

FACULTAD DE INGENIERÍA

UBA

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL


(TB056)

Guía de Trabajos Prácticos

Versión 1.2

Primer cuatrimestre 2025


Guía 1


1.1  Sin utilizar loops (for/while) convertir a escala de grises la imagen `pikachu_vs_charmander.jpeg` implementando las siguientes técnicas:

(a) $\frac{\min(R,G,B) + \max(R,G,B)}{2}$

(b) $\frac{R+G+B}{3}$

(c) $0.3R + 0.59G + 0.11B$ (utilice el comando `@`)

: Utilice las funciones `imread` e `imshow` (matplotlib).

1.2  Sin utilizar loops (for/while), utilizando indexación edite la imagen `AFALogo.bmp` para

- (a) Cortar las letras dentro del logo.
- (b) Cortar las estrellas y transponerlas.
- (c) Generar una mascara separando el color de fondo del logotipo.
- (d) Cambiar el color de fondo de negro a blanco.
- (e) Espejar la imagen (izquierda a derecha).
- (f) Dibujar una grilla sobre la imagen cada 4 píxeles.
- (g) Agregar la 3era estrella.


1.3 Sea la función de densidad de probabilidad

$$p_{XY}(x, y) = \frac{4}{5} \mathbb{1} \{0 < y < 1 + x^3, 0 < x < 1\}$$

(a) Calcular y graficar en una misma figura el soporte, la esperanza condicional $\mathbb{E}[Y|X = x]$ y la recta de regresión.

(b) Calcular el error bayesiano.

: Las únicas integrales que debe resolver son con respecto a la marginal $p_X(x)$. El resto de los cálculos debe hacerse utilizando propiedades.

1.4  Una conocida cadena de comida rápida desea predecir la ganancia de una sucursal en función de la cantidad de habitantes de la ciudad para decidir si conviene abrirla o no. El archivo `mc.txt` contiene la base de datos a utilizar. La primera columna es la población de la ciudad (de a 10.000 personas) y la segunda es la ganancia (de a \$USD 10.000). Los valores negativos indican pérdidas.

(a) Implemente su propio código, utilizando matrices, para realizar una regresión lineal que minimice el error cuadrático medio. ¿Cuanto vale dicho error?


(b) Visualizar los datos con `scatter` (matplotlib) y superponer la recta de regresión estimada sobre ellos.

(c) Diseñe una grilla de puntos que permita graficar el riesgo empírico (en un gráfico 3d) en función de los parámetros (w y b), utilizando `plot_surface` (matplotlib).

Compruebe que los parámetros encontrados en el inciso (a) son efectivamente los que minimizan el riesgo.

(d) Predecir la ganancia de una ciudad de 35.000 habitantes.


(e) Repetir la regresión lineal utilizando `LinearRegression` (sklearn). Comparar resultados.

1.5  Una inmobiliaria desea automatizar la tarea de tasar terrenos. El archivo `inmobiliaria.txt` contiene la base de datos de casas en Portland, Oregon. La primera columna corresponde con dimensión del terreno (en pies cuadrados), la segunda corresponde a la cantidad de dormitorios y la tercera al precio (en dólares).

(a) Realizar una regresión lineal utilizando gradiente descendente (elegir el *learning rate* con prueba y error). Graficar el *error cuadrático medio* en función de las iteraciones del entrenamiento y comparar con el error asociado a la solución matricial.


(b) Predecir el costo de una propiedad de 1650 pies cuadrados y 3 dormitorios. Comparar resultado con la solución matricial.

(c) Calcular el *learning-rate* óptimo y volver a entrenar el algoritmo usando este valor. Repetir los incisos anteriores para este entrenamiento.

1.6  Una inmobiliaria desea automatizar la tarea de tasar terrenos. El archivo `inmobiliaria.csv` contiene la base de datos de casas en California.


(a) Explorar los datos usando `read_csv` (pandas). Indicar cantidad de muestras, nombre y tipo de dato de cada *feature*.

(b) Indicar la proporción de las variables categóricas representando probabilidades.

: Si no sabe que tipo de variable aleatoria es la categórica, deberá buscar dicha información.

(c) Para analizar las variables numéricas utilice el comando `pairplot` (seaborn). Explique que representan los gráficos.

(d) Utilice el comando `SimpleImputer` (sklearn) para completar los valores faltantes con los más frecuentes.


(e) Utilice el comando `get_dummies` (pandas) para codifique las variables categóricas como *one-hot*. : En caso de no conocer el concepto, buscar información sobre *one-hot encoding*.

(f) Utilice el comando `train_test_split` (sklearn) para definir dos conjuntos con las proporciones 75 % y 25 %. Grafique los histogramas de ambos conjuntos (superpuestos) de la mediana del valor de las propiedades.

(g) Utilice el comando `StandardScaler` (sklearn) para normalizar cada variable exceptuando a la mediana del valor de la propiedad. Utilice el conjunto de entrenamiento para fijar la normalización y aplíquela a ambos conjuntos.

(h) Realizar una regresión lineal para predecir la mediana del valor de la propiedad en función del resto de las variables. Indicar el ECM de entrenamiento y testeo.

1.7 Hallar una solución matricial al problema de regresión lineal sin sesgo y con regularización L2. ¿A qué se aproxima la solución si el algoritmo está muy regularizado (pero no tanto como para pensar que es cero)?

1.8  Se desea estimar la cantidad de agua que fluye por una presa a partir de la variación del nivel de agua. El archivo `represa.csv` contiene los datos a utilizar, definiendo los conjuntos de entrenamiento, validación y testeo.

(a) Visualice los tres datasets a partir de un gráfico `scatter` utilizando colores diferentes para cada conjunto.

(b) Realice una regresión lineal utilizando `LinearRegression` de `sklearn`. Grafique la recta de regresión estimada sobre la el `scatter`.

(c) Realice una regresión polinómica de orden 8 sin regularización. Grafique la función de regresión estimada sobre el `scatter`.

(d) Utilizando `sklearn.linear_model.Ridge`, repetir el inciso anterior regularizando con $\lambda = 1$ y $\lambda = 100$.

(e) Graficar el error cuadrático medio en función del hiperparámetro de regularización $\lambda \geq 0$ para el conjunto de entrenamiento y validación. ¿Qué valor minimiza el error de validación?

(f) Calcular el error cuadrático medio de testeo para el hiperparámetro elegido en el inciso anterior.

Guía 2


2.1 Por un canal de comunicaciones se emiten bits de forma aleatoria, siendo el 75 % de ellos 1. Dependiendo del bit transmitido, la comunicación es afectada por un ruido aditivo normal de media nula y varianzas: 4 si el bit es un 0 y 1 si el bit es un 1. Sea X la señal recibida y Y el bit emitido.

- (a) Hallar y graficar $p_X(x)$.
- (b) Hallar y graficar $P_{Y|X}(1|x)$.
- (c) ¿Para que valores de x ocurre que $P_{Y|X}(1|x) > P_{Y|X}(0|x)$?
- (d) Calcular el error bayesiano expresando el resultado de las integrales en función de $\Phi(\cdot)$ (función de distribución de la normal estándar), para luego computar los cuantiles. Comparar el resultado contra un *clasificador al azar* y contra un *clasificador dummy*.


2.2 Sean p y q dos distribuciones Bernoulli de parámetros $\frac{1}{2}$ y $\frac{1}{3}$ respectivamente. Calcular $\text{KL}(p\|q)$ y $\text{KL}(q\|p)$. Expresar el resultado en nats.

2.3 Sea $p = \sigma(z)$ la función sigmoide.

- (a) Calcular la función inversa $\sigma^{-1}(p)$ con $p \in (0, 1)$.
- (b) Calcular la derivada $\sigma'(z)$. Encontrar sus valores ínfimo y supremo, y (en caso que existan) los puntos donde los alcanza.
- (c) Escribir la derivada en función de p .

2.4  Un profesor desea estimar si un alumno va a aprobar o no la materia en base a la nota de dos parcialitos. El archivo `parcialitos.txt` contiene una base de datos con las notas de cada estudiante en los parcialitos y si, efectivamente, aprobó o no la materia (1 es aprobar).

- (a) Hallar una expresión analítica para la función costo y su gradiente.
- (b) Realizar una regresión logística utilizando `LogisticRegression` (sklearn) y graficar la frontera de decisión sobre un `scatter`.
- (c) Predecir si un estudiante con notas 45 y 85 aprobaría la materia.
- (d) Graficar la curva ROC del clasificador, implementando su propio código. Indicar el punto correspondiente a la decisión tomada en el inciso (b) y el EER.

2.5  El gerente de producción de una fábrica de circuitos integrados desea predecir si un determinado integrado pasará el control de calidad. El archivo `microchips.txt` posee datos de la evaluación de dos pruebas diagnóstico de diferentes integrados, y una tercer columna que indica si pasaron el mencionado control (1 es pasar la inspección).

- (a) Construir un mapa polinómico hasta orden 6 inclusive. ¿Como puede relacionar la cantidad de parámetros con el grado del polinomio y la cantidad de *features*? Encontrar una expresión matemática que vincule esas magnitudes.

- (b) Realizar una regresión logística utilizando `LogisticRegression` (sklearn) y graficar la frontera de decisión sobre un `scatter` sin regularización.
 - (c) Realizar una regresión logística y graficar la frontera de decisión sobre un `scatter` con regularización L2 y $\lambda = 1000$.
 - (d) Realizar una regresión logística y graficar la frontera de decisión sobre un `scatter` con regularización L2 y $\lambda = 1$.
- 🔗: Funciones como `meshgrid` (numpy) y `contour` (matplotlib) pueden ser útiles para graficar las fronteras.

2.6 📖 La base de datos MNIST posee imágenes de los dígitos manuscritos (del 0 al 9). Se desea entrenar un clasificador que, a partir de una imagen, prediga que dígito aparece en ella.

- (a) Cargar la base de datos utilizando `tensorflow.keras.datasets.mnist.load_data`. Utilizando `imshow` (matplotlib) represente 10 muestras del conjunto de testeo elegidas al azar.
- (b) Realizar una regresión logística e indicar el *accuracy* de entrenamiento y testeo.
- (c) Utilizando `ConfusionMatrixDisplay` (sklearn) represente la matriz de confusión normalizada (testeo) para mostrar la probabilidad de cada predicción para cada clase con 3 decimales.

2.7 📖 Se denomina formante a las frecuencias donde se dan los picos de intensidad en el espectro de un sonido. El archivo `formantes.txt` contiene ejemplos de los 3 primeros formantes del sonido de las vocales /a/, /o/ y /u/. Utilizando solamente los dos primeros formantes:

- (a) Graficar las muestras en un `scatter`, representando los formantes de cada vocal con colores distintos.
- (b) Superponer a la gráfica anterior las medias y las covarianzas de cada gaussiana (una curva de nivel) del modelo de LDA.
- (c) Implementar un algoritmo de LDA para clasificar los formantes.
- (d) Graficar la predicción de las muestras y la frontera de decisión.
- (e) Generar 50 muestras sintéticas y graficarlas junto a las fronteras. Representar los formantes de cada vocal con colores distintos.

🔗: Tenga en cuenta que, además de las medias y varianzas, deberá utilizar las probabilidades c_k aprendidas durante el entrenamiento. Funciones como `random.choice` y `random.multivariate_normal` (numpy) pueden ser útiles.

2.8 📖 [ver **Ejercicio 2.5**] La fábrica de circuitos integrados desea predecir si un determinado integrado pasará el control de calidad a partir del archivo `microchips.txt`.

- (a) Graficar la frontera de decisión de un algoritmo 1NN sobre el `scatter` de la base de datos. ¿Que puede decir del error de entrenamiento? ¿Puede extraer una conclusión general al respecto?

(b) Repetir para un 7NN. Relacionar el valor de K con los conceptos de *overfitting* y regularización.

(c) Graficar $\hat{P}(1|x)$ para un algoritmo 1NN y 7NN entrenados solamente con la primera de las pruebas diagnósticas.

🔗: La función `argsort` (numpy) puede ser útil. El algoritmo KNN debe estar implementado con una sola función, de modo que ésta pueda usarse para todo $k \in \mathbb{N}$ y cualquier cantidad de *features*.

2.9 📖 El archivo `ejs_svm.pkl` contiene un par de bases de datos. Utilizando la base de datos *1er Dataset*:

(a) Implementar una clasificación SVM utilizando `solve_qp` (qpsolvers), resolviendo el problema primal. Graficar la frontera de decisión y las rectas de vectores soportes sobre un `scatter`.

(b) Repetir el inciso (a) resolviendo el problema dual.

(c) Repetir el inciso (a) relajando los márgenes (utilizando $C = 1$) y resolviendo el problema primal.

(d) Hallar el problema dual con márgenes relajados. Puede buscarlo en la bibliografía o deducirlo. 🔗: Se recomienda practicar la deducción.

(e) Repetir el inciso (a) relajando los márgenes (utilizando $C = 1$) y resolviendo el problema dual.

2.10 📖 El archivo `ejs_svm.pkl` contiene un par de bases de datos. Utilizando la base de datos *2do Dataset*, implementar una clasificación SVM con Kernel gaussiano ($\gamma = 50$) utilizando `svm.SVC` (sklearn) con $C = 1$. Graficar la frontera de decisión y las curvas de vectores soportes sobre un `scatter`.

2.11 📖 La cromatografía de ultra alta performance acoplada a espectrometría de masas de alta resolución permite el diagnóstico del cáncer de próstata. El archivo `prostata.csv` posee datos de la abundancia de concentración de diferentes compuestos químicos y el resultado del diagnóstico: sano, cáncer, benigno y post-cirugía. Se desea predecir el diagnóstico en función del resto de los indicadores.

(a) Las muestras sin etiquetas representan errores de medición. Construir un conjunto de datos con las muestras válidas.


(b) Utilizando `cost_complexity_pruning_path` (sklearn) y utilizando la entropía como impureza, calcular todos los α relevantes para la poda de un árbol de decisión.

(c) Utilizando `GridSearchCV` (sklearn) optimizar el valor de α para un 4-fold, utilizando como métrica la F_1 macro. Graficar los valores de F_1 cross-validada en función de α . ¿Qué α maximiza dicha métrica?

(d) Utilizando `plot_tree` (sklearn) graficar el árbol podado. Indicar la cantidad de nodos y hojas.

(e) Encontrar los 5 *features* más relevantes según la *Gini importance*.

🔗: Es importante resolver todo el ejercicio utilizando solo 1 `fit`.

2.12  [ver **Ejercicio 2.6**] La base de datos FASHION-MNIST posee la mismas características que la MNIST pero para clasificar 10 tipos de ropa. Se desea entrenar un clasificador que a partir de una imagen prediga que dígito aparece en ella.


(a) Cargar la base de datos utilizando `tensorflow.keras.datasets.fashion_mnist.load_data`. Utilizando `imshow` (matplotlib) represente 10 muestras del conjunto de testeo elegidas al azar. ¿Que tipo de prenda representa cada categoría?

(b) Utilizando `RandomForestClassifier` (sklearn), entrenar un bosque aleatorio de 100 árboles con impureza *Gini*. Indicar el *accuracy* de entrenamiento y testeo.

(c) Utilizando `ConfusionMatrixDisplay` (sklearn) represente la matriz de confusión normalizada (testeo) para mostrar la probabilidad de cada predicción para cada clase con 3 decimales.


(d) Graficar en una imagen los 100 píxeles más relevantes según la *Gini importance*. Indique con un punto negro los mencionados píxeles y deje el resto en blanco.

Guía 3

3.1  [ver **Ejercicio 2.12**] Utilizando la base de datos FASHION-MNIST, se desea entrenar un algoritmo de PCA.

(a) Utilizando `linalg.eig` (numpy), calcular y graficar el porcentaje de energía en función del número de componentes principales.

(b) Graficar el error cuadrático medio de testeo en función del número de componentes principales.


: Mucha atención a no repetir cálculos. Es decir, aprovechar la reconstrucción con k componentes principales para el cálculo de $k + 1$ componentes.

(c) Graficar 10 imágenes reconstruidas de testeo (elegidas al azar) utilizando 1, 100 y 784 componentes principales.

(d) Graficar un `scatter` de las dos primeras componentes principales de las imágenes de testeo, indicando en diferentes colores las clases.

(e) Se desea evaluar el desempeño del algoritmo de PCA como detector de anomalías. Para ello, construir una base de datos combinando el conjunto de datos de testeo con el conjunto de datos de testeo de la base de datos MNIST (dígitos).


(f) Diseñar un detector de anomalías comparando el error cuadrático contra un umbral. Graficar la curva ROC y marcar el *equal error rate* para 1, 100 y 784 componentes principales. Interpretar resultados.

3.2  [ver **Ejercicio 2.7**] Utilizando los dos primeros formantes de la base de datos `formantes.txt`:

(a) Implementar K-means para 3 clusters. Utilizar, como condición de parada, tanto cantidad de iteraciones (100 por ejemplo) como convergencia.

(b) Graficar un `scatter` de la clasificación final de los datos de entrenamiento, resaltando los centroides.

(c) Graficar las fronteras de decisión, superpuestos a un `scatter` con las verdaderas etiquetas.

3.3  Se desea comprimir la imagen `pikachu_vs_charmander.jpeg` a 16 colores, utilizando `cluster.KMeans` (sklearn).

(a) Tomando cada pixel como muestras diferentes, implementar un K-means de 16 clusters.

(b) Utilizar los centroides como diccionario, para convertir cada pixel en un centroide (utilizando el algoritmo previamente entrenado). Utilizar `imshow` (matplotlib) para graficar la imagen ya codificada.

(c) Calcular la cantidad de bits necesarios para guardar la imagen antes y después de comprimirla (teniendo en cuenta el etiquetado y los centroides).

BIBLIOGRAFÍA SUGERIDA

1. “Pattern Recognition and Machine Learning”, C. Bishop.
2. “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, J. Hastie, T. Tibshirani, R. Friedman.
3. “Machine Learning: A Probabilistic Perspective”, K. Murphy.
4. “Introduction to Machine Learning with Python: A Guide for Data Scientists”, A. Müller, S. Guido.
5. “Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference”, C. Davidson-Pilon.
6. “Pattern Classification”, R. Duda, P. Hart, D. Stork.
7. “Deep Learning”, I. Goodfellow, Y. Bengio, A. Courville.
8. “Elements of Information Theory”, T. Cover, J. Thomas.
9. “Elements of Causal Inference: Foundations and Learning Algorithms”, J. Peters, D. Janzing, B. Schölkopf.
10. “Foundations of Machine Learning”, M. Mohri, A. Rostamizadeh, A. Talwal-kar.
11. “Data Analysis: A Bayesian Tutorial”, D. Sivia and J. Skilling.
12. “The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation”, C. Robert.