

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

Técnicas de aprendizaje automático.

El aprendizaje automático, o "*Machine Learning*" es una rama de la inteligencia artificial que permite que una máquina aprender sin ser específicamente programada para ello. Esta técnica matemática es indispensable para hacer sistemas con la capacidad de identificar patrones entre los datos, y así lograr desarrollar predicciones mediante distintos algoritmos. Los tipos de Machine Learning se pueden clasificar en 3 categorías, siendo aprendizaje supervisado, aprendizaje no supervisado, y aprendizaje por refuerzo.

1. Aprendizaje supervisado

1.1. Definición

En el aprendizaje supervisado el conjunto de datos de entrenamiento, el cual se utiliza para entrenar a el algoritmo contiene las soluciones o etiquetas (*Labeled Data*). Se buscan relaciones entre ellos según las variables de entrada (*Input Data*) y las etiquetas. Los algoritmos de aprendizaje supervisado son entrenados con un histórico de datos o "*Training Data*" para poder aprender a asignar etiquetas adecuadas en las salidas o predicciones del modelo. La tarea mas común a realizar con algoritmos de aprendizaje supervisado es la clasificación, en donde las etiquetas representan las categorías, es decir según las variables de entrada, el algoritmo debe estimar la distribución de probabilidad $p(y|x)$. Otra aplicación de los algoritmos de aprendizaje supervisado es la regresión, es decir lograr predecir un valor numérico.

► Ventajas

- Tenemos mayor flexibilidad de experimentar con diferentes métricas de evaluación, y experimentar con diferentes algoritmos, ya que conocemos el resultado esperado.
- Aunque requiere más esfuerzo que el aprendizaje no supervisado, el proceso también es relativamente más fácil de comprender que el aprendizaje no supervisado.
- El aprendizaje supervisado permite entrenar a los algoritmos de manera en que los desarrolladores mantienen control total.

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

► Desventajas

- Una desventaja es si data de entrenamiento no sea representativa, el algoritmo al tratar de hacer predicciones en nuevos datos no va a poder generalizar bien y por lo tanto hacer predicciones incorrectas.
- No siempre vamos a contar un conjunto de datos etiquetado, por lo que se necesita un etiquetado manual o utilizando algún algoritmo de aprendizaje no supervisado.
- En tareas de clasificación es muy probable encontrar que existe un desequilibrio entre las categorías de el conjunto de datos.
- La maldición de la dimensión (*Efecto Hughes*). Referente a diversos fenómenos de análisis y organización de espacios de múltiples dimensiones.

1.2. Casos de uso:

- **Caso 1.** Algoritmo arboles de decisión. Enseñar a la maquina diferenciar los datos de entrada y de salida de un perro o un gato. Lo primero que se hace es crear unas etiquetas y con la asignación en las etiquetas de perro o gato se crean rango de etiquetado dando un valor específico a perro o gato en la etiqueta, por ejemplo, en una tabla, y lo que conseguiremos será que a través del algoritmo que se ha programado este tome decisiones sobre los valores que se introducen en la tabla asignados a perro o gato.
- **Caso 2.** Aprendizaje supervisado, Algoritmo de regresión lineal básico para predecir la presión arterial de una persona, según su edad.
- **Caso 3.** Ejemplo en donde tenemos una característica que representa el tamaño de una casa, y otra que representa el precio, tal y como se muestra en las figuras 1 y 2.
- **Caso 4.** En las tarjetas de crédito, un determinado balance x tiene un “x” porcentaje de ser pagada o y porcentaje de no ser pagada. Es decir, es una probabilidad de pagar o no pagar y permite hacer una clasificación por número de tarjeta y documento quien paga y quien no paga.
- **Caso 5 - K-NN** Para clasificar a que raza pertenece una imagen de perro, a modo de experimento se aplico el algoritmo de KNN sobre los valores que se obtienen de una arquitectura de red neuronal pre-entrenada llamada Inception de Google la salida seria una matriz de valores, cada

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

```
In [ ]: de sklearn.pipeline import make_pipeline
de sklearn.preprocessing import PolynomialFeatures
de sklearn.linear_model import LinearRegression
de sklearn.model_selection import cross_val_score

# Función Verdad del terreno
ground_truth = lambda X: np.cos(15 + np.pi * X)

# Generar observaciones aleatorias alrededor de la función verdad del terreno
n_samples = 15
degrees = [1, 4, 30]

X = np.linspace(-1, 1, n_samples)
y = ground_truth(X) + np.random.randn(n_samples) * 0.1

plt.figure(figsize=(14, 5))

models = {}

# Trazar todos modelos del algoritmo de aprendizaje de máquina
para idx, degree in enumerate(degrees):
    ax = plt.subplot(1, len(degrees), idx + 1)
    plt.setp(ax, xticks=(), yticks=())

    # Definir el modelo
    polynomial_features = PolynomialFeatures(degree=degree)
    model = make_pipeline(polynomial_features, LinearRegression())

    models[degree] = model

    # Entrenar el modelo
    model.fit(X[:, np.newaxis], y)

    # Evaluar el modelo usando validación cruzada
    scores = cross_val_score(model, X[:, np.newaxis], y)

    X_test = X
    plt.plot(X_test, model.predict(X_test[:, np.newaxis]), label="Model")
    plt.scatter(X, y, edgecolor='b', s=20, label="Observations")

    plt.xlabel("X")
    plt.ylabel("Y")
```

Figura 1: Desarrollo caso 3

```
X_test = X
plt.plot(X_test, model.predict(X_test[:, np.newaxis]), label="Model")
plt.scatter(X, y, edgecolor='b', s=20, label="Observations")

plt.xlabel("X")
plt.ylabel("Y")
plt.ylim((-2, 2))

plt.title("Degree {} \n MSE = {:.2e}".format(
    degree, -scores.mean()))

plt.show()
```

Figura 2: Continuación del algoritmo desde X-test = X

uno de estos valores representa una característica que aprendió la red neuronal, si aplicamos K-NN a estos valores podemos usar para clasificar a que raza pertenece una nueva imagen de un perro.

1.3. Métodos y/o Algoritmos:

- **Método 1.** Algoritmo K-NN (*K Nearest Neighbour*). Es un algoritmo de clasificación basado en criterios de vecindad. Basado en la idea de que, los nuevos ejemplos se clasificarán con la misma clase de los vecinos mas parecidos, estimados en el conjunto de datos de entrenamiento. Sirve para clasificar nuevas muestras buscando los puntos de datos mas similares o para predecir haciendo conjeturas de nuevos puntos. Muy utilizado en sistemas de recomendación, búsqueda

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	



Figura 3: KNN para clasificar imágenes.

semántica, detección de anomalías, etc.

El algoritmo funciona mediante el siguiente proceso:

1. Se calcula la distancia entre el ítem a clasificar, y el resto de ítems del conjunto de datos de entrenamiento.
2. Se seleccionan los K elementos mas cercanos. KNN trabaja buscando distancias entre una consulta y todos los ejemplos en los datos de entrenamiento.
3. Se realiza una votación entre todos los puntos para decidir la clasificación. Este es el motivo por el cuál, generalmente, se acostumbra tomar un número impar de vecinos para evitar posibles empates.

En este algoritmo no es necesario construir un modelo, ajustar muchos parámetros o hacer suposiciones adicionales.

Si $K_i(x)$ $i = 1, 2, \dots, j$ es el número de prototipos de clase w_i que se encuentran en una vecindad de X , el valor de $\hat{p}(X|w_i)$ puede calcularse como la proporción de los N_i prototipos de la clase que se encuentran en esa vecindad con la ecuación $\hat{p} = \frac{K_i(X)}{N_i}$.

Si consideramos una región muy poblada, la vecindad a considerar tendrá un radio menor. La región poco poblada por lo tanto tendrá un radio mayor. De igual manera se debe modificar la ecuación anterior de manera que se pondere inversamente con el área de la región considerada.

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

Por lo tanto se obtiene que $\hat{p} = \frac{K_i(X)}{N_i V(X)}$.

La elección de k está determinada por la densidad de los puntos y debería hacerse en relación a N_i . Si el estimador es insesgado y consistente se verifica que $\lim_{N_i \rightarrow \infty} (k(N_i)) = \infty$ y $\lim_{N_i \rightarrow \infty} \left(\frac{k(N_i)}{N_i}\right) = \infty$.

- **Método 2.** Regresión Logística. Generalizando una regresión lineal a un problema de clasificación, debemos obtener la probabilidad de pertenecer a una clase, por lo que hay que obtener un valor entre 0 y uno para esto aplicamos la función sigmoide $p(y = 1|x; \theta) = \sigma(\theta^T x)$, donde $\sigma = \frac{1}{1+e^{-x}}$. La función sigmoide nos ayuda a comprimir el valor de salida en el intervalo 0 y 1 e interpretarlo como un valor de probabilidad.
- **Método 3.** Support Vector Machines(SVM). Puede ser utilizado tanto para tareas de clasificación como de regresión, el objetivo de este algoritmo es encontrar un plano en un espacio de N dimensiones en el que se puedan separar de forma lineal los datos de entrenamiento, su objetivo es maximizar el mayor margen de distancia entre los datos. Aunque funciona bien en la mayoría de casos no todos los datos van a ser linealmente separables para esto podemos agregar variables polinomiales. Una pieza clave de este algoritmo y lo que lo hace uno de los mejores es el *kernel trick*, básicamente nos ayuda a calcular los vectores que separan linealmente a la data en un espacio dimensional grande utilizando un hiperplano.

2. Aprendizaje no supervisado

En el aprendizaje no supervisado, el conjunto de datos de entrenamiento de entrada no tienen etiquetas, en este caso, el sistema trata de aprender sin etiquetar ningún dato, por lo cual no existen datos de salida que correspondan específicamente a una entrada determinada, por lo cual solamente podemos describir la estructura de la entrada. Los algoritmos de aprendizaje no supervisado no se pueden aplicar de forma directa a problemas de clasificación o regresión dado que no se tiene conocimiento de cuáles pueden ser los valores de los datos de salida. Los algoritmos de aprendizaje no supervisado permiten realizar procesos más complejos que los algoritmos de aprendizaje supervisado.

- **Ventajas**

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

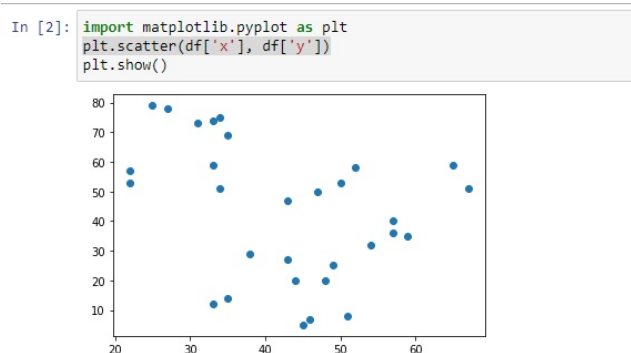


Figura 4: Algoritmo K-means, visualización de los datos.

- El aprendizaje no supervisado encuentra todo tipo de patrones desconocidos en los datos.
- Los métodos no supervisados pueden encontrar características útiles para categorizar.
- Es más fácil encontrar datos no etiquetados que etiquetados o etiquetarlos, y a la vez, más rápido trabajar con datos sin etiquetar.

► Desventajas

- No se puede obtener información precisa con respecto a la clasificación de los datos.
- Las clases espectrales no siempre corresponden a las clases informativas.
- La menor precisión de los resultados es dado a que los datos de entrada no son conocidos y no están etiquetados inicialmente, por lo que la máquina debe hacer este proceso automáticamente.
- La maldición de la dimensión (*Efecto Hughes*). Referente a diversos fenómenos de análisis y organización de espacios de múltiples dimensiones.

2.1. Casos de uso:

- **Caso 1 - Algoritmo de KMeans:** No tiene etiquetas, y puede ser un patrón de datos en una base de datos, como los pacientes que han sido atendidos en urgencias en grupos homogéneos o centrados, pero sin un conocimiento previo de los grupos que queremos obtener. En la figura 4 debemos ver estos datos en una gráfica de inspección donde se comienza a presentar la concentración de casos, ubicamos entonces los centroides de esas concentraciones. Los centroides se definen como clusters, de esta forma gráfico estos e imprimo de nuevo la gráfica pero con los

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

centroides, observando donde se ve la mayor concentración de urgencias. En la figura 5 ya estan

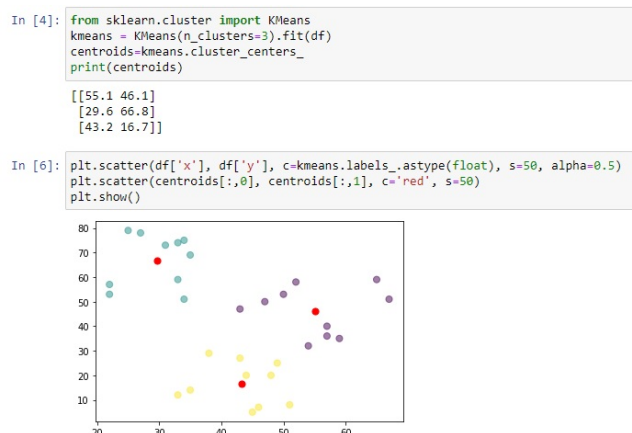


Figura 5: Algoritmo K-means, clusters generados por el algoritmo.

establecidos los centroides, posterior a ellos podremos definir las desviaciones de dispersión que se puedan estar presentando en pacientes de urgencia, o aplicado a bacterias, poblaciones socio económicas, desarrollo de nuevos productos, etc.

- **Caso 2.** Algoritmo de agrupamiento jerárquico. Un ejemplo de agrupamiento de 2 variables es un equipo de Football dónde se enumera cada jugador para hacer más fácil la visualización. Este algoritmo busca los dos puntos con la distancia más corta, los más cercanos, y los agrupa. Luego busca otros dos puntos con la menor distancia y pregunta: ¿la distancia entre estos dos nuevos puntos es menor que la distancia de estos puntos al grupo creado antes? Si la respuesta es sí los agrupa, sino agrupa el punto más cercano al primer grupo creado. Un ejemplo gráfico se muestra en la figura 6
- **Caso 3.** Visualización, los algoritmos de aprendizaje no supervisado se pueden utilizar para entender data compleja que no este etiquetada por ejemplo obtener una representación en 2 o 3 dimensiones.
- **Caso 4.** Al contrario de la estrategia seguida por k-Means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) no presupone clusters convexos, sino que se basa en la densidad de las muestras para identificar los clusters. Por este motivo, los clusters identificados por DBSCAN pueden ser de cualquier forma. El concepto en el que se basa DBSCAN es el de core samples, o muestras base, que son muestras situadas en áreas de alta densidad. Junto a éstas

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

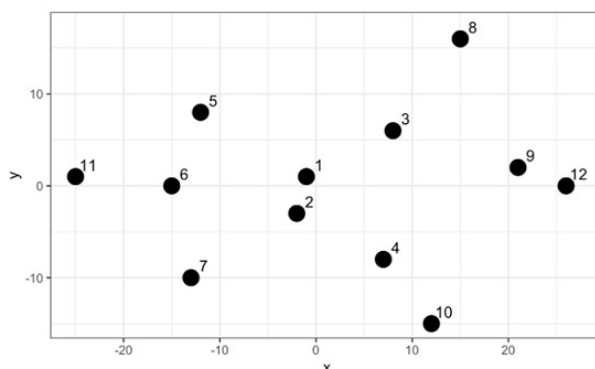


Figura 6: Imagen representativa del ejemplo del caso 2 de aprendizaje no supervisado.

encontramos también las non-core samples, o muestras no-base, que se encuentran próximas a un core sample (sin ser una de ellas). Por este motivo, un cluster va a ser una agrupación de core samples y de non-core-samples situadas a una distancia máxima de alguna core-sample (distancia medida según algún criterio). Dos parámetros importantes: eps, máxima distancia entre dos muestras para poder ser consideradas pertenecientes al mismo "vecindario", y min-samples, número de muestras en un vecindario para que un punto pueda ser considerado core sample. Scikit-Learn implementa este algoritmo en la clase `sklearn.cluster.DBSCAN`. Imagen representativa de este caso en la figura 7.

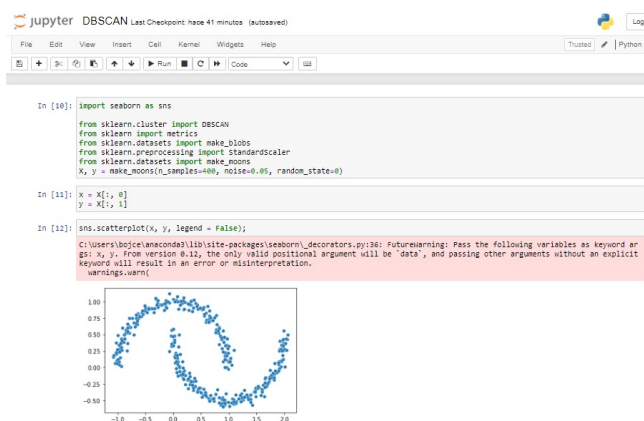


Figura 7: Imagen representativa del ejemplo del caso 5 de aprendizaje no supervisado.

- **Caso 5.** Word2Vec, el objetivo es convertir palabras a vectores con significado semántico. En este caso se usa una arquitectura de red neuronal llamada autoencoder, no necesitamos etiquetas

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

porque la entrada es igual a la salida, por definición pareciera ser aprendizaje supervisado pero lo que nos interesa del modelo es la representación intermedia que aprende la red neuronal y no la capa de salida.

2.2. Métodos y/o Algoritmos:

- **Método 1.** K-Means (Algoritmo de Clustering). Es el algoritmo de clustering más conocido, por lo tanto, es el más utilizado, dado que es muy simple y eficaz. Este algoritmo representa cada uno de sus clusters por la media ponderada de sus puntos, es decir, por su centroide. Esto tiene la ventaja que muestra un significado gráfico y estadístico inmediato y es por esto que se conoce también como K-medias.

Con un conjunto de objetos $D_n = (x_1, x_2, \dots, x_n)$, para todo i , x_i reales y k , v_1 , los centros de los K clusters. El algoritmo K-means se realiza en 4 pasos, que son los siguientes.

1. Elegir aleatoriamente K objetos que forman así los K clusters iniciales. Para cada cluster k , el valor inicial del centro es x_i , con los x_i únicos de D_n pertenecientes al cluster. $\hat{s} = \operatorname{argmin} ||u_k - x||^2$.
2. Se reasignan los objetos del cluster. Para cada objeto x , el prototipo que se le asigna es el más próximo al objeto según una medida de distancia.
3. Una vez colocados todos los objetos, se recalcula el centroide del cluster K .
4. Se repiten las etapas 2 y 3 hasta que no se hagan más reasignaciones. Aunque el algoritmo termina siempre, no se garantiza el obtener la solución óptima. El algoritmo es muy sensible a la elección aleatoria de los K centros iniciales. Por ésta razón, se utiliza el algoritmo sobre un mismo conjunto de datos numerosas veces para intentar minimizar este efecto, considerando que entre más espaciados sean los centros iniciales, se tiene mejor probabilidad de tener mejores resultados.

- **Método 2.** Descomposición en valores singulares(SVD). La descomposición singular de valores es una técnica de factorización de matrices aplicado en Machine Learning que permite descomponer una matriz A en otras matrices U, S, V . $SVD(A) = U * S * V^t$. Con respecto a las dimensiones se parte de que la matriz A tendrá unas dimensiones de $n * m$, donde n representa las filas y

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

m representa las columnas. La matriz U será una matriz ortogonal que tendrá dimensiones de $n * n$ y la matriz V de $m * m$. Existe una propiedad aplicada a SVD enfocada a sistemas de recomendación, esta consiste en que, reduciendo el número de valores singulares de la matriz S a los primeros K valores, se puede obtener una aproximación de la matriz original A .

► **Método 3.** Análisis de componentes principales(PCA). Este algoritmo aprende la representación de los datos, el objetivo es poder representar los datos en una dimensión menor, incluso si no existe una correlación linear entre las dimensiones, esta representación debe ser la que mejor reconstruya los datos originales. Los pasos para PCA explicados a un alto nivel de abstracción serian:

1. Resta de la media: Debemos centrar la data, es decir la media debe ser 0, esto lo logramos calculando la media y restarla a todos los datos.
2. Estandarización: Hay que remover las unidades en los datos, esto lo logramos dividiendo los datos por la desviación estándar en cada una de las dimensiones.
3. Descomposición-Eigen de los matriz de covarianza: se calcula la matriz de covarianza y los eigenvalues correspondientes a los eigenvectors, el vector de mayor longitud define el sub-espacio principal.
4. Proyección: Debemos proyectar cualquier punto en el sub-espacio principal.

3. Conclusiones

- En ambos tipos de aprendizaje uno de los componentes principales si no el mas importante es tener un buen conjunto de datos, ya que si existe un desequilibrio en los datos, existirá un sesgo al tratar de hacer predicciones con data nunca antes vista, perdiendo el objetivo principal del algoritmo de aprendizaje el cual es poder generalizar bien con nuevos datos.
- Al determinar las redes neuronales como método para solucionar algún problema, debemos tener bien definido cual es el problema y comprenderlo a profundidad, dado que esto facilita la elección de los algoritmos a utilizar y estos dependen a la vez del tipo de aprendizaje.
- Ambos tipos de aprendizaje tienen en común la maldición de la dimensión, mientras mas carac-

Asignatura	Datos del alumno	Fecha
Técnicas Multivariantes	Apellidos: Balsells Orellana	26/04/2021
	Nombres: Jorge Augusto	

terísticas tiene la data se hace mas complejo al algoritmo aprender, si lo vemos desde el punto de vista estadístico, es cuando el numero de posibles combinaciones de las variables de entrada es mayor al numero de ejemplos que tenemos en los datos de entrenamiento.

Referencias

- [1] Cristina García Cambroner and Irene Gómez Moreno. Algoritmos de aprendizaje: knn & kmeans. *Inteligencia en Redes de Comunicación, Universidad Carlos III de Madrid*, 23, 2006.
- [2] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.
- [3] Aaron Courville Ian goodfellow, Yosua Bengio. *Deep Learning*. The MIT Press.
- [4] Cheng soon Ong Marc Peter Deisenroth, A. Aldo Faisal. *Mathematics for Machine Learning*. Cambridge University Press.
- [5] Camilo Antonio Ramírez Morales. Algoritmo svd aplicado a los sistemas de recomendación en el comercio. *Tecnología Investigación y Academia*, 6(1):18–27, 2018.
- [6] Andreas C Müller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. .O'Reilly Media, Inc.", 2016.
- [7] Francisco Javier Gómez Quesada, Miguel Angel Fernández Graciani, María Teresa López Bonal, and María Alonso Díaz-Mata. Aprendizaje con redes neuronales artificiales. *Ensayos: Revista De La Facultad De Educación De Albacete*, (9):169–180, 1994.