueden tratar con datos parcialmente rotulados; esto se llama aprendizaje semi- supervisado.

Algunos servicios de almacenamiento de fotos, como Google fotos son buenos ejemplos de este tipo de aprendizaje. Una vez usted descarga las fotos de su familia allí, este reconoce automáticamente la persona que aparece en diferentes fotografías.

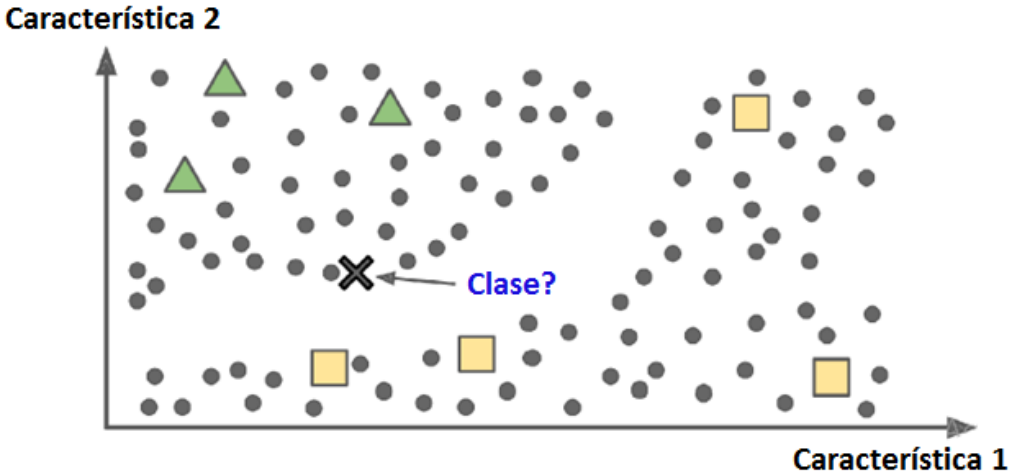


Figura 6.10. Aprendizaje semi-supervisado.

También son varios los proyectos de este tipo que ha desarrollado y está desarrollando FaceBook. En estos momentos hay un concurso abierto Image Similarity Challenge, que se cierra el 21 de octubre de 2021, que tiene disponible un dataset de 1 millon de imagenes y50.000 imagenes refrenciadas, para identificar imagenes manipuladas a escala.

6.6.4 Aprendizaje reforzado

En el aprendizaje reforzado, el sistema de aprendizaje se llama “agente” en este contexto. El agente puede observar el ambiente, seleccionar y ejecutar acciones. Recibe recompensas de retorno si la acción fue exitosa o penalidades si no lo fue. Debe aprender por sí mismo cuál es la mejor estrategia, llamada política, para obtener la mayor recompensa. En el tiempo, una política define qué acción debe escoger el agente cuando está en una situación dada.

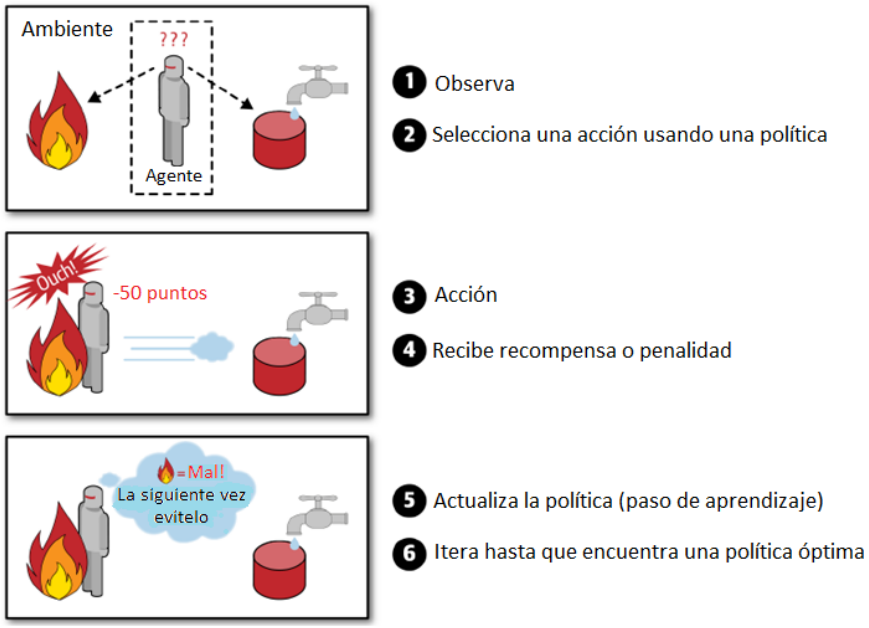


Figura 6.11. Aprendizaje reforzado.

Al principio, el agente no conoce nada sobre el ambiente, por lo que toma las acciones de forma aleatoria. Si una acción trae una recompensa positiva, el agente deberá aprender a escoger esa acción más frecuentemente, mientras que, si una acción trae una recompensa negativa, el agente deberá aprender a escoger esa acción menos frecuentemente. Así, **el agente aprenderá a escoger las acciones que maximicen la suma de recompensas recibidas,** también conocida como el **retorno.**

Ver ejecución del juego Bong, en un Notebook de Jupyter.

6.7 Los datos

En el capitulo 1, hablamos de la utilización de los datos como una estrategia para realizar IA, lo que se puede observar directamente en ML. Para resaltar la importancia de los datos, vamos a hablar sobre la irrazonable efectividad de los datos.

En un artículo famoso publicado en 2001, de M. Banko y E. Brill de Microsoft, mostraron que algoritmos muy diferentes a ML, incluyendo los más simples, se comportan casi idénticamente bien, alrededor de la solución de un problema complejo una vez se tengan los datos suficientes: eliminar la ambigüedad en los lenguajes naturales. La idea de que los datos son más importantes que los algoritmos para solucionar problemas complejos fue popularizada por P. Norving, en el artículo “La irrazonable efectividad de los datos”.

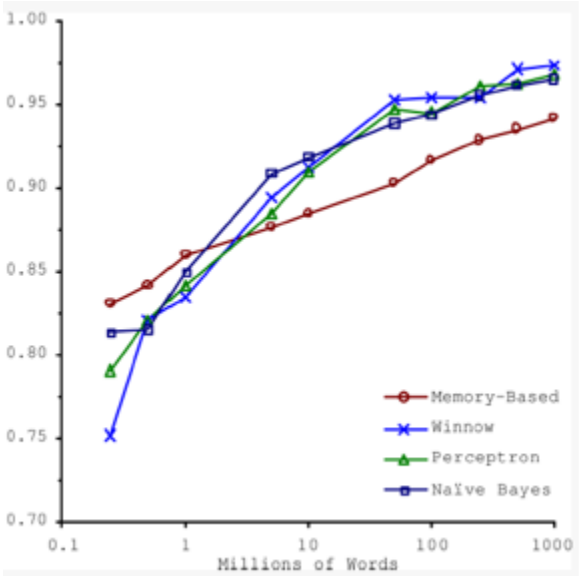


Figura 6.12. La irrazonable efectividad de los datos.

Sin embargo, se debe observar que los datos de entrenamiento de tamaño pequeño y de tamaño medio, son aún bastante comunes; y no es siempre fácil y/o barato obtener datos de entrenamiento. Ni se deben olvidar los algoritmos.

6.7.1 Problemas con los datos

Datos de entrenamiento no representativos

Con el fin de hacer una buena generalización, es crucial que los datos de entrenamiento sean representativos de los nuevos casos que se quieren generalizar. Esto es más difícil de lo que parece: si la muestra es muy pequeña puede ser no representativa, pero aun los grandes muestreos pueden ser no representativos, si el muestreo es defectuoso.

Datos con calidad pobre

Si los datos de entrenamiento están llenos de valores atípicos, errores y ruido, se hará más difícil y poco probable que el sistema detecte patrones subyacentes.

El esfuerzo que se haga para depurar los datos de entrenamiento es bastante provechoso. Recordemos la Ciencia de Datos. Este es el trabajo al que más le dedican tiempo los científicos de datos.

Datos sesgados

Es un problema implícito a muchos datasets, consiste en que desde la fuente de generación de los datos ya existe un sesgo o predisposición. Es el caso que se veía en el libro “Armas Matemáticas de Destrucción”, donde la política, la economía, la educación y la cultura, influyen e el comportamiento de la gente llevando a la producción de datos sesgados.

Es no de los problemas más graves con los que se enfrenta la tecnología ML, el problema sesta en que hasta cuando la herramienta no está en funcionamiento sus problemas no se

pueden detectar, precisamente, porque las personas que los debieran detectar están imbuidas dentro del mismo sistema.

6.8 Lenguajes y librerías

Para el desarrollo de aplicaciones ML, es preciso tener conocimientos de programación, estructuras de datos, manejo de memoria y un lenguaje de programación. Además, hay muchas librerías que se ofrecen para diferentes lenguajes de programación para hacer ML, por lo que es muy fácil para personas con los conocimientos básicos señalados.

Por otra parte, en concepto de otros expertos, no hay un mejor lenguaje para ML, cada uno es bueno en determinado campo. Por ejemplo, la mayoría de los ingenieros de ML prefieren usar Python para problemas de lenguaje natural NLP, mientras que para análisis de sentimientos prefieren usar Python o R, y algunos probablemente van a usar Java, para otras aplicaciones de ML como seguridad y detección de amenazas.

Python

En estos momentos hay más de 8 millones de desarrolladores alrededor del mundo que usan Python. Según Spectrum de la IEEE, es el primer lenguaje utilizado, en este último año.

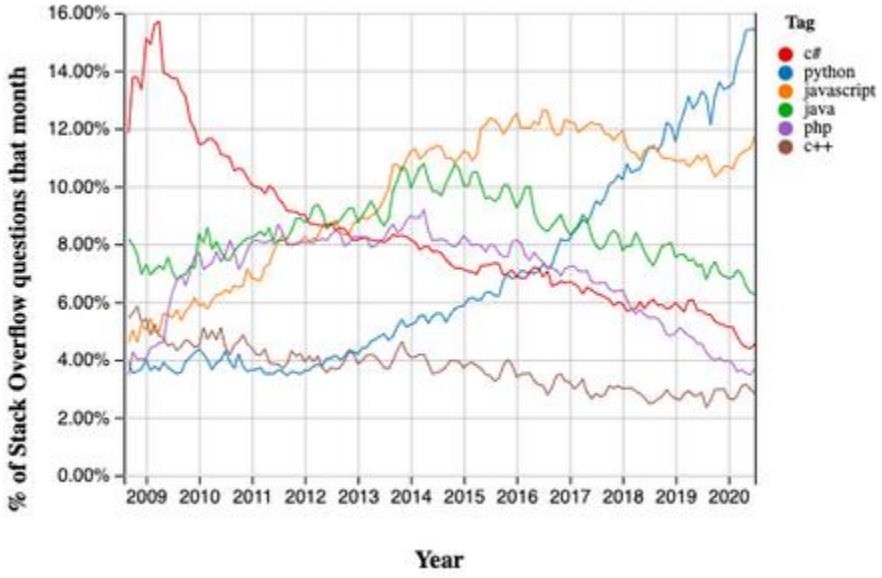


Figura 6.13. Uso de lenguajes de programación. Fuente Stack Overflow

R

Con más de dos millones de usuarios, 12000 paquetes en el repositorio CRAN, aproximadamente 206 grupos Meetup, más de 4000 preguntas sobre programación R cada mes, R es un lenguaje para escribir ML de estadísticos para estadísticos.

6.9. Panorama de ML

Como se puede observar ML es el paradigma de momento, que posiblemente dure bastante tiempo. El hecho de que su fortaleza esta basada en la disponibilidad de conjuntos de datos

bien conformados, llevan a pensar que esta puede ser su debilidad. Como ya se ha visto, los problemas en el dataset pueden llevar a que el modelado con ML sea un completo desastre.

Con respecto a su utilización en robótica, es muy clara cuando hablamos de vehículos autónomos: carros autónomos, drones, vehículos para la exploración espacial y otras máquinas similares.

En ese sentido, para la habilitación de estos dispositivos se requiere el uso de ML en: visión, imitación de aprendizaje, aprendizaje no supervisado, tecnologías medicas y de asistencia a pacientes, aprendizaje multi agentes.

Para el desarrollo de un proyecto de ML, se deben seguir los siguientes pasos:

1. Definir el objetivo del proyecto
2. Definir el criterio de evaluación
3. Evaluar la solución actual
4. Obtener un dataset confiable y con los datos suficientes
5. Seleccionar el modelo
6. Analizar los errores del modelo
7. Aplicar el modelo para el objetivo planteado.

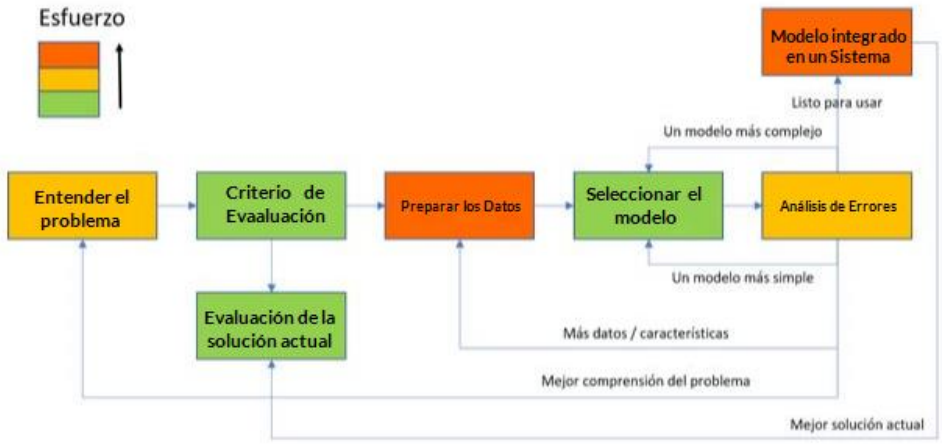


Figura 6.14, modelo para la implementación de sistemas de ML

6.10. ML oscura

Como la base de los algoritmos de ML son los datos, hechos ya pasados, sus resultados son conservadores. Funcionan en el pasado, por lo que, en muchos casos, son incapaces de predecir los cambios culturales y los gustos. Incluso en compañías con mucha experiencia en IA, como Amazon, ha habido problemas.

Según un reporte de Reuters, este gigante del e-comerce tenía un sistema interno que usaba IA para examinar las aplicaciones, que hacen las personas interesadas en trabajar con Amazon. Después de algunas quejas se encontró que una herramienta de reclutamiento de personal tenía perjuicios contra las mujeres degradándolas consistentemente. Este sistema fue descartado por Amazon,

luego de observarse estos comportamientos. Como se ha visto, los sistemas de ML funcionan con datos históricos, en los que se han mantenido estos prejuicios, generalmente llevando a que se perpetúen los prejuicios existentes. En este caso el prejuicio estaba en los ambientes de trabajo dominados por hombres.

Otra manera en que se implementan prejuicios de género y raza se debe a cómo, de manera inconsciente, los desarrolladores hacen las implementaciones con estos sesgos. Este es un problema muy complicado de resolver y la única solución es a través de una muy buena educación, consideración que las empresas de desarrollo en general no son conscientes.

En Colombia, por ejemplo, el ICBF se ha apoyado en el software "BETTO" para tomar decisiones con respeto a la contratación de entidades para el cuidado de niños en condiciones de vulnerabilidad. BETTO, según el gobierno del momento, es una estrategia, una política, y un vehículo de transparencia para la contratación en el ICBF. BETTO incorpora IA, con la capacidad de utilizar algoritmos de control en las plataformas de ciberseguridad, para dar certeza en los procesos de contratación administrativa.

Sin embargo, varias ONG, que habían trabajado durante varios años en esta tarea con el ICBF, no fueron contratadas, debido a que BETTO escogió otras entidades para su contratación. Según la directora del ICBF del momento, fue BETTO quien tomó la decisión, lo que según ella garantiza la idoneidad del proceso. El desconocimiento, la ignorancia y el miedo a enfrentarse a cierto tipo de herramientas hace que las persona le deleguen la responsabilidad de decisiones importantes al "sistema", algo realmente inconcebible pero real.

En su libro, Cathy O´Neal, "Armas Matemáticas de Destrucción", plantea que estamos viviendo en la edad del algoritmo, que las decisiones que afectan nuestras vidas no las toman los humanos sino los modelos matemáticos. En teoría esto lleva a una mayor equidad, pues en principio, todos somos juzgados por las mismas reglas. Pero en la práctica ocurre todo lo contrario, los modelos son opacos, no están regulados, son incontestables. Esto lleva a un refuerzo en la discriminación.

Supongamos que ustedes se gradúan, generalmente van a trabajar en el desarrollo de un proyecto. En poco tiempo ven la necesidad de salir de la casa y adquirir un apartamento pequeño. Van al banco donde le consignan su salario y manifiestan su interés de compra con el fin de conocer todas las condiciones para que le presten para una hipoteca. Y comienzan los problemas, si bien su ingreso es alto, no tiene contrato a término indefinido, no tiene experiencia crediticia y otros. Las reglas de la banca actual no permiten que se le otorgue un préstamo, estas reglas se llevan a los algoritmos, perpetuando la discriminación.

En el libro "THE BOOK OF WHY", j Pearl y D. Mackenzie, Planten el problema de ML en el sentido que llevan a que características (features) altamente correlacionados se asumen como relaciones causa efecto, llevando a veces a resultados completamente desatinados.

6.11. **Ejercicio**

1. Con base en la figura 6.1, tome una ecuación determinada, una razi, un seno, genere muchos valores para usarlo como data set y usando ML encentre el resultado de un determinado valor.

2. Diseñe e implemente un sistema de almacenamiento de datos, para la entrada de productos para una pequeña bodega, utilizando la voz. Ver el siguiente link: https://cloud.google.com/speech-to-text/pricing.
3. Desarrolle un problema de su escogencia.

6.11 Bibliografía

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Concepts, Tools, and Techniques to Build Intelligent Systems, **Aurélien Géron.**

Machine Learning in Robotics, 5 Modern Applications, Daniel Fegguella, https://emerj.com/ai-sector-overviews/machine-learning-in-robotics/

Cathy O´Neal, Armas Matemáticas de Destrucción, 1972

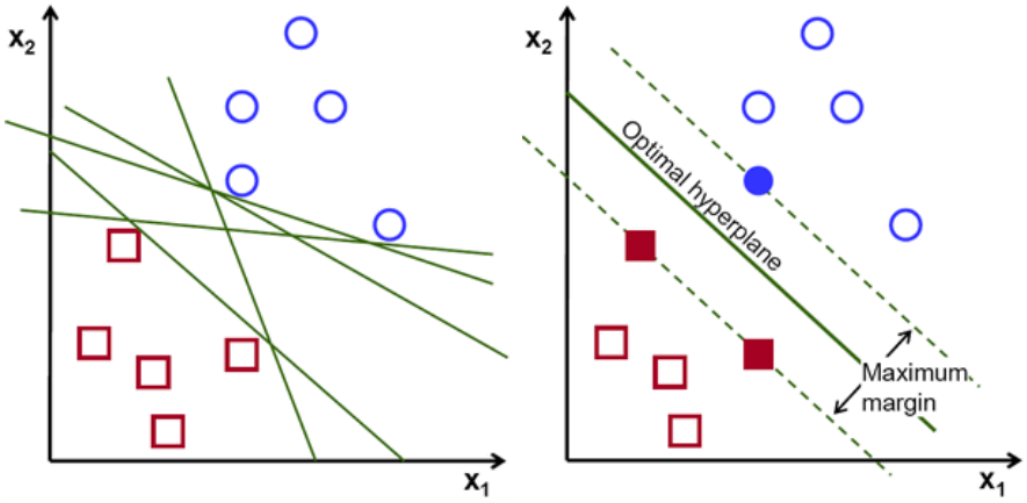
J. Pearl y D. Mackenzie, THE BOOK OF WHY, The New Science of Cause and Effect, Basic Books, 2018.

APENDICE

SVM (Support Vector Machine)

Una de la herramientas más conocidas y utilizadas en clasificación es el SVM (Support Vector Machine), fue desarrollada en los 90s para clasificación binaria y ahora su utilización se ha extendido para trabajar problemas de clasificación múltiple y regresión. Se utiliza por su precisión y bajas necesidades de poder de cómputo. Es una herramienta básica en ML.

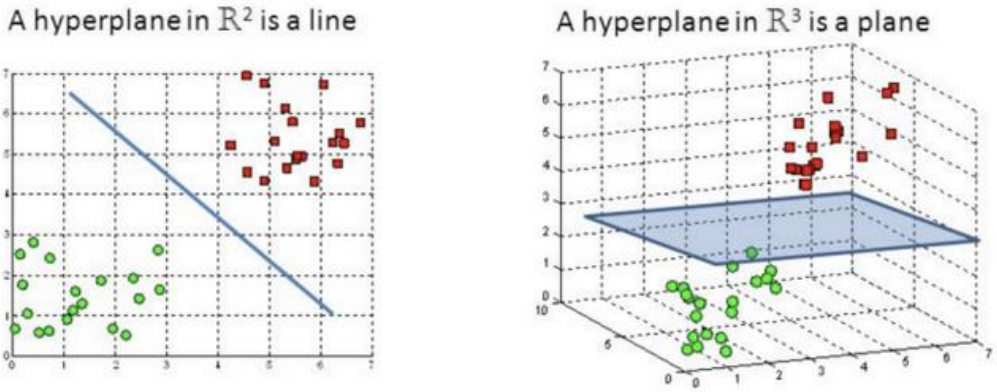
El objetivo del algoritmo VMS es encontrar un hiperplano en un espacio $n - dimensional$, donde n es el número de atributos o características (features) , que clasifique los puntos de datos.



Posibles hiperplanos

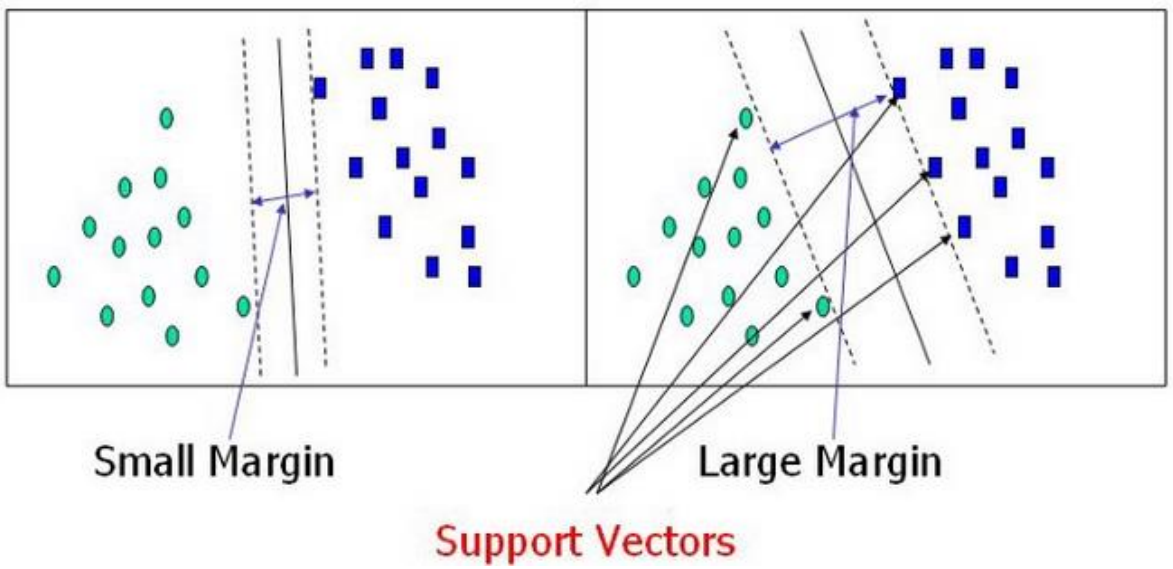
Para separar las dos clases de datos hay muchos posibles hiperplanos. La idea es encontrar un hiperplano que nos dé el máximo margen, es decir la mayor distancia entre los puntos de datos. Al maximizar la distancia del margen nos aseguramos de alguna manera, nos da confianza para que la llegada de futuros puntos de datos queden bien clasificados.

Hiperplanos y vectores de soporte



Hiperplanos en el espacio de características 2D y 3D

Los hiperplanos son límites de decisión que nos ayudan a clasificar los puntos de datos. Los puntos de datos caen ya sea en uno u otro lado del hiperplano dependiendo del número de características. Si el número de características es 2, entonces el hiperplano es solo una recta. Si el número de características es 3, entonces el hiperplano se convierte en un plano bidimensional. Cuando se tiene más de 3 características es difícil imaginarse el hiperplano.



Vectores de soporte

Los vectores de soporte son los puntos mas cercanos son los puntos de datos más cercanos al hiperplano que influyen en la posición y orientación del hiperplano. Utilizando estos vectores de soporte, maximizamos el margen del clasificador. Cuando se quitan los vectores de soporte se cambia la posición del hiperplano. Estos son los puntos que ayudan a construir la maquina VMS.

Intuición de un margen grande

En regresión logística, tomamos la salida de la función lineal y la llevamos a un valor entre usando

https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47