Aprendizaje Automático – Guía de Ejercicios*

Departamento de Computación – FCEyN
Universidad de Buenos Aires

Segundo cuatrimestre 2018
Versión: 19 de octubre de 2018

1. Herramientas

Ejercicio 1.1. Revisar y completar el notebook `notebook_1_herramientas.ipynb` disponible en la sección de Descargas.

2. Introducción

Ejercicio 2.1. ¿Cuál es la diferencia entre el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzos?

Ejercicio 2.2. ¿Cuál es la diferencia entre un problema de clasificación y uno de regresión?. Determinar para los siguientes problemas de aprendizaje supervisado si se trata de problemas de clasificación o de regresión.

(a) Dado un tweet, determinar si habla en contra o a favor de un candidato presidencial.

(b) Predecir cuánto gastará una empresa en luz el próximo semestre.

(c) Predecir la nota que tendrá un alumno en un examen cuya nota puede ser $0, 1, 2, \ldots, 10$

(d) Predecir la nota que tendrá un alumno en un examen cuya nota puede ser "A", "R" o "I".

(e) Predecir donde vive una persona.

(f) Predecir donde vivirá una persona dentro de 5 años.

(g) Predecir si se gastará más o menos que $50.000 por mes de luz el próximo semestre.

(h) Predecir la probabilidad de que una persona haya comprado un bote el último año.

Tip: Pensar con qué etiquetas se entrena al modelo.
¿Qué diferencia hay entre los items (e) y (f)? ¿Qué relación hay entre el momento de toma de los atributos y la etiqueta?

Ejercicio 2.3. Describir la tarea, medida de performance y experiencia para (a) filtro de spam; (b) dictado de textos; (c) autenticación biométrica (ej: huellas dactilares); (d) detección de fraude en tarjetas de crédito.

Ejercicio 2.4. Sea un problema de clasificación en el cual cada instancia tiene 2 atributos numéricos (coordenadas x e y) y pertenece a una de dos clases posibles (blanco o negro). Para cada uno de los tipos de hipótesis ilustrados en la Figura 1 se pide:

- identificar los parámetros de la hipótesis y describir el espacio de hipótesis H;
- pensar un algoritmo para encontrar una hipótesis y describir su sesgo inductivo.

Ejercicio 2.5. Completar el notebook `notebook_2_titanic.ipynb` disponible en la sección de Descargas. En este ejercicio, deberán descargar y explorar el contenido del archivo `titanic-train.csv` que contiene datos de pasajeros del naufragio del transatlántico Titanic en 1912, incluyendo edad, sexo, clase del pasaje y supervivencia a la tragedia, entre otros. Para una descripción completa, ver `https://www.kaggle.com/c/titanic/data`. Completar el notebook de manera de clasificar a los pasajeros en supervivientes y no supervivientes tan bien como sea posible. No está permitido usar ninguna técnica de Aprendizaje Automático, por ahora. El objetivo es conseguir un buen porcentaje de aciertos sobre estos datos (ver final del notebook).

*Algunos ejercicios fueron adaptados de los libros "Machine Learning", de Tom Mitchell (McGraw-Hill, 1997); "Pattern Recognition and Machine Learning", de Christopher Bishop (Springer, 2006); y "An Introduction to Statistical Learning", de James, Witten, Hastie & Tibshirani (Springer, 2015).

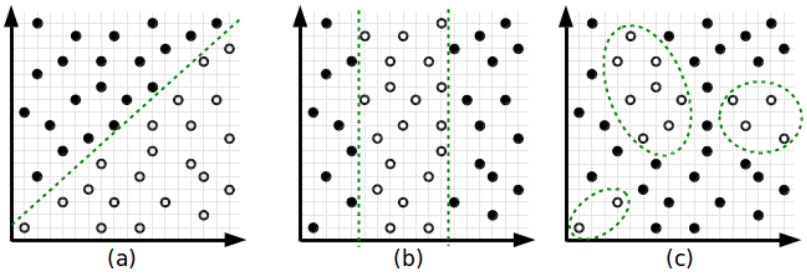


Figura 1: Tipos de Hipótesis

3. Repaso de probabilidades

Recomendamos revisar los apuntes de la materia Probabilidad y Estadística para resolver los siguientes ejercicios: `http://cms.dm.uba.ar/academico/materias/verano2018/probabilidades_y_estadistica_C/apunte-probaC-2017-2C.pdf`

Ejercicio 3.1. Explique por qué los siguientes eventos son independientes de a pares pero no independientes entre todos. Dadas 2 monedas,

(a) la primera moneda es cara;

(b) la segunda moneda cara;

(c) las dos monedas son iguales.

Ejercicio 3.2. Demostrar el teorema de Bayes: dados dos eventos A y B,

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Ejercicio 3.3. Demostrar el teorema de Probabilidad Total: dados una partición A_i del espacio muestral tal que $P(A_i) > 0$ para todo i, y un evento B:

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) \cdot P(A_i)$$

Ejercicio 3.4. Supongamos que la probabilidad de que un paciente tenga una forma determinada del virus de herpes es $P(herpes) = 0{,}008$. Se tiene un test con una sensibilidad de $0{,}98$ (es decir, $P(\oplus \mid herpes) = 0{,}98$) y una especificidad de $0{,}97$ (es decir, $P(\ominus \mid \neg herpes) = 0{,}97$), donde $\oplus$ y $\ominus$ representan los resultados positivo y negativo del test, respectivamente. Si un paciente se realiza el test y le da resultado positivo, ¿cuál es la probabilidad de que realmente tenga ese virus de herpes? Es decir, se pide calcular $P(herpes \mid \oplus)$.

Ejercicio 3.5.

(a) Escribir en Python una función $f(n, a, b)$ con n, a, b naturales tal que imprima en pantalla n números reales aleatorios en el intervalo $[a, b]$;

(b) Escribir en Python una función $g(n, m, v)$ imprima en pantalla n números reales aleatorios muestreando una distribución normal con media m y varianza v.

Plotear luego histogramas para $f(1000, 10, 20)$ y $g(1000, 3, 9)$.

4. Árboles de decisión

Cielo	Temperatura	Humedad	Viento	¿Salgo? (clase a predecir)
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

Tabla 1: Salgo a caminar

Ejercicio 4.1. Hacer en papel y lápiz un árbol de decisión correspondiente a entrenar con los datos de la Tabla *Salgo a caminar*. Utilizar el criterio "Gini Gain" para calcular el feature que mejor separa cada decisión. Armar luego tres ejemplos posibles de instancias nuevas (no existentes en la tabla) y usar el árbol para predecir la clase de salida. Calcular además la importancia de cada atributo (decrecimiento promedio en la impureza Gini).

Ejercicio 4.2. ¿Cómo cambiaría el árbol si restringiéramos su altura a 2 niveles?

(a) ¿Cuál sería el resultado de las predicciones del ejercicio anterior?

(b) ¿Algunas de las instancias existentes en la tabla, serían clasificadas incorrectamente?

Ejercicio 4.3. En la Figura *Cortes en el espacio de atributos* puede verse diversas regiones en el espacio de atributos.

- Determinar cuáles de ellas pueden haber sido generadas por árboles de decisión.
- Para las que lo sean, mostrar un árbol que hubiese generado estas regiones (suponer ejes x_1 y x_2)

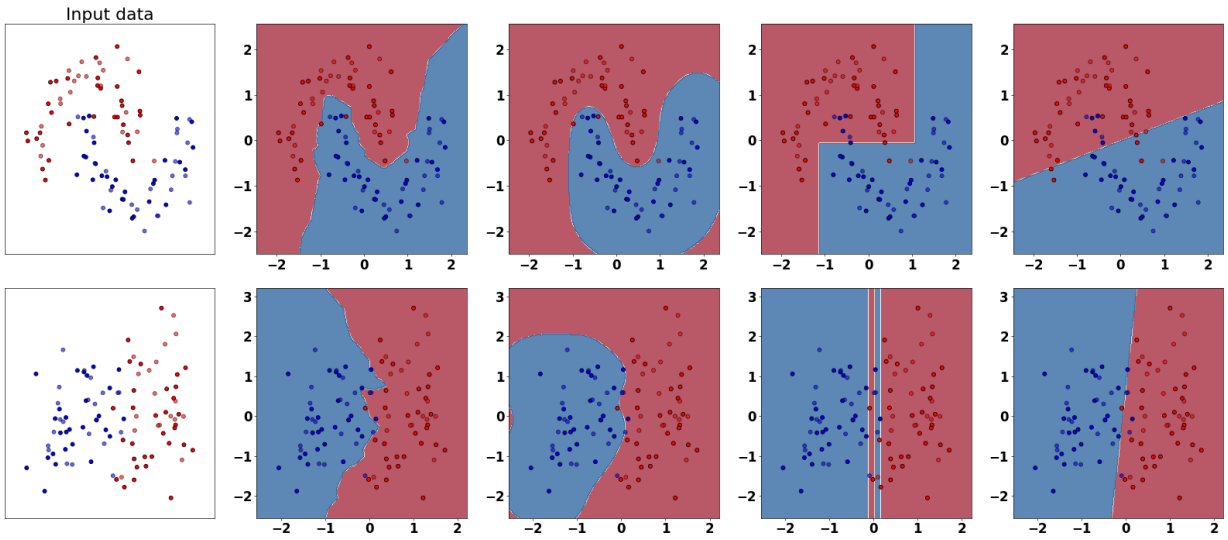


Figura 2: Cortes en el espacio de atributos

Ejercicio 4.4. Determinar cuáles de las siguientes son afirmaciones verdaderas:

(a) El objetivo de construir un árbol de decisión es crear el árbol de menor tamaño posible en el cual las hojas contengan valores de una sóla clase.

(b) Los algoritmos de construcción vistos (CART, ID3, etc) exploran todos los posibles árboles y se quedan con el que mejor separa a las instancias.

(c) La pureza describe qué tan cerca está un nodo de contener instancias de una sola clase.

(d) Un atributo puede aparecer sólo una vez en cada rama del árbol (llamamos rama a un camino directo desde una hoja hasta la raíz).

(e) Un par (atributo, corte) puede aparecer sólo una vez en cada rama del árbol (llamamos rama a un camino directo desde una hoja hasta la raíz).

(f) Para cada nueva instancia, un árbol permite predecir la clase a la que pertence. Por otra parte, para predecir **la probabilidad** de pertencer a una clase u otra, es necesario modificar el algoritmo de creación de árboles.

Ejercicio 4.5. Revisar y completar el notebook `notebook_3_arboles_de_decision_sklearn.ipynb`.

Ejercicio 4.6. Preguntas conceptuales para discutir:

(a) ¿Cuál el sesgo inductivo del algoritmo que construye el árbol de decisión?

(b) ¿Qué sucede cuando dos atributos empatan en ganancia de información? ¿Esta decisión es parte del sesgo inductivo?

(c) ¿Cómo se comporta la ganancia de información en comparación a impureza Gini cuando se comparan atributos con gran cantidad de valores distintos? Por ejemplo, si el atributo x_1 tiene dos valores posibles (true y false) y el atributo x_2 tiene 40 valores distintos, ¿es justo usar ganancia de información para elegir entre ellos? ¿Qué desventajas tiene? ¿Cómo se podría mitigar?

Ejercicio 4.7. Completar el notebook `notebook_4_implementacion_arbol.ipynb`. Este notebook contiene una implementación parcial de un algoritmo de creación de árboles de decisión.

5. Evaluación de modelos

Validación cruzada

Ejercicio 5.1. Verdadero o Falso. Justificar

(a) Un árbol de decisión puede conseguir 100 % de aciertos en los datos de entrenamiento.

(b) Un árbol de decisión consigue 100 % de aciertos en los datos de entrenamiento siempre y cuando no haya contradicciones entre las etiquetas de instancias muy similares.

(c) Hacer validación cruzada evita el overfitting de los modelos sobre los datos.

(d) Hacer validación cruzada ayuda a obtener estimaciones más realistas de la performance de un modelo sobre nuevos datos.

(e) En K-fold cross validation, conviene que K se acerque a N. De esta manera el resultado será lo más realista posible.

(f) Evaluar un modelo sobre el held-out set resultará en un valor siempre peor al conseguido en desarrollo.

Ejercicio 5.2. Explicar las posibles causas por las cuales un modelo entrenado y evaluado de la siguientes maneras puede funcionar peor que lo esperado al ser llevado a producción.

(a) Entrenado y evaluado sobre datos de entrenamiento.

(b) Seleccionado entre muchas posibilidades sobre datos de desarrollo (utilizando grid o random search junto a cross-validation).

(c) Seleccionado entre muchas posibilidades sobre datos de desarrollo (utilizando grid o random search junto a cross-validation) y evaluado en datos held-out.

Ejercicio 5.3. Preguntas

(a) En la clase teórica vimos que al hacer cross validation los datos no siempre deben separarse al azar, ¿por qué?. Pensar ejemplos de al menos dos situaciones en las cuales no sea conveniente.

(b) ¿Por qué se deberían usar una sola vez los datos held-out?

Métricas

Ejercicio 5.4. Matriz de Confusión

(a) En un problema de clasificación binaria, ¿a qué se denomina clase positiva y a qué clase negativa? Si nuestro problema consiste en clasificar spam vs. no-spam, ¿cuál es la clase positiva? Si nuestro problema es clasificar imágenes de perros vs. gatos, ¿cuál es la clase positiva?

(b) Explicar con tus palabras la definición de *verdadero positivo*, *verdadero negativo*, *falso positivo* y *falso negativo*.

(c) Completar la Primera Parte del notebook `notebook_5_metricas.ipynb`. El Test 1 debería pasar.

(d) ¿Por qué podría un falso positivo ser considerado más (o menos) importante que un falso negativo? Dar un ejemplo en donde es más grave tener falsos negativos que falsos positivos.

Ejercicio 5.5. Métricas

(a) Explicar con tus palabras la definición de *accuracy*, *precision* y *recall*.

(b) Completar la Segunda Parte del notebook `notebook_5_metricas.ipynb`. El Test 2 debería pasar.

(c) ¿Por qué es un problema medir *accuracy* de un clasificador para compararlo con otro? (Pensar en desbalance de clases.) Dar un ejemplo en donde sería engañoso utilizar esta comparación.

Ejercicio 5.6. Considerar la Figura *Umbral de clasificación*. En esta figura se ven instancias ordenadas según la probabilidad detectada por un clasificador (entre 0 y 1). Además, se encuentran marcados cuatro umbrales de decisión.

(a) Calcular las tablas de confusión resultantes para cada uno de los cuatro umbrales de decisión. Recordar que si la probabilidad está por debajo del umbral, la instancia será clasificada como perteneciente a la clase negativa; si está por encima, como clase positiva.

(b) ¿Cuál es el mejor umbral?

(c) Calcular la curva ROC para dicha clasificación.

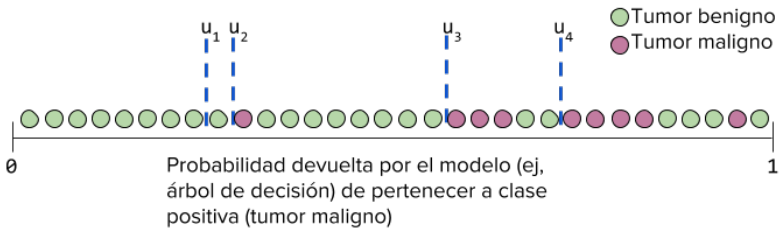


Figura 3: Umbral de clasificación

Ejercicio 5.7. Verdadero o Falso (justificar)

(a) Tanto *recall* como *precision* no toman en cuenta qué tan bien el modelo maneja los casos negativos.

(b) Un modelo que no produce falsos positivos tiene *precision* $= 1.0$.

(c) Un modelo que no produce falsos negativos tiene *recall* $= 1.0$.

(d) Si un clasificador devuelve probabilidades, la matriz de confusión se construye de manera ponderada según la probabilidad de cada clase.

(e) Si un clasificador devuelve probabilidades, hay muchas matrices de confusión asociadas dependiendo del umbral de clasificación.

(f) Si un clasificador devuelve probabilidades, hay infinitas matrices de confusión asociadas dependiendo del umbral de clasificación.

(g) Aumentar el umbral de clasificación produce que la *precision* siempre suba.

(h) Aumentar el umbral de clasificación produce que el *recall* baje o se mantenga igual.

(i) La métrica *precision* es parte fundamental del cálculo de la *curva ROC*.

Ejercicio 5.8. ¿Binaria o 2 clases?
Sean A y B clasificadores que distinguen entre imágenes de perros e imágenes de gatos. Al medir la performance del clasificador (utilizando F_1 para evitar los problemas de utilizar *accuracy*) y "gato" como clase positiva, obtenemos $F_1(A) = 0{,}9$, $F_1(B) = 0{,}8$. ¿Podemos concluir que el clasificador A es mejor que el clasificador B para este problema?
Resolver los siguientes ítems para poder responder a la pregunta:

(a) Al calcular F_1 utilizando "gato" como clase positiva, ¿importa qué ocurre con los perros que fueron clasificados correctamente? Revisar la Tercera Parte del notebook `notebook_5_metricas.ipynb` y decidir cuál clasificador funciona mejor, basándose en las métricas obtenidas. Observar el cambio que ocurre al intercambiar cuál es la clase positiva.

(b) ¿Para qué sirve el parámetro average en la función f1_score de la librería sklearn?

(c) ¿Qué sucede si la cantidad de instancias sobre las que fueron testeados es distinta? ¿Cómo se ve afectada la métrica F_1 al cambiar los True Negatives? Correr la Cuarta Parte del notebook `notebook_5_metricas.ipynb`. El gráfico muestra cómo varía la métrica F_1 al aumentar la cantidad de True Negatives (observar que estamos cambiando la cantidad de instancias sobre las que testeamos). ¿Qué se puede concluir de este experimento?

6. Clasificadores

Ejercicio 6.1. Imaginen que se tienen instancias con sólo dos atributos: altura de una persona (medido en metros) y edad de la persona (medida en años). Se quiere saber si la persona se convertirá o no es basquetbolista profesional tomando en cuenta la experiencia de muchas personas.

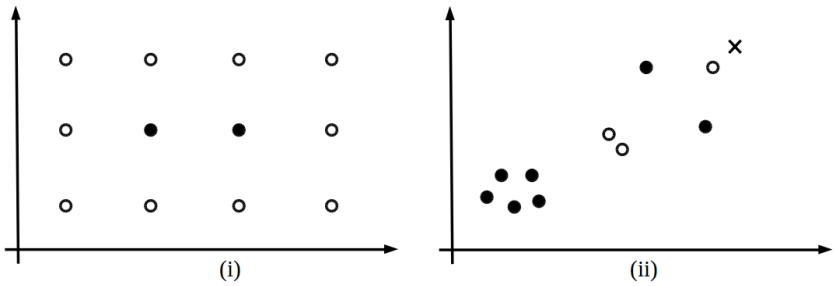
(a) ¿Es buena idea utilizar el algoritmo de K-vecinos más cercanos con estos datos?

(b) ¿Suponiendo que se utiliza dicho modelo, será útil realizar alguna transformación a los datos previo a ejecutar el algoritmo? ¿Cuál?

Ejercicio 6.2. Considerar el algoritmo de k vecinos más cercanos (kNN) para un conjunto de puntos bidimensionales, con dos clases posibles: ∘ y •. Tener en cuenta la distancia euclideana.

(a) En la figura (i), dibujar el Diagrama de Voronoi correspondiente a $k = 1$.

(b) Teniendo en cuenta los datos de la figura (ii), estimar la probabilidad de pertenencia a la clase ∘ de la nueva instancia ×, cuando $k = 1$, $k = 3$, $k = 5$ y $k = 10$.



Ejercicio 6.3. The Prosecutor's Fallacy.
Imaginen que se encuentran en un juicio en donde se debe determinar si una persona (Mónica Gaztambide) se encontraba o no conversando con un atracador en la escena de un crimen. Para ello se cotejan grabaciones de Mónica con grabaciones de la persona que se encuentra conversando con el atracador en el momento del robo. Un modelo entrenado para reconocer voces de personas sobre una gran cantidad de datos (es decir, un clasificador que predice nombres de personas) devuelve el siguiente resultado: $P(X = voz_{sospechosa}|y = monica) = 0{,}99$. ¿Dirían en base a este resultado que Mónica es culpable?

Ejercicio 6.4. Determinar cuales de las siguientes distribuciones alcanzan por sí solas para decidir la clase de una instancia $x^{(t)}$ (suponer clasificación binaria con clases "0" y "1").

(a) $P(y = 1|X = x^{(t)})$

(b) $P(X = x^{(t)})$

(c) $P(X = x^{(t)}|y = 1)$

(d) $P(X = x^{(t)}|y = 1)$ y $P(X = x^{(t)}|y = 0)$

(e) $P(y = 1)$

Ejercicio 6.5. Linear discriminant analysis.

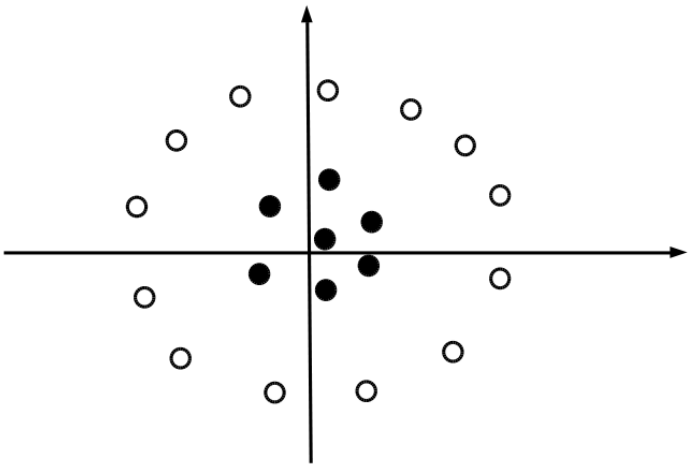
(a) Derivar la fórmula de las funciones discriminantes, a partir de la probabilidad a posteriori $P(Y = k|X = x)$ como vimos en clase:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

(b) Demostrar que la frontera de decisión para el caso $p = 1$, $k = 2$, $\pi_1 = \pi_2$ es $x = (\hat{\mu}_1 + \hat{\mu}_2)/2$.

(c) La palabra *linear* en LDA se debe a que $\delta_k(x)$ es una función lineal en x. Mostrar que si se elimina la suposición de que todas las clases tienen la misma varianza, entonces la función discriminante pasa a ser cuadrática en x. (A esa técnica se la conoce como *quadradic discriminant analysis*, o QDA).

Ejercicio 6.6. En el siguiente diagrama se muestran 20 puntos bidimensionales. Explicar qué puede hacer SVM con algún kernel para discriminarlos correctamente, y por qué SVM con un kernel lineal fallaría inexorablemente.



Ejercicio 6.7. Hasta ahora vimos 3 tipos de modelos de clasificación:

(I) Algoritmos que minimizan una función de costo asociado al error cometido al clasificar.

(II) Algoritmos basados en el clasificador de Bayes en donde se aproximan las distribuciones de la fórmula bajo ciertas suposiciones.

(III) Algoritmos que no construyen un modelo sino que se trata de heurísticas aplicadas para consultar los datos de entrenamiento al momento de clasificar.

Determinar en qué categoría se encuentran los siguientes algoritmos:

(a) Naive Bayes

(b) Linear discriminant analysis

(c) Árboles de decisión

(d) Support Vector Machines

(e) K-vecinos más cercanos

7. Ensambles

Ejercicio 7.1. Dar una explicación general del algoritmo Bagging.

Ejercicio 7.2. Dar una explicación general del algoritmo Random Forest. Basar la explicación en el algoritmo original presentado en el artículo: Breiman, Leo. "Random Forests." Machine learning 45.1 (2001). Incluir:

- ¿Cuál es su principal diferencia con Bagging?
- ¿Cómo funciona la estimación de error "out-of-bag"? [1]
- ¿Cómo propone Breiman medir la importancia de features? [2] ¿Qué diferencia hay con la manera en que la importancia se mide en el paquete sckit-learn de python? [3]

Ejercicio 7.3. Sea un clasificador binario de tipo Random Forest entrenado sobre un conjunto de datos de entrenamiento. Al evaluar 10 instancias el clasificador devuelve las siguientes probabilidades de pertenencia a la clase positiva: `[0.75, 0.75, 0.25, 0.75, 1.0, 0.0, 0.50, 0.50, 0.75, 1.0]`. Determinar la cantidad de árboles utilizados en el ensamble. Justificar.

Ejercicio 7.4. Explicar la idea conceptual del meta-algoritmo AdaBoost. Incluir:

(a) ¿Qué significa "weak-learners"?

(b) ¿Cómo se calculan los pesos para las instancias en cada iteración?

(c) ¿Cuál es el criterio para determinar la cantidad de modelos en el ensamble?

Ejercicio 7.5. (Opcional) Explicar la diferencia entre AdaBoost y GradientBoosting.

Ejercicio 7.6. (Opcional) Utilizar la implementación propia de árboles de decisión para construir modelos Random Forest siguiendo la interfaz de sklearn.

8. Diagnóstico de Sesgo y Varianza

Ejercicio 8.1. Supongamos que se construyen cuatro clasificadores para discriminar grabaciones de conversaciones en inglés contra grabaciones de conversaciones en español. La siguiente tabla muestra los resultados obtenidos según cuatro algoritmos.

(a) ¿Cuáles de los algoritmos dirían que sufren de alto sesgo?

(b) ¿Cuáles de los algoritmos dirían que sufren de alta varianza?

(c) Imaginen ahora que las grabaciones tienen mucho ruido de fondo y hasta un humano tiene problemas para detectar su origen, ¿dirían que el algoritmo A2 tiene sesgo alto? (suponer que los demás resultados no existen)

Algoritmo:	**A1**	**A2**	**A3**	**A4**
Accuracy (sobre entrenamiento)	.99	.90	.90	.99
Accuracy (sobre validación)	.89	.89	.75	.98

Tabla 2: Sesgo y Varianza

Ejercicio 8.2. Verdadero o Falso:

(a) Aumentar la cantidad de datos suele ayudar a contrarrestar problemas de varianza.

(b) Aumentar la cantidad de datos suele ayudar a contrarrestar problemas de sesgo.

(c) Un modelo muy complejo suele producir sesgo alto.

(d) Un modelo muy complejo suele producir varianza alta.

[1] ver Sección 3 del paper.
[2] ver Sección 10 del paper.
[3] ver la sección *1.11.2.5. Feature importance evaluation* en la documentación de scikit-learn sobre ensambles

(e) Sesgo alto está asociado a problemas de undefitting.

(f) Varianza alta está asociado a problemas de overfitting.

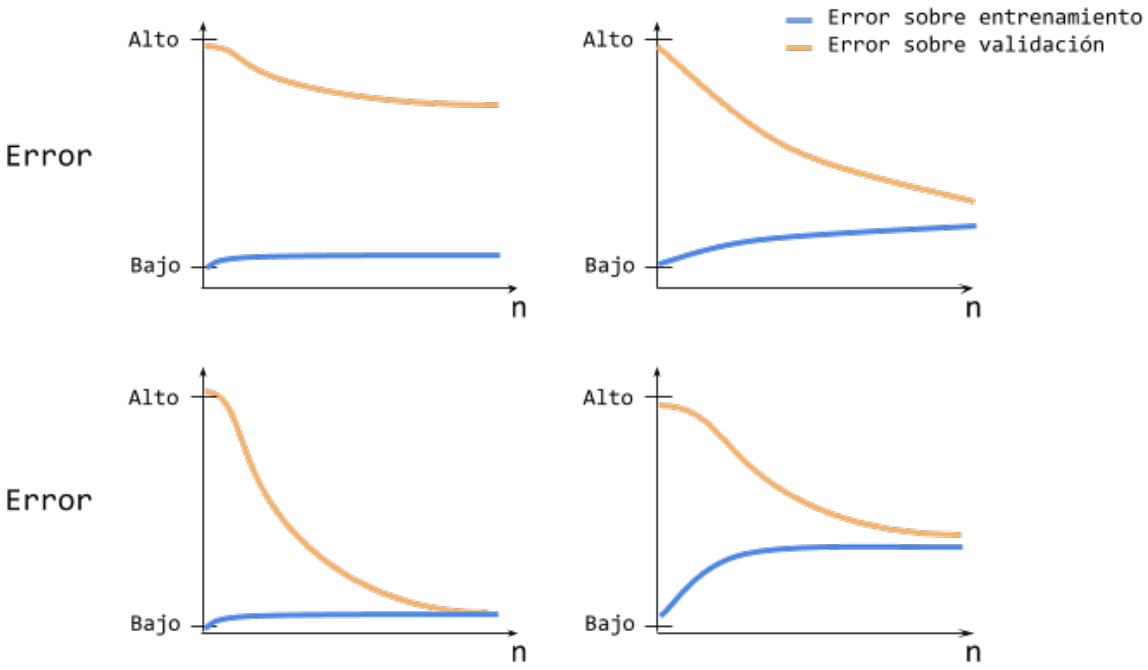
Ejercicio 8.3. Dadas las siguientes curvas de aprendizaje, diagnosticar si se trata de algoritmos con alto o bajo sesgo y alta o baja varianza. En base a su análisis, discutir posibles formas de mejorar la performance de sus modelos. Incluir:

(a) ¿Servirá recolectar más datos de entrenamiento?

(b) ¿Servirá construir un ensamble basado en el algoritmo analizado?

(c) ¿Servirá agregar o quitar features de nuestros datos?

(d) ¿Servirá modificar la altura máxima en árboles o cambiar el parámetro C en SVM?



9. Ingeniería de Atributos

Ejercicio 9.1. Tenemos un problema de clasificación con instancias con p atributos. ¿Cuántos posibles subconjuntos de atributos habría que explorar, si quisiéramos hacer una búsqueda completa? ¿Qué complejidad temporal tendría? ¿Qué complejidad temporal tienen las técnicas de *Forward selection* y *Backward elimination*?

Ejercicio 9.2. Dados los siguientes problemas de clasificación, indique qué transformaciones aplicaría a los datos antes de utilizar el algoritmo de K-vecinos más cercanos. En caso de no tener que hacer transformaciones justificar por qué. Todos los atributos agregados deberán ser transformaciones de atributos ya existentes entre los datos.

(a) Se quiere determinar si una persona utilizará una bicicleta de la ciudad para volver de su trabajo en base a: momento en el que salió del trabajo (timestamp que contiene fecha y hora), latitud y longitud en donde se encuentra su trabajo.

(b) Se quiere determinar si una persona visitará o no el barrio chino el día de hoy. Para eso, tenemos el día (fecha completa), la temperatura (medida en grados) y la longitud y latitud en la que se encuentra. Suponer que hay una gran cantidad de gente que visita el barrio en el año nuevo.

(c) Se quiere determinar la raza de un perro en base a: color de ojos (azul, verde, gris, etc.), color del pelo predominante (marrón, verde, gris, etc.), y tamaño del animal (chico, mediano, grande)

(d) Se quiere determinar si una persona nació o no en una región en base a su nombre y apellido.

¿Cómo modificaría la respuesta para el punto (c) en caso de utilizar árboles de decisión en vez de KNN?

Ejercicio 9.3. ¿Por qué se dice que PCA y MDS son técnicas "no supervisadas"?

Ejercicio 9.4. Sea un problema de clasificación binaria para instancias con 2 atributos (puntos en el plano). Dibujar un conjunto de instancias tal que un árbol de decisión obtenga mejores resultados si primero se aplica una transformación PCA sobre los datos. (Es decir, un árbol ajustado a los datos originales da peores resultados que otro árbol ajustado a los datos transformados.)

Ejercicio 9.5. Hacer un gráfico (teórico) que muestre cómo evoluciona el cubrimiento de la varianza de los datos (eje y) en función de la cantidad de componentes principales consideradas (eje x).

Ejercicio 9.6. Investigar qué son las *eigenfaces* (autocaras) y cómo se pueden usar para reconocimiento de caras.

Ejercicio 9.7. Verdadero o Falso:

(a) El resultado de PCA es una transformación que puede aplicarse a nuevos datos. De esta manera puede ser utilizado como una etapa de pre-procesamiento en el entrenamiento de un clasificador.

(b) La cantidad de componentes principales a utilizar se selecciona automáticamente eligiendo componentes hasta que la varianza explicada por las componentes restantes sea cero.

(c) El escalamiento de datos no afecta a PCA. Estandarizar a las instancias no produce ningún efecto en el resultado. (Estandarizar consiste en restar la media y dividir por la desviación estándar, para llevarlo a N(0,1).)

(d) Si queremos evaluar la performance de un modelo mediante cross-validation, lo correcto es hacer PCA antes de dividir los datos en folds.

10. Integradores

A continuación una serie de ejercicios que deberán implementar en el notebook `notebook_6_integrador.ipynb`. Para estos, consideraremos los siguientes conjuntos de datos:

Ejemplo A: Lectura de siglas (`siglas-dev.csv`)

Tarea: Determinar cómo pronunciar siglas en español. Este clasificador es útil en un sistema *text-to-speech* al encontrar una sigla desconocida en un texto a sintetizar.
Input: Sigla. Ejemplos: DGI, IBM, FMI, UBA, ALUAR, CONADEP. Por simplicidad, excluimos siglas con pronunciación especial: MIT (emaití), CNEA (conea), FCEN (efecen).
Output: Decidir si debe deletrearse (clase 'deletreo'), o leerse como un acrónimo ('acronimo').
Atributos:

`longitud`,
`cant_vocales`,
`cant_consonantes`,
`prop_vocales`,
`prop_consonantes`,
`cant_consonantes_vecinas`,
`cant_vocales_vecinas`,
`cant_consonantes_distintas`,
`cant_vocales_distintas`,
`cant_consonantes_iniciales`,
`cant_vocales_iniciales`,
`cant_consonantes_finales`,
`cant_vocales_finales`,
`cant_letra_[A-Z]` (26 atributos).

Ejemplo B: Reconocimiento del género del hablante (`genero.csv`)

Tarea: Determinar el género de una persona (masculino/femenino), a partir de una grabación corta de su habla.
Input: archivo wav.
Output: m / f.
Atributos: 1582 atributos acústicos extraidos con OpenSmile. Datos originales y más información: `http://habla.dc.uba.ar/gravano/ith-2014/tp2/`

Ejercicio 10.1. Experimentar con diferentes algoritmos de clasificación (árboles, Naive Bayes, SVM, KNN) para los ejemplos A y B, usando 10-fold CV. ¿Qué algoritmos funcionan mejor para los datos originales?

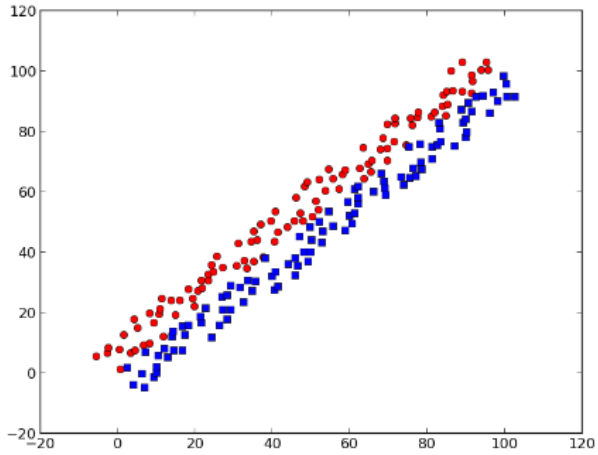
Ejercicio 10.2. Rankear los atributos de cada ejemplo según su ganancia de información. Conservar sólo los primeros k atributos (probar con distintos valores de k) y ver cómo afecta esto al desempeño de los algoritmos del ejercicio anterior. ¿Hay diferencia en los atributos encontrados si la seleccion se hace al principio o en cada iteracion del folding?

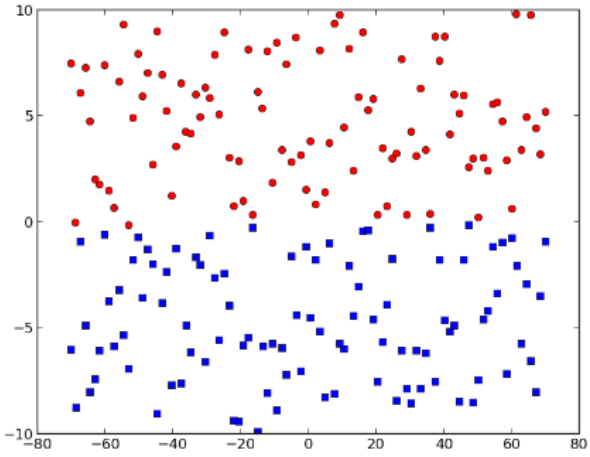
Ejercicio 10.3. Experimentar con otros métodos de selección de atributos. Utilizar las siguientes políticas de selección de atributos y comparar cómo funcionan en los dos datasets

- Ranking de atributos por p-valor: Para cada atributo comparar usando un test de hipótesis (ttest[4], kruskall[5], ks_test[6], etc) la diferencia entre la proyección del atributo separando por clase y se quede con los k más significativamente distintos.
- Random Search: Elegir un subset de k atributos al azar
- Genetic Search [7]
- Recursive Feature Elimination utilizando Random Forest como algoritmo base.

Usando los distintosalgoritmos de clasificación vistos en clase, ¿cómo afecta al desempeño hacer selección de atributos? En general, ¿cuán sensible es cada algoritmo de aprendizaje a la cantidad y calidad de atributos?

Ejercicio 10.4. En el notebook encontrarán un ejemplo de ejecución de Análisis de Componentes Principales (PCA) en Python, usando la biblioteca `sklearn.decomposition`. Este código primero genera al azar los datos de la izquierda, y los transforma mediante PCA en los datos de la derecha:





Transformar con PCA los datos de los ejemplos A y B, conservar sólo las primeras k Componentes (probar con distintos valores de k) y volver a evaluar los algoritmos de clasificación.

[4] `http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ttest_ind.html`
[5] `http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.mstats.kruskalwallis.html`
[6] `http://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.ks_2samp.html`
[7] `http://topepo.github.io/caret/feature-selection-using-genetic-algorithms.html`