Clase 5: Procesos en minería de datos

Simón Leiva Meza

Pontificia Universidad Católica de Chile
Facultad de Matemáticas
Programa de Magister en Estadística

Introducción
Enfoques del Analytics

¿Qué pasó?

Analítica Descriptiva

Buscamos entender el proceso, describir lo que ha ocurrido en base a información pasada. Esto nos permite tener una major comprensión del problema, y desde ya pensar como se podría solucionar.

¿Qué podría pasar?

Analítica Predictiva

El foco es que mediante el uso de modelos, podamos saber que es lo que ocurrirá (o lo que podría ocurrir), esto permite tomar decisiones para tomar acciones futuras.

¿Qué debiéramos hacer?

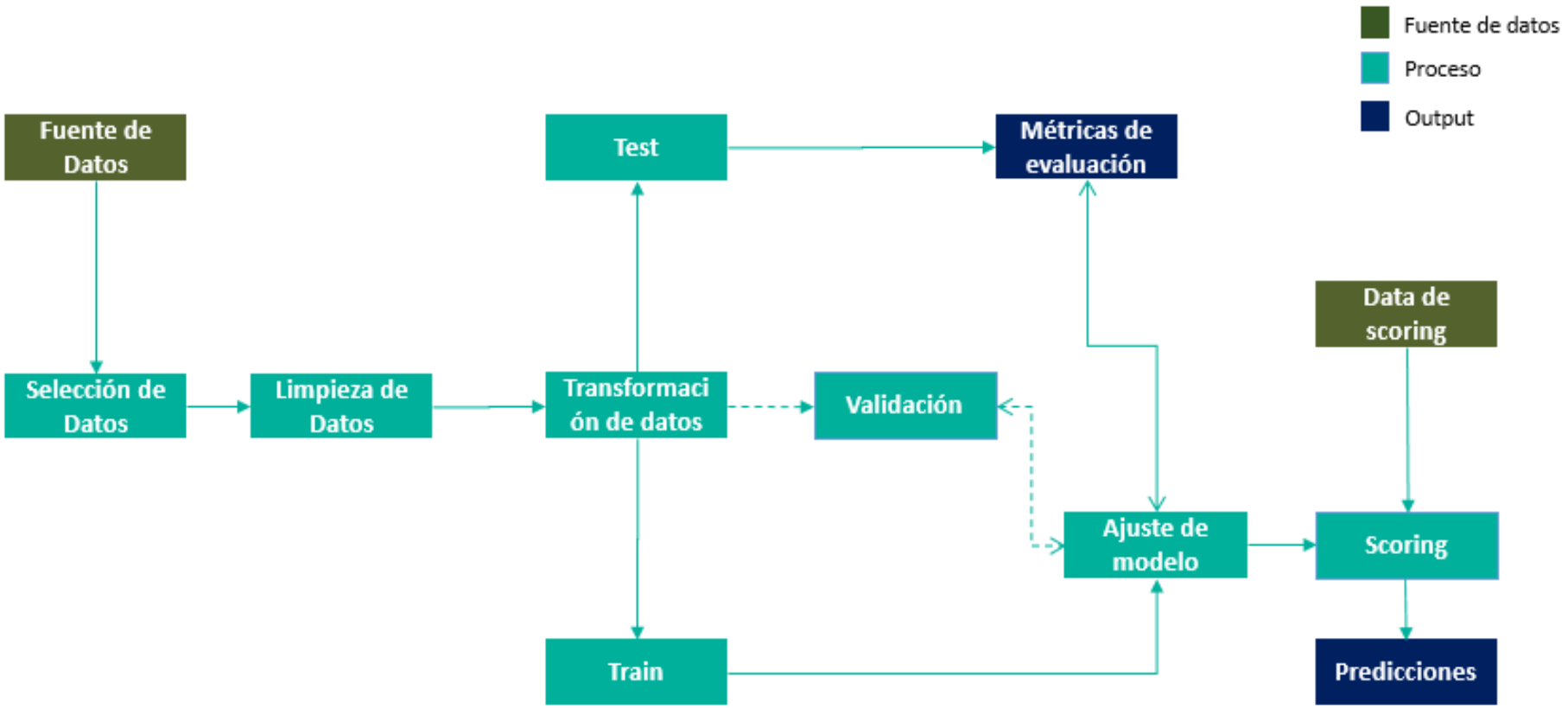
Analítica Prescriptiva

En base a modelos predictivos, se busca generar escenarios o simulaciones de lo que podría ocurrir, esto ayuda a guiar la desición que conduce al scenario ganador.

El Ciclo de un Modelo

La primera clase se mostró cómo es el ciclo del proceso KDD, sin embargo esa tiende a ser una visión general, la cual aún puede ser ambigua para el proceso de modelamiento predictivo.
En la siguiente lámina, un flujo estándar que suelen tener los procesos de modelamiento.

El Ciclo de un Modelo



El Ciclo de un Modelo

	Explicación	Ejemplos
Fuentes de datos	Pueden ser relacionales o no relacionales, son todos los orígenes de información	Extracción por ODBC, HDFS, texto plano, consulta de API, etc.
Selección de datos	Proceso en el cual se reduce la dimensión de los datos originales mediante selección de variables por viabilidad.	No utilizaré el número de teléfono ni el rut para ajustar un modelo de fuga. Solo puedo usar variables financieras para mi modelo de provisiones.
Limpieza de datos	Hace referencia a todos los procesos de transformaciones tal que las variables representen lo que deben representar.	Los datos tienen que comenzar de la misma fecha. Hay que rellenar las fechas faltantes con ceros, hay que categorizar los ingresos en rangos.
Transformación de datos	En lugar de reducir dimensionalidad, se expande o reduce a fin de encontrar mejores patrones ocultos.	Genero el los promedios de los últimos tres meses, aplico expansión por kernel, escalo los datos.
Train	Datos con los cuales se ajustan los modelos.	Ventas históricas, comportamiento financiero del cliente, colección de imágenes etiquetadas.
Validación	Datos excluidos a fin de ir probando mejores ajustes con data no usada para entrenar.	20% de imágenes fuera, próximas 2 semanas, dejar 20% aleatorio fuera, dejar uno fuera rotativo.

El Ciclo de un Modelo

	Explicación	Ejemplos
Test	Data reservada a fin de probar la efectividad del modelo.	Dejar las últimas tres semanas fuera para probar la capacidad predictiva del modelo.
Ajuste de modelo	Usando la data de Train (y a veces la de validación también) se ajusta el modelo.	Clasificación por árboles de decisión, pronóstico por ARIMA, regresión con modelos aditivos generalizados.
Métricas de evaluación	KPI o gráficas a fin de observar la calidad predictiva y explicativa del modelo.	RMSE, AUC, Accuracy, MAPE, curva ROC, Lift.
Scoring	Ponderación de nueva data de acuerdo al modelo ajustado.	Obtener los propensos a fuga dado el modelo ajustado, aplicar reconocimiento de imágenes.
Data de scoring	Datos para ser aplicados en el modelo.	Nuevas imágenes, atributos de clientes, etc.
Predicciones	Resultado de los modelos luego de ser scoreados.	Probabilidad de fuga, etiqueta de imagen, pronóstico de demanda.

Fuentes de datos

Bases SQL
En general la mayoría de las bases SQL, al menos las más populares pueden conectarse mediante ODBC, en Python existen las librerías Pyodbc y también SQLalchemy, las cuales permiten conectarse mediante el string ODBC de conexión.

Formatos de texto
Formatos de texto como txt, csv, json, etc. Gran parte de estas simplemente se pueden leer con pandas mediante la función pd.read_xxxx, donde xxxx es csv, json, etc.

Otros formatos
Imágenes, audio, páginas webs. Data no relacional guardadas en data lakes, generalmente el proveedor de la nueve tienen librerías para conectarse. Para audios, videos, web requieren librerías y funciones especializadas.

Procesos de datos

Mala calidad del dato
Evaluar la calidad del datos ¿Está muy desordenado? ¿Es confiable la data?. En ese caso pedir apoyo externo de especialista, validar información con ellos. Nunca comenzar hasta que haya seguridad del dato.

Datos faltantes
En caso de ser muchos (más de 50%) descartar la variable, en caso de estar entre un 20% y 50% evaluar generar una categoría missing. Si hay menos imputar, lo más fácil es por media, mediana o por clúster.

Data no explica
Realmente no explica? Evaluar con especialistas si tienen sentido las variables. Estudiar distribuciones ¿Son muy asimétricas? ¿existen outliers? ¿Se correlacionan con la variable respuesta? Lo primero se puede obtener gráficamente o con coeficiente de asimetrías, la correlación gráficamente o con métricas.

Procesos de ajuste

Train
La data de train por lo general se toma 60% a 80%. Cuando la volumetría de datos es muy elevada y la metodología es simple, se puede tomar un 40% o incluso menos!. Como regla general, mientras más complejo es el modelo les gustaría tener un mayor porcentaje de train.

Test
Por lo general un 30% a 20%. Al principio del proceso apartar, y no volver a calcular para hacer los resultados comparables. Ideal reservarla aparte en otra tabla o bien generarla por semilla.

Validación
No es usado en todos los métodos pero es aconsejable, su tamaño varía dependiendo del tipo de validación, pero nunca sobre 20%. Algunas formas de validación incluyen k-fold cross validation, holdout validation o leave one out cross validation.

Próximos pasos

Proceso Ajuste
Se sugiere seguir un estructura de clases, con métodos que hagan cada parte del proceso a fin de ser replicable con mayor facilidad o bien al menos con cada paso modularizado, es decir sin variables globales. El código debe estar versionado y documentado.

Proceso scoring
Proceso más ligero, no debiera ajustar el modelo otra vez a menos que se descalibre constantemente. Importante ir haciendo seguimiento de la calidad predictiva del modelo con cada ejecución.

Automatización
Cuál es la periodicidad del proceso? Cómo se va a consumir? Que tan rápido es el scoring? Si es de baja periodicidad basta con un proceso batch alojado en un servidor, si es de periodicidad alta es mejor opción disponibilizar mediante API REST, consultándose a medida que lo requiera el usuario. No olvidar documentación y logs de cada proceso

Versionamiento

Versionamiento

El versionamiento de código es una buena práctica en el desarrollo de productos basados en código. Algunas de sus ventajas son.

- Otorgan mayor control sobre cambios al código.
- Disminuye el riesgo asociado a cambios inesperados, malos guardados o errores de hardware.
- Facilita el trabajo en equipo
- Permite tener un mejor seguimiento de los cambios hechos.
- Mantiene versiones estables mientras se trabajan otras de forma simple.
- Facilita migraciones de arquitectura.

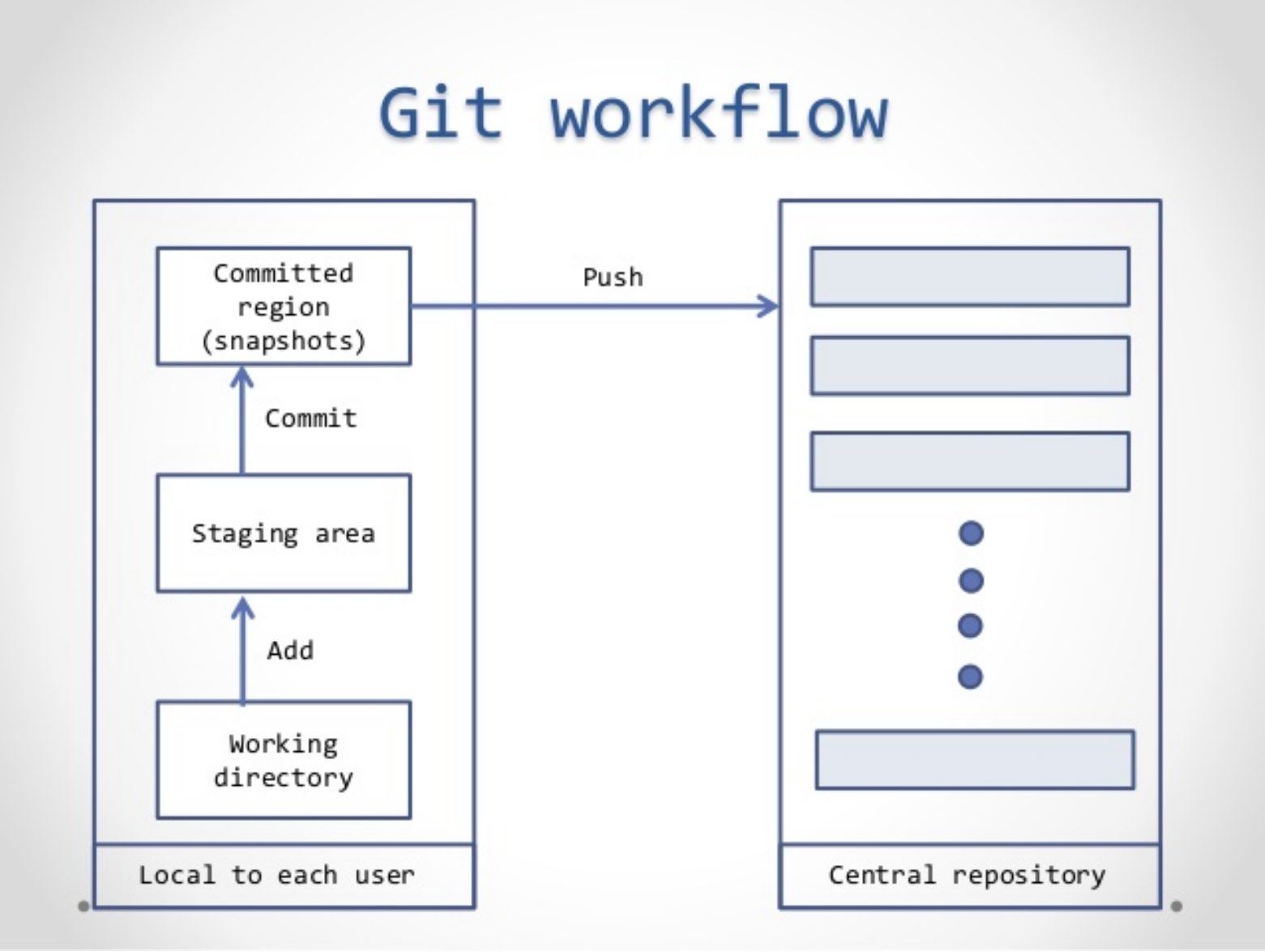
Uso de GIT

Git es un software pensando en la eficiencia y la confiabilidad del mantenimiento de versiones. Es uno de los tantos sistemas de control de versiones que existen. Si bien veremos como instalarlo y sus comandos básicos, no profundizaremos más en su uso, pues se escapa al alcance del curso.

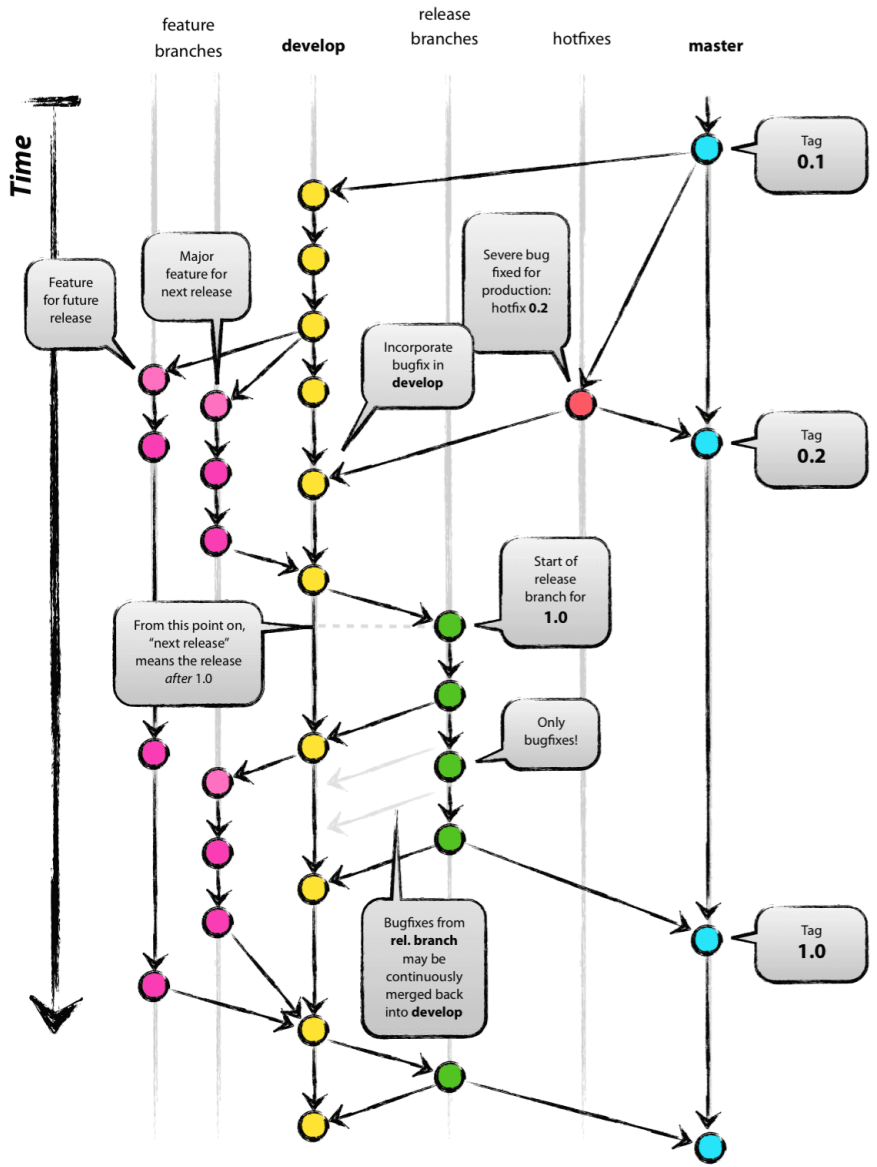
Para usarlo deben descargar GIT en la página
`https://git-scm.com/`

Una vez descargado, instálenlo con las opciones default.

Conceptos Básicos



Esquema de versionamiento



Ejercicios

1. Créese una cuenta en github y cree un repositorio privado.
2. Clone el repositorio que creo
3. Versione la ruta donde va a guardar el proyecto de esta clase
4. Cree un .txt con algo dentro y añádalo al área de staging
5. Haga un commit para generar el snapshot de los cambios en el repositorio local
6. Cree una rama development a partir de la rama principal master.
7. Cámbiese a la rama development
8. Añada un par de líneas al .txt
9. Vuelva a la rama master y vea los cambio
10. Una la rama development con la master
11. Revise el log de los cambios hechos
12. Suba los cambios a github

Ejercicio: Preparando datos

En la página del curso se encuentran disponibles los datos hongos.csv, el cual consiste en una tabla con atributos de hongos. El objetivo será predecir en base a atributos visuales cuando un hongo es venenoso o comestible.
La idea es hacer un modelo simple, interpretable que se nos ayude a nosotros a saber cuando comer o no una seta silvestre.
En el Archivo hongosdiccionario.xlsx se encuentran las descripciones de algunas de las columnas.

Ejercicio: Preparando datos

Vamos a hacer entre todos un modelo predictivo muy sencillo, sin transformaciones ni mayor ingeniería de atributos, en particular tomemos la siguiente guía:

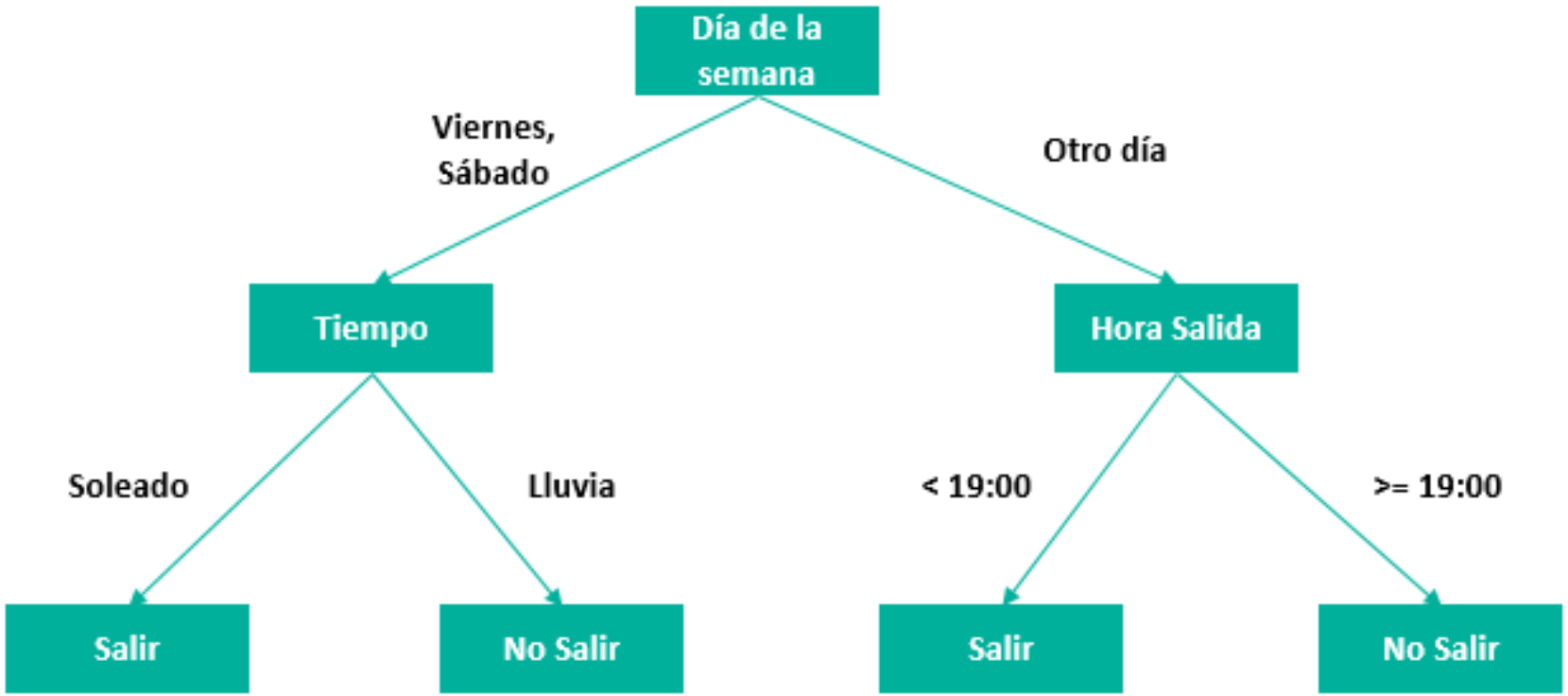
1. Genere una función que lea las fuentes de datos y retorne un data frame con los datos seleccionados, listos para ser limpiados. Dado que buscamos hacer un modelo en base a características visuales, estas son las que nos debieran interesar.
2. Genere una función que tome como argumentos los datos seleccionados, y evalúe la cantidad de nulos (si es que hubiera), y tome medidas en caso de haber muchos de estos.
3. Genere una función que reciba como argumento los datos transformados, y genere una partición de train y test de acuerdo a porcentajes indicados por el usuario.

Árboles de Decisión

Los árboles de decisión son considerados algoritmos débiles, esto quiere decir que tiene baja capacidad predictiva frente a otro algoritmo para un mismo conjunto de datos. En la práctica es de las metodologías más simples pues simplemente son decisiones con puntos de elección elegidos inteligentemente.

La principal virtud de un árbol de decisión es su interpretabilidad y su capacidad de ensamblaje para armar modelos más fuertes.

Árboles de Decisión

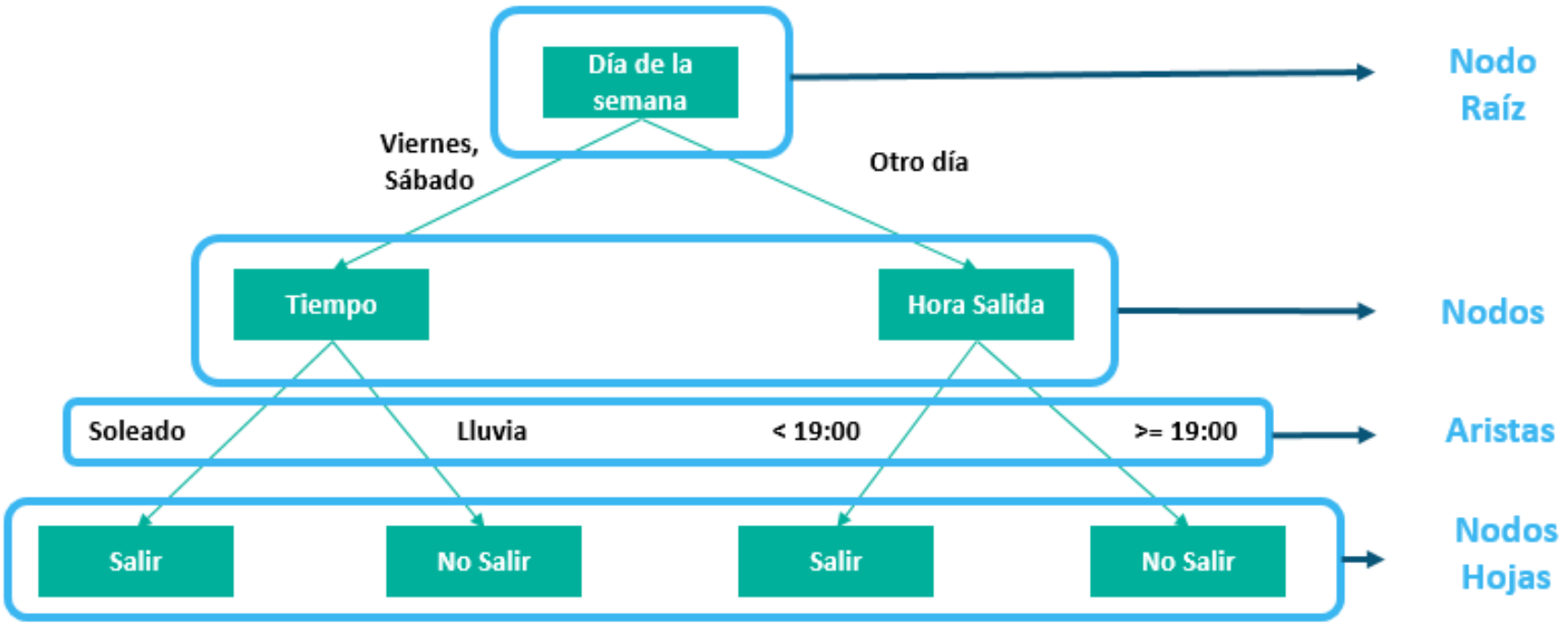


Árboles de Decisión

Un árbol tiene los siguientes componentes

- **Nodo:** Corresponde a un atributo, en nuestras bases correspondería por ejemplo a una columna. Al primer nodo se le suele llamar nodo raíz.
- **Arista:** Son las reglas para pasar de un nodo a otra.
- **Hoja:** Corresponde al último nodo del árbol, esta contiene la variable respuesta, es decir contiene una de las clases a predecir.

Árboles de Decisión



Árboles de Decisión

Vamos a ajustar nuestro primer árbol en Python, tomemos como base los datos que preprocesamos. Va a ser algo sencillo sin adentrar aún en los detalles.

Árboles de Decisión
Más detalles

Algunas de las preguntas lógicas son las siguientes:

1. ¿Por qué atributo partir?
2. ¿por qué atributo seguir?
3. ¿Cuál es la regla de decisión?
4. ¿Cuando dejamos de generar mas nodos hijos?

Árboles de Decisión

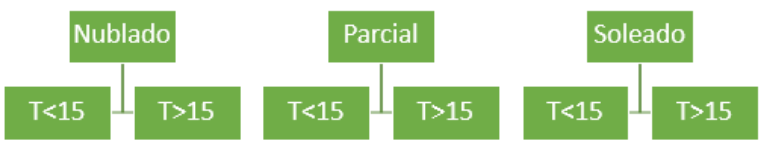
- El mejor nodo es el que tiene mejor poder de separar la información en clases en las cuales una sola clase predomina en cada grupo.
- Para medir esto último se suele usar la Entropía, definida como:

$$H(X) = -E(\log(X)) = -\sum_{i=1}^{n} P(x_i)\log(P(x_i))$$

Esta métrica nos entrega un grado de desorden, cuando las clases son homogéneas (por ejemplo todas cero o 1), este valor es mínimo. Por el contrario, cuando son menos homogéneas se maximiza.

Árboles de Decisión

Llueve	Temperatura	Cielo
1	12	Nublado
1	9	Nublado
1	16	Nublado
1	5	Parcial
0	14	Soleado
0	13	Nublado
0	24	Parcial
0	31	Soleado
0	13	Parcial



Queremos hacer un árbol para determinar si llueve o no. Tenemos dos posibles reglas a modo de ejemplo

Regla 1: llueve si la temperatura es menor a 15
Regla 2: de acuerdo a como se ve el cielo decidimos.

La entropía de la lluvia es:

$$H(L) = -\frac{4}{9}\log\left(\frac{4}{9}\right) - \frac{5}{9}\log\left(\frac{5}{9}\right) = 0.2983$$

Por otra parte la entropía dado que elegimos la regla 1 es

$$H(L|T) = \frac{6}{9} \cdot H(L|T < 15) + \frac{3}{9} \cdot H(L|T \geq 15) = 0.2928$$

Y con la regla 2 es

$$H(L|C) = \frac{4}{9} \cdot H(L|C = N) + \frac{3}{9} H(L|C = P) + \frac{2}{9} H(L|C = S) = 0.2006$$

Luego

$$IG(L, R_1) = 0.005$$
$$IG(L, R_2) = 0.097$$

Por tanto el árbol parte con el nodo de elección según el cielo

Árboles de Decisión

- Para elegir un nodo, nos basamos en el criterio de ganancia de información, esto es cuanta entropía ganamos al usar una partición.

$$IG(X, I) = H(X) - H(X|I)$$

  La elección del próximo nodo debiera ser la que nos de siempre la mejor ganancia de información.
- Otras métricas son el Gini, el Chi cuadrado y la tasa de clasificación.

Árboles de decisión

Árboles de Decisión

- **Gini:**

$$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

- **Chi-cuadrado:**

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Árboles de Decisión

Existen muchas maneras de ajustar un árbol, pero los algoritmos difieren principalmente en la métrica usada para generar la división. Algunos de los algoritmos son lo siguientes:

1. **ID3:** Dicotomizador iterativo 3, usa la ganancia de información a fin de seleccionar los mejores nodos.
2. **CART:** Classification and Regression Trees, usa el Gini como métrica.
3. **CHAID:** Chi Squared Automatic Interaction Detection, usa como métrica el estadístico chi-cuadrado de independencia.

Árboles de Decisión

Para evaluarlos, se puede usar la matriz de confusión:

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment)	
		positives	negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative