Resumen Organizado de los Documentos

0 - ¿Qué son los features, observaciones, moda?

Los datos se estructuran en **variables** y **observaciones**.

- **Variables (Features/Atributos):** Se clasifican en **Variables Independientes** (entradas) y **Variables Dependientes** (salidas o categorías).
    - **Tipos de Variables:** Pueden ser **Cualitativas** (Texto, Nominales, Ordinales) o **Cuantitativas** (Discreta, Continua). En el contexto de PCA, las variables se refieren a las dimensiones (como "árboles", "RNA", "PLN").
- **Observaciones (Instancias/Ejemplos):** Representan los casos o puntos de datos sobre los que se realizan las mediciones. En el contexto de los ejemplos, se identifican como "Puntos A 1, A 2, A 3, etc.".
- **Moda:** La moda no es un concepto que se defina o discuta explícitamente en los extractos proporcionados.

1 - Falacias con los datos y Visualización de datos

Falacias Comunes con los Datos

Existen diversos errores de interpretación que pueden surgir al analizar datos:

- **Paradoja de Simpson:** Un resultado o tendencia observada al agregar datos se revierte cuando los datos se analizan en subgrupos. Por ejemplo, un tratamiento puede parecer mejor en general, pero al segmentar por complejidad (ej., tamaño de una piedra renal), el otro tratamiento resulta superior en ambos subgrupos. Esto sucede porque a menudo los casos más complejos se asignan a un método, sesgando los resultados generales.
- **A/B Testing:** Es una prueba que se utiliza para saber si un cambio (ej., el color de un botón de donación) fue la causa de una variación en los resultados. Se divide el tráfico en dos grupos, A y B, para comparar si la variable modificada realmente produce el efecto observado.
- **Sesgo de Supervivencia:** Ocurre cuando se extraen conclusiones solo a partir de los casos o datos que "sobrevivieron" a un proceso. El ejemplo clásico es el de los aviones que regresaron de combate: se debe reforzar las áreas **sin daños** en los aviones que volvieron, ya que los aviones dañados en esas áreas críticas no sobrevivieron.

Visualización de Datos

La visualización es fundamental para entender de forma eficiente los datos y comunicar los resultados de manera clara.

- **Importancia:** El análisis estadístico no siempre revela toda la información importante. El ejemplo del **Datasaurus** muestra que distintos conjuntos de datos

pueden tener la misma media y desviación estándar, pero sus gráficos son radicalmente diferentes.

- **Usos en Machine Learning (ML):**
    - **Análisis inicial:** Examinar si los datos cumplen los supuestos requeridos por el método y detectar complicaciones inesperadas, como valores atípicos o no linealidad.
    - **Evaluar el ajuste del modelo:** Comparar lo predicho contra lo observado, y analizar los residuos.
- **Gráficos Comunes:**
    - **Box Plot (Diagrama de Caja):** Muestra visualmente la distribución, la asimetría, y resume las estadísticas en cinco cantidades: mínimo/máximo, cuartiles (Q1, Q3) y mediana (Q2). También permite identificar *outliers*.
    - **Histograma:** Utilizado para la distribución continua, requiere elegir un número apropiado de *bins* (contenedores).
    - **Scatter Plot (Diagrama de Dispersión):** Muestra la relación entre dos variables y es útil para visualizar si están correlacionadas linealmente.
    - **Violin Plots:** Combinan un diagrama de caja con un diagrama de densidad kernel rotado, mostrando la densidad de probabilidad en diferentes valores.

2 - Introducción a la ciencia de datos 01, Clasificación con SGD, Metricas

Introducción a la Ciencia de Datos

Los tipos de problemas se definen por la variable dependiente:

- **Clasificación:** Si la variable dependiente es cualitativa (categoría).
- **Regresión:** Si la variable dependiente es cuantitativa (valor continuo, ej. temperatura o valor de propiedad).
- **Agrupamiento (Clustering):** Si no hay variable dependiente.

Correlación y Causalidad: Dos variables están correlacionadas si varían sistemáticamente de igual forma. La **Correlación NO IMPLICA Causalidad**; las correlaciones pueden suceder por una tercera variable o por azar. El coeficiente de **Correlación de Pearson** mide la relación lineal entre dos variables, variando de -1 (negativa perfecta) a 1 (positiva perfecta) y 0 (no existe correlación).

Clasificación con SGD (Stochastic Gradient Descent)

- **Gradient Descent (GD):** Es un mecanismo para encontrar valores óptimos para los parámetros de un modelo (ej., pendiente e intersección en regresión). Se busca el mínimo de la **Función de Pérdida (Loss)** de forma iterativa, avanzando en la dirección opuesta al gradiente (que apunta al máximo crecimiento).
- **Learning Rate (Tasa de Aprendizaje):** Es un parámetro (entre 0 y 1) que define cuánto se debe mover el modelo en la dirección de decrecimiento del error en cada paso.

- **SGD:** Se utiliza cuando el GD es muy lento (Big Data). En lugar de usar todos los ejemplos, **toma un ejemplo al azar** (o un *mini-batch*) y computa la derivada en relación a ese solo valor.
- **Regresión Logística:** Busca una función (curva sigmoide) que separe los puntos en dos conjuntos, asociada a problemas de probabilidad (asignando un valor entre 0 y 1).

Métricas de Evaluación

Las métricas sirven para **comparar y medir el rendimiento** de modelos, especialmente en clasificación.

- **Matriz de Confusión:** Muestra el desempeño del clasificador, resumiendo los resultados en: **Verdaderos Positivos (TP)**, **Verdaderos Negativos (TN)**, **Falsos Positivos (FP)**, y **Falsos Negativos (FN)**. Se busca que la diagonal (aciertos) sea lo más alta posible.
- **Métricas Derivadas:**
    - **Precisión (Precision):** Mide la exactitud de las predicciones positivas: $VP/(VP+FP)$. Es alta si el objetivo es minimizar las pérdidas por predicciones erróneas.
    - **Recall (Exhaustividad):** Mide la capacidad del modelo para encontrar todos los casos positivos: $VP/(VP+FN)$.
    - **F-Score:** Media armónica entre Precision y Recall.
- **Curva ROC (Receiver Operating Characteristic):** Traza la Tasa de Verdaderos Positivos (Recall) versus la Tasa de Falsos Positivos (FPR) para todos los umbrales posibles.
- **Área bajo la Curva ROC:** Mide el rendimiento global. Un clasificador perfecto tiene un área de 1, mientras que uno al azar tiene 0.5.

3 - Análisis de Valores Atípicos, Preprocesamiento y Transformación de Datos

Análisis de Valores Atípicos (Outliers)

Un *outlier* es una **observación que se desvía tanto de las otras** como para sospechar que fue generada por un mecanismo diferente. Su detección es crucial porque influyen en los resultados estadísticos, y en algunos casos (detección de fraudes o patologías) la tarea es precisamente encontrarlos.

- **Outliers Univariados:** Valores atípicos en una sola variable.
    - **Z-Score:** Valores con $|Z| > 3$ son la "regla de oro" para considerarlos atípicos.
    - **Z-Score Modificado (usando MAD):** Se usa cuando la media y la desviación estándar son afectadas por extremos. Valores mayores a 3.5 son *outliers*.
    - **IQR (Rango Intercuartílico):** Los *box-plots* usan +/- 1.5*IQR para outliers moderados y +/- 3*IQR para *outliers* severos.

- **Outliers Multivariados:** Valores atípicos en un espacio n-dimensional que requieren ajustar un modelo.
    - **Distancia de Mahalanobis:** Mide la distancia de un punto a un conjunto de observaciones, considerando la media y la matriz de covarianza.
    - **LOF (Local Outlier Factor):** Método basado en densidad que compara la densidad de un punto con la de sus vecinos.
    - **Isolation Forest:** Algoritmo no supervisado que aísla anomalías mediante particiones recursivas en árboles de decisión. Identifica anomalías como observaciones con **longitudes de ruta promedio cortas**.

Preprocesamiento y Transformación de Datos

Estas son etapas clave del Proceso KDD (Knowledge Discovery in Databases).

- **Limpieza de Datos (Datos Faltantes):** Se debe entender el motivo de la ausencia para elegir la estrategia adecuada.
    - **Clasificación de Rubin (1976): MCAR** (Missing Completely At Random), **MAR** (Missing At Random - faltantes explicados por otras columnas) y **MNAR** (Missing Not At Random - falta de dato con explicación no aleatoria).
    - **Estrategias de Imputación:** Reemplazar datos faltantes con estimaciones estadísticas. Ejemplos incluyen sustitución por Media o Mediana (lo cual puede distorsionar la distribución y deprimir correlaciones), Hot Deck, Regresión, **MICE** (proceso iterativo bajo supuesto MAR), y **KNN** (estimación por semejanza con vecinos).
- **Transformación de Datos (Feature Engineering):** Modifica la forma de los datos para mejorar el rendimiento del modelo.
    - **Normalización:** Escalar *features* numéricas (ej. Min-Max, Z-score) para que atributos de mayor magnitud no dominen.
    - **Discretización (Binning):** Dividir variables continuas en intervalos, utilizando criterios como Igual-Frecuencia, Igual-Ancho o Cuantiles.
    - **Variables Dummies / One Hot Encoding:** Recodificación de variables categóricas en variables *dummies*.

4 - Árboles

Los árboles de decisión son modelos utilizados para la clasificación.

- **ID3 (Iterative Dichotomiser 3):** Genera un árbol de decisión.
    - **Entropía:** Mide la impureza o desorden. La entropía es 0 para una muestra homogénea y máxima incertidumbre es $log_2(n)$.
    - **Ganancia de Información:** Se utiliza para cuantificar qué característica proporciona la máxima información, seleccionando el atributo que produce la mayor reducción de entropía para ser el nodo raíz.
- **Impureza de Gini:** Es una medida alternativa a la Ganancia de Información (y computacionalmente menos costosa, utilizada por Scikit-learn) que mide cuán a menudo un elemento elegido al azar sería etiquetado incorrectamente si se hiciera de manera aleatoria. Se calcula un promedio ponderado de la impureza de Gini en los nodos hoja para determinar la impureza total del nodo raíz.

- **C4.5:** Implementa mejoras sobre ID3, como el manejo de **campos numéricos** (creando umbrales booleanos), **datos faltantes** (omitiéndolos en los cálculos), y la **Poda** (eliminación de nodos si no incrementa el error de clasificación sobre el conjunto de test).
- **Random Forest:** Es un meta-algoritmo que combina muchos estimadores (árboles) para mejorar la predicción.

  En Random Forest: muchos árboles independientes, cada árbol se entrena con bootstrap y con selección aleatoria de features por split; la predicción es promedio (regresión) o voto mayoritario (clasificación).

  - **Bagging (Bootstrap Aggregating):** Consiste en construir múltiples conjuntos de entrenamiento tomando muestras aleatorias con reemplazo.
  - **Attribute Bagging (Random Subspace):** Para cada tabla generada, se escogen solo algunos atributos (columnas) de forma aleatoria, típicamente la raíz cuadrada del número total de atributos.
  - **Clasificación:** Una vez entrenados los K árboles, la clasificación de un nuevo ejemplo se determina por la **votación** de la mayoría de los árboles.

5 - Ensambles (Introducción, AdaBoost, Gradient Boost, XGBoost, Híbridos)

Los ensambles buscan **entrenar varios modelos** para reducir la varianza y mejorar la predicción.

Bagging vs. Boosting

- **Bagging:** Entrena modelos de forma independiente sobre subconjuntos de datos (muestras con reemplazo) y los combina. Disminuye la varianza. Ejemplo: Random Forest.
  - Es una técnica que consiste en construir nuevos conjuntos de entrenamiento usando bootstrap (muestras aleatorias con reemplazo) para entrenar distintos modelos, y luego combinarlos.
- **Boosting:** Busca modelos nuevos para **corregir los errores** de los modelos anteriores. Necesita pesos y no puede hacerse en paralelo. Se hace de forma secuencial, donde cada nuevo modelo intenta corregir los errores cometidos por los modelos anteriores

Características Bagging

● Disminuye la varianza en nuestro modelo final

● Muy efectivo en conjuntos de datos con varianza alta

● Puede reducir el overfitting

● Puede reducir el ruido de los outliers (porque no aparecen en todos los datasets)

● Puede mejorar levemente con el voto ponderado

AdaBoost

- **Mecanismo:** Entrena clasificadores base (a menudo **tocones** o *stumps*). Pondera las instancias mal clasificadas, aumentando su peso relativo para que el siguiente predictor se focalice en corregir esos errores.
- **Amount of Say:** Es la ponderación relativa del tocón en el clasificador final, calculada a partir del Error Total del tocón.
- **Votación:** Los árboles votan de forma ponderada según su *Amount of Say*.

Gradient Boost

- **Mecanismo (Regresión):** Inicia con una predicción inicial (promedio). Construye una cadena de árboles de profundidad fija, donde cada árbol predice el **Residuo** (error) cometido por el estimador anterior.
- **Learning Rate ($epsilon$):** Parámetro entre 0 y 1 que escala la contribución de cada árbol. Se utiliza para manejar el *overfitting*. La estrategia es tomar muchos pasos pequeños ($epsilon$ bajo) para lograr una varianza baja.
- **Predicción Final:** Es la suma del valor inicial más los residuos ponderados de cada árbol.

XGBoost (eXtreme Gradient Boost)

XGBoost fue diseñado para Big Data y maneja mejor el *overfitting* que Gradient Boosting mediante regularizaciones.

En XGBoost: boosting basado en gradiente; cada árbol se ajusta a los residuos (gradiente negativo) del ensemble anterior y usa regularización (L1/L2), shrinkage (learning rate), pruning, etc.

- **Regularización ($lambda$):** Introduce un sesgo (*bias*) para reducir el error por varianza (prevenir *overfitting*). Minimiza la suma del cuadrado de los residuos más una penalización proporcional al cuadrado de la pendiente.
- **Similarity Score:** Se calcula usando la suma de residuos al cuadrado, dividido por la cantidad de residuos más el parámetro de regularización ($lambda$).
- **Ganancia (Gain):** Determina el mejor umbral para dividir el árbol, siendo igual a (Score Izquierda + Score Derecha) - Score Raíz.
- **Poda ($gamma$):** Se utiliza el parámetro *gamma* ($gamma$); si la Ganancia - $gamma$ es negativa, se remueve el nodo.

Ensambles Híbridos

Combinan clasificadores de distinto tipo.

- **Voting (Votación):** Consiste en construir N modelos y tomar la predicción mayoritaria.
- **Stacking:** Entrenar múltiples **modelos base** y un **meta-modelo** que decide qué predicción usar, reemplazando el mecanismo de voto.

- **Cascading:** Los datos se pasan secuencialmente. El *input* de cada modelo son las instancias predichas con **poca certeza** por el modelo anterior. Se usa típicamente para asegurar una alta certeza en la predicción.

6 - K-nearest neighbors (KNN), Support Vector Machines

K-Nearest Neighbors (KNN)

- **Mecanismo:** Método de clasificación. Clasifica una nueva observación asignándole la clase más común entre sus **K vecinos más cercanos**.
- **Hiperparámetros:** El valor K, el tipo de distancia (Euclídea, Manhattan), y si los vecinos estarán ponderados.
- **Sensibilidad:** Es sensible a conjuntos de datos **no balanceados** y a la presencia de **outliers**.

Support Vector Machines (SVM)

Los SVM son clasificadores que buscan el hiperplano que maximice la separación entre clases.

- **Maximal Margin Classifier:** El umbral se coloca en el punto medio entre las observaciones límite, logrando el margen más grande posible. Es extremadamente sensible a los *outliers*.
- **Soft Margin Classifier (SVC):** Permite clasificaciones erróneas dentro de un **margen blando** para lograr una mejor generalización (baja varianza). Las observaciones en los límites y dentro del margen blando se llaman **Support Vectors** (Vectores Soporte). En N dimensiones, el clasificador es un **hiperplano** de N-1 dimensiones.
- **SVM (Clasificación No Lineal):** Para problemas no linealmente separables (ej. XOR), la idea es llevar los datos a una dimensión mayor para que puedan ser separados linealmente en ese nuevo espacio.
- **Kernel Trick:** Las **Funciones Kernel** (ej. Kernel Polinómico, Radial/RBFK) realizan este mapeo implícitamente, calculando la relación entre pares de observaciones como si estuvieran en una dimensión superior, sin transformar realmente el espacio. Esto reduce el tiempo de cómputo.
    - El **Kernel Radial (RBFK)** puede soportar clasificadores en **infinitas dimensiones** y funciona de manera similar a un modelo de Vecinos Más Cercanos ponderado.

7 - 01-Introducción y PCA, 02-MDS y PCoA, 03-t-SNE, 04-ISOMAP

Todos estos son algoritmos de **Reducción de la Dimensionalidad**, utilizados para proyectar datos de un espacio de alta dimensión a uno de menor dimensión (generalmente 2D o 3D).

PCA (Principal Component Analysis)

- **Mecanismo:** Se centra en maximizar la varianza (dispersión). Centra los datos y busca el primer eje (PC1) que maximice la suma de las distancias cuadradas de los puntos proyectados al origen.
- **Componentes:** PC1 es una **combinación lineal** de las variables originales. La pendiente de esta recta se relaciona con los **Loading Scores**. Las distancias cuadradas proyectadas se llaman **autovalores**.
- **Varianza Acumulada:** Los autovalores permiten calcular la proporción de variación total acumulada por cada componente principal. El **Scree Plot** gráfica este porcentaje.

MDS (Multi-Dimensional Scaling)

- **Mecanismo:** Se enfoca en **preservar las distancias originales** entre los puntos en la nueva dimensión. Inicia con una matriz de distancias (ej., Euclídea).
- **Optimización:** Utiliza métodos iterativos como el **Descenso por Gradiente** para minimizar la función de **Stress**, que mide la diferencia entre las distancias originales y las distancias en el nuevo espacio de menor dimensión.
- **Flexibilidad:** MDS es más flexible que PCA ya que soporta varios tipos de distancias (Manhattan, Hamming, Mahalanobis).

t-SNE (t-distributed Stochastic Neighbor Embedding)

- **Mecanismo:** Método estocástico e iterativo desarrollado para la **visualización**. Su principal fortaleza es que **preserva los *clusters***.
- **Similitud:** Calcula la similitud de los puntos usando la distribución **Normal** en el espacio original. El ancho de la curva Normal está dado por el parámetro de **Perplejidad**, que corresponde al número esperado de vecinos.
- **Proyección:** En el nuevo espacio de baja dimensión, utiliza la **distribución t-Student** (la "t" en t-SNE), que permite que los *clusters* aparezcan un poco más dispersos y facilita la visualización.
- **Limitaciones:** Es lento con grandes conjuntos de datos y **no se puede usar para proyectar nuevos puntos**.

ISOMAP

- **Mecanismo:** Asume que los datos residen en una **variedad** (subespacio de menor dimensión). Su objetivo es usar la **Distancia Geodésica** (la ruta más corta siguiendo la superficie de la variedad) en lugar de la distancia Euclídea.
- **Aproximación:** Aproxima la distancia geodésica construyendo un **grafo pesado** (conectando K vecinos más cercanos con distancias euclídeas). Luego usa algoritmos de ruta más corta (Dijkstra o Floyd-Warshall) para construir una matriz de distancias geodésicas.
- **Fase Final:** Aplica **MDS** a la matriz de distancias geodésicas calculada.

8 - 01-Redes Neuronales, 02-Backpropagation, 03 - Redes Neuronales implementación

Redes Neuronales y Backpropagation

- **Perceptrón Simple:** Utiliza la **Función Escalón**. Se entrena ajustando pesos (W) y *bias* (b) basándose en el error y la tasa de crecimiento ($alpha$).
    - **Pro:** Es conceptualmente **muy simple** y fácil de entender.
    - **Su gran limitación:** sólo puede resolver problemas **linealmente separables**.
- **Perceptrón Multicapa (PMC):** Se inicializa con pesos aleatorios. El proceso comienza con el cálculo de las salidas (usando funciones de activación como la sigmoide) y el cálculo del **Error Cuadrático Medio**.
- **Backpropagation:** Algoritmo utilizado para entrenar el PMC. Es una aplicación del Descenso por Gradiente. Utiliza la **Regla de la Cadena** para calcular la derivada del error total con respecto a cada peso, indicando la dirección de máximo decrecimiento del error. Los pesos se actualizan usando este cálculo y el **Learning Rate**.

Implementación y Optimizadores

- **Funciones de Activación:** Para clasificación de múltiples clases se usa la función **Softmax** en la capa de salida.
- **Regularización:** Métodos para evitar el *overfitting* y lograr una mejor generalización.
    - **L1 y L2:** Penalizan el valor de los pesos.
    - **Dropout:** "Apaga" activaciones de neuronas aleatoriamente durante el entrenamiento para evitar codependencias.
    - **Early Stopping:** Detiene el entrenamiento cuando el error del conjunto de validación comienza a aumentar.
- **Optimizadores (Implementaciones de Backpropagation):** Un optimizador es el algoritmo que actualiza los parámetros del modelo a partir de los gradientes de la función de pérdida (determina la regla de actualización y su dinámica). Es una implementación concreta del algoritmo de backpropagation.
    - **SGD (Stochastic Gradient Descent):** El *backpropagation* simple.
    - **Momentum:** Utiliza el gradiente para la aceleración (como una pelota rodando por una colina). NO para la velocidad.
    - **AdaGrad:** Reduce el vector gradiente a lo largo de las dimensiones más empinadas (dividiendo el gradiente por un término proporcional a su pendiente), lo que evita que el descenso se desvíe en N dimensiones. No es ideal para redes profundas.
    - **RMSProp:** Soluciona el problema de AdaGrad al olvidar las pendientes anteriores, acumulando solo gradientes recientes.
    - **Adam (Adaptive moment estimation):** Combina las ideas de Momentum y RMSProp, haciendo seguimiento de las medias de decaimiento exponencial de gradientes pasados y de gradientes cuadrados pasados. Tiende a converger más rápido.
    - **Adamax:** Modificación de Adam.
    - **Nadam:** Es Adam + Nesterov, así que a menudo converge más rápido que Adam.
    - **Nesterov:** Variante de Momentum. en vez de calcular el gradiente del error en el punto actual, lo calcula un poco más adelante.
- **Arquitectura:** El número de neuronas de entrada y salida está determinado por el problema a resolver. Una capa oculta puede ser suficiente para muchos problemas,

pero problemas más complejos requieren redes profundas. El **Learning Rate** es el hiperparámetro más importante.

Bag of Words

Es una bolsa de palabras y no una lista de palabras o un conjunto de palabras por:

- No están ordenadas las palabras de ninguna forma
- Cada palabra puede aparecer más de una vez.