Organización de Datos

Tomás Villegas

2022-2C

Índice

1. Introducción

2. Introduccion a la Ciencia de Datos

La ciencia de datos, estamos aplicando en general, machine learning. Donde machine learning es el aprendizaje automatico que se basa en ciertas tecnicas o algoritmos para resolver problemas complejos que quizas no tienen una solucion unica o alguna solucion optima. Entonces necesitamos algunas euristicas o atajos para encontrar una solucion lo suficientemente buena. *INSERTAR DIAGRAMA DE VENN DE LA PPT*

2.1. Variables

Cuando nosotros implementamos/creamos nuestro clasificador, generalmente utilizaremos ciertos datos de entrada que al pasar por nuestro clasificador, nos devolveran en la salida nuevas variables. Generalmente podemos discernir entre las variables de entrada y salida como

- Variables Inpendientes, que son todas las variables de entrada.
- Variables Dependientes, que son todas las variables de salida/categorias.

Pero a veces, se puede dar que en las variables de entrada existan algunas variables dependientes. Por ejemplo, digamos que tenemos un dataset de casas de la linda ciudad de Ezeiza, entre nuestro dataset tenemos las siguientes columnas:

- Direccion
- Habitaciones

- Latitud
- Longitud
- Codigo Postal

En este caso, el codigo postal depende de la latitud y longitud de la casa, es decir, de la ubicacion de la misma. Las variables independientes se pueden clasificar de la siguiente forma:

- Cualitativas
  - Texto
    - Nominales (Una columna paises)
    - Ordinales (Una columna que describa la cantidad de forma textual, por ejemplo 'poco, 'mucho', 'muchisimo')
  - Numericas
    - Nominales
    - Ordinales
- Cuantitativa
  - Discreta
  - Continua

2.2. Variables y tipos de Problemas

1. Si la variable dependiente es *cualitativa*, el tipo de problema es de *clasificacion*
2. Si la variable dependiente es *cuantitativa*, el problema es de *regresion*
3. Si *NO* hay variables dependientes, el problema es de *agrupamiento*

2.3. Outliers - Valores Atipicos

Son valores que no estan dentro del rango esperado. Esos valores pueden ser errores de medicion, puede ser un valor real muy poco probable de ocurrir. Generalmente podemos eliminarlos o ponderarlos, porque puede darse el caso en el que queremos encontrar los valores atipicos.

2.4. Correlacion de Variables

Si dos variabels estan correlacionadas, quiere decir que dos variables varian de igual forma sistematicamente.

2.4.1. Correlacion *NO IMPLICA* Causalidad

- Que dos variables tengan alto indice de correlacion no significa que una cause la otra.
- Las relaciones de causalidad son mucho mas dificiles de encontrar y demostrar
- Las correlaciones pueden por otros motivos como que haya una tercer variables que .empuja.a ambas o simplemente el azar.

2.4.2. Varianza

La varianza es el promedio entre todas las observaciones respecto de su media. Nos permite saber como estan distribuidas las observaciones respecto de la media.

2.4.3. Covarianza

En probabilidad y estadistica, la *covarianza* es un valor que indica el grado de variacion conjunta de dos variables aleatorias respecto de su media.

Es el dato basico para determinar si existe una dependencia entre ambas variables y ademas es el dato necesario para estimar otros parametros basicos, como el coeficiente de correlacion lineal o la recta de regresion.

2.4.4. Correlacion de Pearson

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{Cov(X,Y}{\sqrt{Var(X)Var(Y)}} \tag{1}$$

Donde

$$\sigma_{XY} es la covarianza de de (X,Y)$$
$$\sigma_X es la desviacion estandar de X$$
$$\sigma_Y es la desviacion estandar de Y$$

Para dos variables, podemos medir su correlacion lineal con el coeficiente ρ (Pearson). Este coeficiente es una funcionque mide cuan relacionadas estan dos variables de forma linea.

- Si es 0 *NO EXISTE CORRELACION LINEAL*
- Si es 1 estan relacionadas linealmente de forma perfecta (todos los puntos estan en una linea)
- Si da -1 estan relacionadas linealmente de forma negativa perfecta

2.4.5. Correlacion de Pearson: desvio estandar

Es una medida que se utiliza para *cuantificar la variacion o la dispersion de un conjunto de datos numericos*. Una *desviacion estandar baja* indica que la mayor parte de los datos muestra tienden a estar agrupados *cerca de su media*, mientras que una *desviacion estandar alta* indica que los datos se extiende sobre un rango de valores mas amplio.

2.5. Modelos de Regresion

2.5.1. Regresion

La primera forma de regresion lineal documentada fue el metodo de de los minimos cuadrados que fue publicada por *Legendre en 1805*. Gauss publico un trabajo en donde desarrollaba de manera mas profunda el metodo de de los minimos cuadrados. El concepto de regresion proviene de la genetica y fue popularizado por *Sir Francis Galton* a finales del siglo XIX con la publicacion de *Regression towards mediocrity in hereditary stature*. Galton observo que las *caracteristicas extremas* (Por ejemplo la estarura) de los padres no se transmiten por completo a su descendencia. Mas bien, las caracteristicas de la descendencia retroceden a un punto mas mediocre (un punto que desde entonces ha sido identificado como la media)

```
# Pseudo codigo de minimos cuadrados para generar una recta de
    regresion lineal
def calcular_coeficientes(xarray, yarray, largo):
    x = y = xy = xx = a = b = resultado = 0
    for i in range(largo):
        x += xarray[i]
        y += yarray[i]
        xy += xarray[i]*yarray[i]
        xx += xarray[i]*xarray[i]
    b = ((largo * xy) - (x * y) / ((largo * xx) - (x * x))
    a = (y - (b - x)) / largo
```

Cabe mencionar que el ajuste lineal es muy sensible a outliers.

2.6. Metodos de Clasificacion

Cuando resolvemos un problema de clasificacion, buscamos, para ciertos datos de entrada, una categoria c de un conjunto C de categorias posibles. Estas categorias no solo son finitas, sino que son conocidas de antemano.

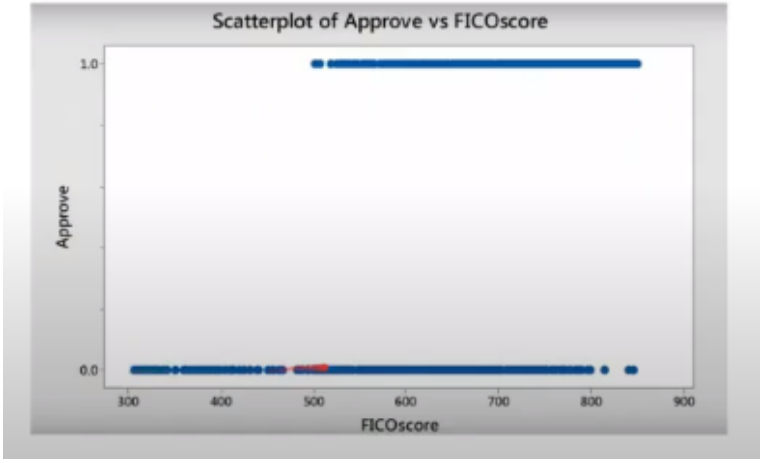
2.6.1. Regresion Logistica

En la regresion logistica lo que quiero es categorizar, clasificar. Es decir que dado una serie de puntos quiero encontrar una funcion (no es una recta en este caso) que separe los puntos en dos conjuntos. Y una vez que encontre el conjunto

puedo determinar cualquier valor de X futuro, el conjunto al cual pertenecera. Esta asociado a problemas de probabilidad.

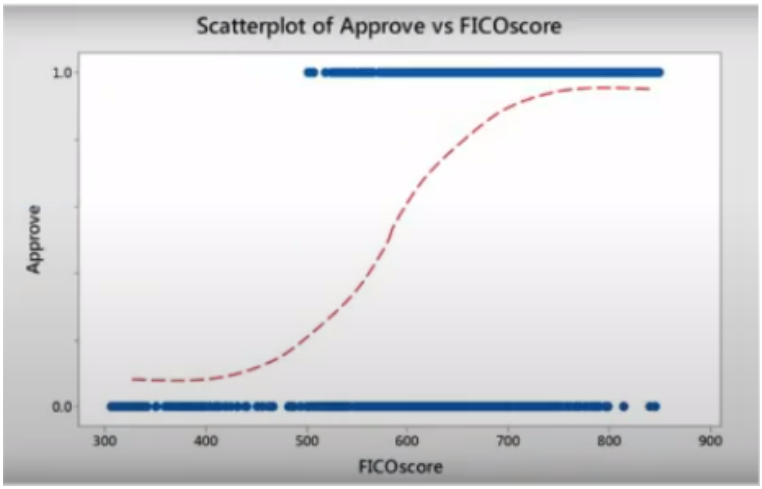
Una persona quiere comprar una casa y para ello necesita pedir un prestamo hipotecario. Esta persona quiere saber si se lo van a otorgar o no. Pero el unico dato fehaciente que tiene es su puntaje crediticio, el cual es de 720. Ahora, supongamos que tenemos un conjunto de datos con las siguientes columnas

1. *creditScore* (El puntaje crediticio, variable independiente)
2. *approved* (La variable dependiente, indica si el credito fue aprobado o no)



En la anterior tabla, se puede ver que antes de llegar a los 500 puntos de creditScore, no se aprobo ningun credito, pero, al pasar la barrera de los 500 puntos, vemos que ya empieza a haber una cantidad considerable de aprobados y luego de pasar las barrera de los 800 puntos, empezamos a ver que flaquea el hecho de ser desaprobado. Para este caso, podemos utilizar la *funcion sigmoide*, que es un caso particular de la *funcion logistica*.

$$Probabilidad\ De\ Una\ Clase = \theta(y) = \frac{1}{1 + e^{-}x}$$



Podemos ver que en el grafico hay una funcion que va entre 0 y 1. Es decir, nos sirve para ver la probabilidad de que a una persona le aprueben el credito en base a su puntaje crediticio.

2.7. Metodos de Clusterizacion

Los metodos de Clusterizacion, se centran en problemas de agrupamiento de datos. Agruparlos de tal manera que queden definidos N conjuntos distinguibles, aunque no necesariamente se sepa que signifiquen esos conjuntos. El agrupamiento siempre sera por caracteristicas similares.

2.7.1. K-Means

1. El usuario decide la cantidad de grupos.
2. K-Means elige al azar K centroides
3. Decide que grupos estan mas cerca del centroide. Estos puntos forman un grupo
4. K-Means recalcula los centroides de cada grupo
5. K-Means vuelve a reasignar los puntos usando los nuevos centroides. Calcula los nuevos grupos.
6. K-Means repite los puntos 4 y 5 hasta que los puntos no cambian de grupos

2.8. Entrenamiento

2.8.1. Conjuntos de entrenamiento y Prueba

Cuando tenemos metodos supervisados, nostros tenemos que decirle al metodo, con que datos lo vamos a entrenar. El conjunto de datos de entrenamiento,

es un conjunto de datos que tiene asignada la categoria de la variable de salida, es decir, la variable dependiente que nosotros queremos que regrese para determinada observacion. Por ejemplo, un metodo supervisado debe ser entrenado, en cambio, K-Means no es un metodo supervisado, porque a K-Means no le tenemos que dar un conjunto previamente categorizado. Las tecnicas mas comunes realizadas en nuestro conjunto de datos, es tomar una parte del conjunto como entrenamiento y otra parte como test. Generalmente se toma como un 70 % de entrenamiento y otro 30 % para test.

2.8.2. Conjuntos balanceados

A veces, nuestros conjuntos de datos pueden llegar desbalanceados, pueden haber mayoria de casos de algo en particular y lo que a nosotros nos interesa analizar, parte de esa minoria del conjunto, entonces, para este tipo de problemas existen dos tipos de estrategias:

- **Undersampling** Remover muestras del caso mayoritario para ponerlo a la par con las muestras minoritarias
- **Oversampling** Agregar muestras al caso minoritario para ponerlo a la par con el caso mayoritario

2.8.3. Model Fitting

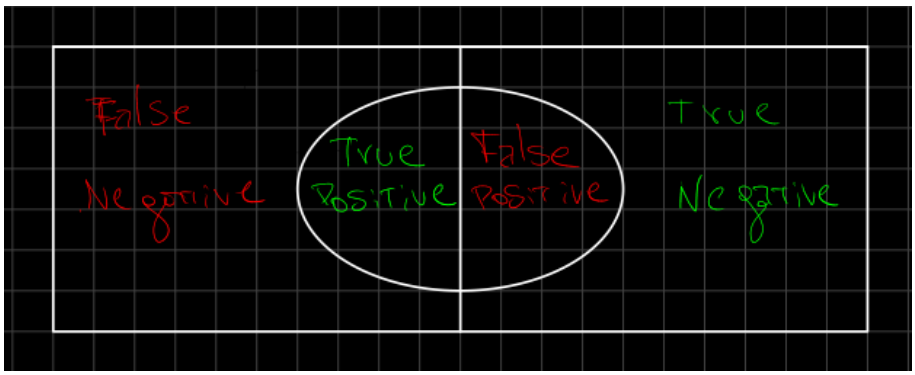
Algunas veces, nuestro modelo puede estar realizando un muy bajo rendimiento al esperado, por lo que, es importante comprender correctamente nuestro modelo para encontrar la raiz del problema de nuestra baja exactitud.

- Underfitting
- Overfitting
- Balanceado

2.9. Metricas

A veces, cuando realizamos metricas de nuestro analisis de datos, podemos llegar a caer en dos tipos de errores muy comunes:

- *Falsos Negativos*: Los falsos negativos se dan cuando nuestro clasificador, determina que uno de los datos es invalido cuando en realidad era correcto. Ejemplo: En controles de calidad, un falso negativo, seria cuando un producto cumple los estandares de calidad pero es rechazado.
- *Falsos Positivos*: Los falsos positivos se dan cuando nuestro clasficador, determina que uno de los datos es correcto cuando en realidad es invalido. Ejemplo: En la misma situacion, un falso positivo, seria cuando un producto no cumple los estandares de calidad y es aceptado.



2.9.1. Precision

La precision es la fraccion entre verdaderos positivos sobre la suma entre verdaderos positivos y falsos positivos. Una forma de decirlo mas claro, es la siguiente: La precision es la fraccion entre la cantidad de predicciones positivas correctas sobre la suma de predicciones positivas correctas e incorrectas.

$$Precision = \frac{VP}{VP + FP} \tag{2}$$

2.9.2. Recall

La exhaustividad (Recall) es la fraccion entre verdaderos positivos sobre la suma entre verdaderos positivos y falsos negativos. Tambien se puede llamar como la fraccion de elementos relevantes que retorna nuestro clasificador.

$$Recall = \frac{VP}{VP + FN} \tag{3}$$

2.9.3. True Positive Rate

Es un sinonimo del Recall.

2.9.4. False Positive Rate

2.9.5. ROC - (Receiver Operating Characteristic)

2.9.6. AUC - Area Under (ROC) Curve

2.9.7. Hiperparametros

Nosotros cuando queremos entrenar un modelo, decimos que existe algo llamado 'parametros', que en nuestro caso seria el set de datos, pero, existe algo por encima de ellos que son llamados 'hiperparametros', que son configuraciones que podemos realizar sobre nuestro modelo modificaran el aprendizaje de nuestro modelo.

3. Aprendizaje Bayesiano

3.1. Aplicado a la clasificacion de Texto

La clasificacion de textos sirve para asignar un topico o una categoria de forma automatica a cualquier extracto de un texto. Lo podriamos utilizar, por ejemplo para:

- Clasificar un email como "spam.º "no spam"
- Identificar al autor de un texto
- Identificar el sexo o edad del autor de un texto
- Identificar el lenguaje en el cual esta escrito un texto
- Realizar trabajo de analisis de sentimiento

Entradas

- Un Documento d
- Un conjunto prefijado de clases $C = \{c_1, c_2, ..., c_j\}$

Salidas

- Una clase c perteneciente al conjunto C

3.2. Metodos de clasificacion de Textos

3.2.1. Reglas escritas a manos

Para la **deteccion del spam**, por ejemplo, podria tener una serie de reglas escritas por una personas que conozca sobre ese topico. **REGLA**: Si el remitente esta en una lista-negra **O** el asunto contiene la palabra: "viagra", entonces se lo considera como spam.

- **Pros**: La precision puede ser muy alta.
- **Contras**: Un recall muy bajo. Construir y mantener las reglas puede ser muy costoso.

3.2.2. Aprendizaje automatico supervisado

Entradas

- un documento d
- un conjunto prefijado de clases $C = \{c_1, c_2, ..., c_j\}$
- un conjunto m de documentos clasfiicados $m = \{(d_1, c_1), ..., (d_n, c_j)\}$

Salidas Un clasificador entrenado $y : d \longrightarrow c$

3.3. Tipos de clasificadores

- Naive Bayes (Bayes ingenuo o bayes simple)
- Logistic Regression (Regression Logistica)
- Support-Vector Machines (Maquinas de Soporte de vectores)
- K-Nearest Neighbors (K-Vecinos mas cercanos)

3.4. Clasificacion de Textos - Naive Bayes

Un enfoque posible para la resolucion del problema de la clasificacion de texto, es encararlo por el lado estadistico, entonces diria que si tengo n documentos y x clases posibles, podria preguntarme:

Cual es la probabilidad de que el docoumento d pertenezca a la clase c?

Parafraseado como la probabilidad condicional: Dado el documento d, Cual es la probabilidad de que pertenezca a la clase c?

$$P(C|D) \overset{\text{Por Teorema de Bayes}}{=} \frac{P(D|C)P(C)}{P(D)} \tag{4}$$

Si tengo un conjunto **C** de clases, segun Bayes un documento **d**, pertenecera a aquella clase que maximice su probabilidad condicional

Entonces, sea

$$C_{map} = argmax P(C|D) \; para \; c \in \; C \tag{5}$$

- **map**: Maximo a posteriori C_{map} es la clase candidata
- **argmax**: Funcion que devuelve el argumento maximo

Entonces,

$$C_{map} = argmax \; P(C|D)P(C), \; c \in C \tag{6}$$

Aunque en el denominador tendria que aparecer $P(D)$, no nos sera necesario para determinar el maximo.

- Como calculamos P(D—C)?
- Como calculamos P(C)?

La mas sencilla de calcular es $P(C)$.

$$P(C) = \frac{cantidad \; de \; documentos \; de \; la \; clase \; \mathbf{c}}{cantidad \; de \; documentos \; totales} \tag{7}$$

Este valor no podemos conocerlo. Pero usando el conjunto de entrenamiento **T**, podemos estimar cual seria esta probabilidad.

$$P^{'} = \frac{cantidad \; de \; documentos \; de \; la \; clase \; \mathbf{c} \; en \; \mathbf{T}}{cantidad \; de \; documentos \; totales \; en \; T} \tag{8}$$

$P(D|C)$ es la probabilidad de que dada una clase **c, d** sea un documento de ella. Esto es un poco mas dificil de identificar. Y para ello tendremos que definir una forma de representar un documento.

Un documento, para Bayes Naive sera una bolsa de caracteristicas/palabras: $x_1, x_2, ..., x_n$. Para nosotros, estas caracteristicas seran las palabras que componen al documento Para ello, asumiremos dos supuestos, muy importantes:

- No importa el orden de las palabras
- Las probabilidades de cada caracteristica, dada una clase c: $P(x_i, c_j)$ son independientes entre si.

Problemas con los supuestos El es una buena persona y no un violento = El es un violento y no una buena persona. Tambien existe problemas si algunas palabras que consideramos caracteristicas, conforman el nombre de un lugar o conforman el sentido de una frase. Por ejemplo:

- Buenos... Aires
- Nueva... York
- Troche y... Moche.

Entonces, si nosotros tokenizamos nuestro documento en un vector de tokens: $d = (x_1, x_2, ..., x_n)$. Por lo que, podemos realizar lo siguiente:

$$P(D|C) = P(x_1, x_2, ..., x_n|C)$$
$$P(x_1, x_2, ..., x_n|C) = P(x_1|c) \cdot P(x_2|C) \cdot ... \cdot P(x_n|C)$$

Entonces

$$C_{map} = P(c_j) \prod P(x_i|c_j)$$
$$c_j \in C \ , \ x \in Posiciones$$

Entonces, como se calcula $P(x_i|c_j)$

3.5. Laplace Smoothing

3.6. Redes Bayesianas

Una forma de modelar el conocimiento de un clasificador de Bayes Naive, es utilizando una red bayesiana.

3.6.1. Red Bayesiana

La Red Bayesiana consiste en tres cosas esenciales

- Un grafo aciclico dirigido
- Los nodos representan variables
- Las aristas representan dependencias condicionales

3.7. Aprendizaje Bayesiano

Si bien las las redes bayesianas permiten inferencias mucho mas precisas que la version simplificada que construye Bayes Naive, son mas complejas de construir y de mantener. Por otro lado, Bayes Naive conjuga varias caracteristicas positivas

- Es muy rapido y poco almacenamiento
- Robusto ante caracteristicas (palabras) irrelevantes
- Muy bueno en dominios en dondem hay muchas caracteristicas y todas son importantes

Ademas, si resulta que el supuesto sobre la independencia de las palabras es cierto. Bayes Naive es optimo.

3.7.1. Comentarios

- Lenguaje natural, es el lenguaje que utilizan las personas.
- Bolsa de palabras, es una forma mas sencilla de modelizar documentos, para que puedan ser entendidas o trabajadas por algoritmos.
- El proceso de filtrar palabras de un texto para Bayes Naive, se conoce como Tokenizacion y cada palabra es un token.

4. Analisis de Sentimientos

El analisis de sentimientos es una tarea de clasificacion de textos. El cual posee multiples usos. Por ejemplo, en el mundo del entretenimiento, existen distintos tipos de opiniones, criticas y comentarios, las cuales pueden llegar a ser positivas o negativas. Por ejemplo:

- Decepcionante $\longrightarrow$ **Negativa**
- Aburrida $\longrightarrow$ **Negativa**
- Personajes memorables y bien desarrollados $\longrightarrow$ **Positiva**
- Una gran puesta en escena que no defraudara $\longrightarrow$ **Negativa**
- Increiblemente predecible $\longrightarrow$ **Negativa**

4.1. Tareas complejas del analisis de sentimientos

4.1.1. Estimar la confianza del consumidor

La confianza del consumidor es un indicador económico que mide el grado de optimismo que los consumidores sienten sobre el estado general de la economía y sobre su situación financiera personal. Qué tan seguras se sienten las personas sobre la estabilidad de sus ingresos determina su **actividades de consumo** y por lo tanto sirve como uno de los indicadores claves en la forma general de la economía.

4.1.2. Predecir el mercado de valores

Google utiliza dos herramientas para este tipo de tareas

- **OpinionFinder** Que mide opinion negativa versus opinion positiva
- **Google-Profile of Mood States** (perfil Google de estados de ánimo) que mide el humor en término de 6 dimensiones: calmado, alerta, seguro, vital, amable y feliz.

Luego, se descubrio que *la calma* podia predecir el indice Dow Jones con 3 dias de anticipacion.

4.1.3. Usos del analisis de sentimientos

- **Peliculas** Si es una critica positiva o negativa.
- **Productos** Indagar que piensa la gente sobre un nuevo telefono.
- **Sentimientos publicos** Como es la confianza del consumir, de que forma crece.
- **Politica** Que piensa la gente sobre cierto candidato o sobre cierta situacion.
- **Prediccion** Predecir el resultado de una eleccion o tendencencia del mercado a partir de los sentimientos.

4.1.4. Tipologia de Scherer de los estados efectivos

- **Emocion:** Respuesta relativamente corta del organismo a estímulos externos. Ejemplos de emoción son la ira, la tristeza, la alegría, el miedo, la vergüenza, el orgullo, la alegría y la desesperación.
- **Estado de animo:** Sentimiento subjetivo de baja intensidad y larga duración. Ejemplos de estados de ánimo : alegre, triste, irritable, apático.
- **Postura interpersonal:** Posición afectiva respecto a otra persona en una interacción específica. Ejemplos de posturas interpersonales son: distante, frío, cálido, de apoyo y de desprecio.

- **Actitudes:** Preferencia o predisposición de una persona respecto a otras personas u objetos. Ejemplos de actitudes son: simpatía, amor, odio, deseo y valoración.
- **Rasgos de Personalidad:** Tendencias en el comportamiento típico de una persona. Ejemplo de rasgos de personalidad: nervioso, ansioso, imprudente, taciturno, hostil, envidioso y celoso.

Cuando se realiza un analisis de sentimientos, lo que en realidad se hace es un analisis de actitudes, es decir, se esta detectando la preferencia o predisposicion de una persona respecto a una persona u otros objetos.

4.1.5. Definicion de Tareas

- El portador de dicha actitud
- El destinatario de dicha actitud
- El tipo de actitud
  - **simpatia, amor, odio, deseo y valoracion**(lista de actitudes candidatas)
  - Una polaridad ponderada: **positiva, neutral y negativa** (y a veces un valor asociado)
- El texto que contiene la actitud (documento, parrafo u oracion)

Resumen

- Una **tarea sencilla** del analisis de sentimientos consiste en **determinar** si la actitud de un texto es positiva o negativa
- Una **tarea mas compleja** del analisis de sentimientos consiste en puntuar la actitud de un texto del 1 al 5.
- Una **tarea avanzada** del analisis de sentimientos consiste en detectar el portador, el destinatario y el tipo de actitud de un texto.

4.2. Algoritmo de Pang y Lee

4.2.1. Deteccion de Polaridad

1. Tokenizacion
2. Extraccion de caracteristicas (palabras, frases, n-gramas)
3. Clasificacion utilizando distintos tipos de algoritmos
   - Naive Bayes
   - MaxEnt
   - SVM

4.2.2. Problemas comunes de la tokenizacion

-

4.2.3. Comentarios

- El índice bursátil Dow Jones es cualquiera de los 130.000 índices bursátiles1 elaborados por la empresa Dow Jones Indexes, LLC, originalmente propiedad de la empresa Dow Jones Company. Por su importancia a veces se denomina índice bursátil Dow Jones al más importante de ellos, cuyo nombre real es Promedio Industrial Dow Jones (DJIA); pero es una forma incorrecta de expresarlo, ya que Dow Jones genera una diversidad de índices. Los índices Dow Jones fueron creados por dos periodistas estadounidenses, Charles Dow y Edward Jones, los cuales fundaron en 1882 la empresa Dow Jones Company.

5. Lexicon de Sentimientos

Un Lexicon, es un diccionario. Con un Lexicon, no es necesario hacer un analisis clasificatorio de palabaras, podemos utilizar el Lexicon directamente.

5.1. Creacion de un Lexicon propio

- Un puniado de ejemplos previamente clasificados
- Algunas reglas escritas a mano que identifique ciertos patrones en una frase

5.1.1. Algoritmo de Hatzivassiloglou y McKeown para la ampliacion de un Lexicon

Abjetivos unidos por z"tienen la misma polaridad

- Justo y legitimo
- Corrupto y brutal

Adjetivos unidos por "pero"tienen distinta polaridad

- Justo pero brutal
- Corrupto pero legitimo
- Hermosa pero malvada

Si bien lograron ampliar considerablemente el Lexicon original, el nuevo lexicon contenia algunos errores, es decir, palabras mal catalogadas. Es por ello que este algoritmo necesariamente necesita de un paso extra que consiste en la revision de los datos obtenidos

5.1.2. Algoritmo de Turney para obtener la polaridad de Frases

- Extrar frases de opiniones/criticas y armar un Lexicon de Frases
- Aprender la polaridad de cada frase
- Puntuas las criticas segundo el promedio de las polaridades de sus frases

6. Aspectos

- Como detectar mas de un sentimiento en la misma frase?
- Que sucede cuando tenemos frases como la siguiente?. La comida era excelente pero el servicio es pesimo!

6.0.1. Metodo de Minqing Hu y Bing Liu

- Frecuencia
- Reglas

Frecuencia: Buscaron todas las **Frases frecuentes** en las criticas de un lugar dado

Por ejemplo: Para un mismo restaurant, encontraron que se repitia muchas veces la frase: **Tacos de pescado**. A este tipo de frases las llamaron: **aspectos, atributos** o bien **objetos de sentimiento** ya que representan el objeto al cual se esta criticando

Reglas: Filtraron todas esas frases frecuentes con algunas reglas como: **Ocurre despues de una palabra que indica sentimientos** Ejemplo: "geniales **Tacos de pescado**" $\longrightarrow$ *indicaque*"*tacosdepescado*"*esmuyprobablementeunaspecto*.**Consideraciones finales**

El aspecto puede no ser mencionado en una sentencia

Para hoteles/restaurantes los aspectos son generalmente conocidos y facilmente de identificar

Es posible una clasificacion supervisada:

- Para pequenios corpus, se puede utilizar una clasificacion manual de aspectos: comida, decoracion, servicio, precio, nada

Y luego entrenar un clasificador:

- Dada una sentencia, tiene alguno de estos aspectos:çomida, decoracion, servicio, precio, nada"

Si la cantidad de criticas no esta balanceada (entre positivas y negativas o entre los rangos elegidos)

- No se puede utilizar el estimador: precision
- Hay que utilizar F-score anteriormente

Si el desbalanceo es muy pronunciado se puede degradar severamente el rendimiento del clasificador. Por lo cual se dan dos soluciones posibles:

- Tomar un muestreo parejo
- Penalizar mas severamente al clasificador por un error al categorizar la clase mas rara

Aspectos de grano grueso, ubnicacion, precio, servicio, comida. Aspectos de grano fino, cosas puntuales, como el nombre del platillo.

7. Extracion de Informacion

El objetivo de la extraccion de informacion es capturar ciertas partes relevantes de un texto. Muchas veces en el contexto de varios documentos distintos, y generar luego, con dicha informacion, una representacion estructurada, limpia y legible, como podria ser una tupla de una base de datos relacional.

7.1. Informacion Factica

Quien hizo que a quien y cuando?

Las oficinas de Google en Argentina ya tienen su historia. La empresa abrio su filial local en 2008 en Puerto Madero. Alli trabajan 215 empleados en los 6000 m2 que ocupan las instalaciones

7.2. Metodos supervisados y auto-supervisados

7.2.1. Supervisados

Requieren un conjunto de datos previamente etiquetados. Requieren tiempo, esfuerzo y una inversion humana importante

7.2.2. Auto-Supervisados

Aprenden a etiquear y generar su propio conjunto de entrenamiento. Son escalables.

7.3. Metodos de Extraccion para la Web(Open Information Extraction)

- No supervisados
- Independientes del Dominio
- Trabajan con grandes cantidades de datos(corpuses)

7.4. Reconomiento de Nombres de Entidades(NER)

En el ejemplo del video, se pueden observar nombres propios. Eso es el reconomiento de nombres de entidades. Los anios pueden ser una entidad porque el numero es quien los identifica. **Utilidades**

- Indices o enlaces a contenidos relacionados
- Destinatarios de los sentimientos en Sentiment Analysis
- Extraccion de Informacion
- Preguntas y Respuestas (question answering)

7.4.1. Modelos de Etiquetamiento secuencial para el reconocimiento de nombre de entidades

Pasos del etiquetamiento

1. Conseguir un conjunto de documentos representativos de nuestro dominio.
2. Etiquetar cada palabra (token) con la clase que le corresponda (persona, organizacion, etc) o bien marcarla con la etiqueta: .ºtra"
3. Especificar caracteristicas de extraccion que se adecuen a las clases y el texto que tenemos
4. Entrenar un clasificador secuencial para predecir las etiquetas del conjunto de prueba

7.4.2. Identifacion de Caracteristicas

Basadas en las palabras

- Palabra Actual
- Palabra Previa o Siguiente

- Substring de una palabra
- Forma de una palabra

Basadas en otro tipo de Inferencia Linguistica

- Etiquetado Gramatical
- Etiqueta Anterior y Siguiente

7.4.3. Algoritmos de Inferencia

- Greedy Inference
- Beam Inference
- Viterbi Inference
- CRFs

7.4.4. Extraccion de Relaciones Semanticas

Ontologia, es una forma de representar el conomiento

- **Is-a(Hiponimo)**: Jirafa es un rumiante es un mamifero es un vertebrado es un animal
- **Instance-of**: Buenos Aires es una instancia de Ciudad

7.4.5. Como construir una ontologia

1. Reglas escritas a mano, de tipo pattern-matching
2. Aprendizaje automatico supervisado
3. Auto-supervisado
4. No supervisado para la web

Reglas escritas a mano, de tipo pattern matching
Reglas Ontologicas Is-A

- Y como X
- Tanto Y como X
- X u otra Y
- X y otra Y

- Y incluyendo X
- Y, especialmente X

Pros de este Metodo:

- Tienden a tener una alta precision
- Puede ser adaptado a dominios especificos

Contras:

- Suelen tener muy bajo recall
- Implica una gran cantidad de trabajo pensar en todos los patrones posibles... mas aun para todas las relaciones
- Se puede mejorar la precision con otros metodos

8. PreProcesamiento y Transformacion de Datos

- Integracion de Datos
- Limpieza de Datos
- Reduccion de Datos
- Transformacion de Datos

8.1. Limpieza de Datos

Datos faltantes
Missing Completely at random MCAR
En este caso la razon de la falta de datos es ajena a los datos mismos. No existen relaciones con la variable misma donde se encuentran los datos faltantes, o con las restantes variables en el dataset que expliquen porque faltan.
Missing Not A Random MNAR
La razon por la cual faltan los datos depende precisamente de los mismo datos que hemos recolectado (esta relacionado con la razon por la que falta)
Ejemplo:
Cada vez que una variable entre 10 y 20, el mismo no se encuentra registrado (independientemente de los valores que tomen las variables restantes)
Missing at Random MAR Punto intermedio entre las dos anteriores La causa de estos datos faltantes no depende de estos mismos datos faltantes, pero puede estar relacionada con otras variables del dataset.
Por ejemplo: Encuestas mal diseniadas

8.2. Estrategias para trabajar con datos faltantes

Eliminar registros o variables
Si la eliminacion de un subconjunto disminuye significativamente la utilidad de los datos, la eliminacion del caso puede no ser efectiva

8.2.1. Imputar datos

Utilizar metodos de relleno de faltantes.
Sustitucion de casos Se reemplaza con valores no observados. Deberia ser realizado por un experto en el area.
Sustitucion por media o mediana Se reemplaza utilizando la media calculada de los valores presentes
Algunas desventajas:

- La varianza estimada de la nueva variable no es valida porque esta atenuada por los valores repetidos
- Se distorsiona la distribucion
- Las correlaciones que se observan estaran deprimidas debido a la repiticion de un valor constante

Imputacion Cold Deck Selecciona valores o usa relaciones obtenidas de fuentes distintas de la base de datos actual
Imputacion Hot Deck Se reemplazan los datos faltantes con valores obtenidos de registros que son los mas similares. (Hay que definir que es similar, K vecinos mas cercanos, por ejemplo)
Imputacion por Regresion El dato faltante es reemplazado con el valor predicho por un modelo de Regresion.
MICE - Multivariate Imputation by Chained Equations
Trabaja bajo el supuesto de que el origen de los faltantes es Missing at Random (MAR)
Es un proceso de imputacion de datos faltantes iterativo, en el cual, en cada iteracion, cada valor faltante de cada variable se predice en funcion de las variables restantes. Esta iteracion se repite hasta que se encuetre convergencia en los valores. Por lo general 10 iteraciones es suficiente **(En cada iteracion se genera un dataset)**

8.3. Analisis de Valores Atipicos

8.3.1. Analisis de Outliers

Un outlier es una observacion que se desvia tanto de las otras observaciones como para despertar sospechas que fue generado por un mecanismo diferente

- Es un concepto subjetivo al programa
- Son observaciones distantes al resto de los datos
- Pueden deberse a un error de medicion, aleatoriedad, que esa instancia pertenezca a una familia distinta al resto, etc.

La deteccion de outliers es importante, su presencia puede influir en los resultados de un analisis estadistico clasico.
Al querer eliminarlos, se debe tener en cuenta algunas cosas:

- Deben ser cuidadosamente eliminadas
- Pueden estar alertando anomalias, en algunas situaciones nuestra tarea de interes sera encontrarlos:
  - Deteccion de Fraudes
  - Deteccion de Fallas
  - Patologias Medicas

8.3.2. Tipos de outliers

A grandes rasgos tenemos 3 grupos.

- Global Outlier, son datos que se alejan del conjunto de puntos.
- Conextual Outlier, dependen del contexto en el cual se hizo la observacion
- Collective Outlier, son conjuntos de observaciones que se comportan en forma anomala colectivamente.

Univariado

- Son valores atipicos que podemos encontrar en una simple variable
- El problema de los enfoques univariados es que son buenos para deteccion de extremos pero en otros casos

Multivariado

- Los valores atipicos se pueden encontrar en un espacio n-dimensional
- Para detectar valores atipicos en espacios n-dimensionales es necesario ajustar un modelo.

En grandes volumenes de datos la deteccion de outliers resulta mas eficiente estudiando todas las variables
Los outliers, en casos multivariados, pueden provocar dos tipos de efectos.

- El efecto **de enmasracamiento** se produce cuando un grupo de outliers esconden a otro/s. Es decir, los outliers enmascarados se haran visibles cuando se elimine/n el o los outliers que los esconden
- El efecto **inundacion** se produce cuando una observacion solo es outlier en presencia de otras observaciones. Si se quitaran las ultimas, la primera dejaria de ser outlier

Metodos para detectar outliers univariados

- IQR: Analizar los valores que estan por fuera del IQR (Analizar Boxplots)
- Z-score y Z-score modificado
- Identificar valores extremos a partir de 1,2 o 3 desvios de la media

Metodos para detectar outliers multivariados

- Analisis de Clustering, utilizando medidas de distancia como Mahalanobis. Los valores similares son agrupados y los que quedan aislados pueden ser considerados outliers.
- Local Outlier Factor (LOF), es un metodo de deteccion de outliers basado en distancias, calcula un score de outlier a partir de una distancia que se normaliza por densidad
- Metodos basados en arboles de busqueda: Isolation Forest

8.4. Metodos Univariados para la deteccion de Outliers

8.4.1. Z-score

Z-score es una metrica que indica cuantas desviaciones estandar tiene una observacion de la media muestral, asumiendo una distribucion gaussiana.

$$z_i = \frac{x_i - \mu}{\sigma} \tag{9}$$

Cuando calculamos Z-score para cada muestra, debemos fijar un umbral: Por ejemplo, la regla de oro es Z ¿3.

8.4.2. Z-score Modificado

La media de la muestra y la desviacion estandar de la muestra, pueden verse afectados por los valores extremos presentes en los datos.

$$M_i = \frac{0{,}6745(x_i - \tilde{x})}{MAD} \tag{10}$$

Donde:

$$MAD = median\{|x_i - \tilde{x}|\} \tag{11}$$

Donde MAD es la mediana de los desvios absolutos respecto de la mediana. Para hacer MAD comparable a la desviacion estandar, se normaliza con 0.6745 Regla de oro: Valores mayores a 3.5 se consideran outliers.

8.4.3. Analisis de BoxPlots

Los boxplots permiten visualizar valores extremos univariados. Las estadisticas de una distribucion univariada se resumen en terminos de cinco cantidades:

- Minimo/Maximo (Bigotes)
- Primer y tercer cuartil(Caja)
- Mediana (Linea media de la caja)
- IQR = Q3 - Q1

Generalmente la regla de decision:

- **+/- 1.5*IQR** Outliers moderados
- **+/- 3*IQR** Outliers severos

8.5. Metodos multivariados para la deteccion de Outliers

8.5.1. Distancia de Mahalanobis

Es una medida de distancia entre el punto y un conjunto de observaciones con media y una matriz de covarianza S.

8.5.2. LOF

El metodo LOF valora los puntos en un conjunto de datos multivariados
Es un metodo basado en la densidad que utiliza la busqueda de vecinos mas cercanos

- Se compara la densidad de cualquier punto de datos con la densidad de sus vecinos
- Parametro K (cantidad de vecinos) y metrica de distancia

El metodo calcula los scores para cada punto, se debe definir un umbral de corte (depende del dominio)
Si el score del punto X es 5, significa que la densidad promedio de los vecinos de X es 5 veces mayor que su densidad local.

8.5.3. Isolation Forest

Es un algoritmo no supervisado y no parametrico basado en arboles de decision.
Idea principal: Los datos anomalos se pueden aislar los datos normales mediante particiones recursivas del conjunto de datos
Algoritmo

1. Tomar una muestra de los datos y construir un arbol de aislamiento
2. Seleccionar aleatoriamente n caracteristicas
3. Dividir los puntos de datos seleccionando aleatoriamente un valor entre el minimo y el maximo de las caracteristicas seleccionadas
4. La particion de observaciones se repite recursivamente hasta que todas las observaciones esten aisladas

Isolation Forest identifica anomalias como las observaciones con longitudes de ruta promedio cortas en los arboles de aislamiento

9. Feature Engineering

Esta etapa incluye cualquier proceso de modificacion de la forma de los datos (es comun que los datos sufran algun tipo de modificacion). El objetivo principal de esta etapa es mejorar el rendimiento de los modelos creados mediante la transformacion de los datos que se utilizan.

9.1. Tecnicas

- Normalizacion
- Discretizacion
- Lograr normalidad
- Imaginacion (Generar nuevas variables)

9.1.1. Normalizacion

Se aplica solamente sobre valores numericos. Consiste en **escalar las features** de manera que puedan ser mapeados en un rango mas small, ya sea entre 0 y 1 o -1 y 1.
Es principalmente utilizada cuando:

- Las unidades dificultan la comparacion entre features

- Se quiere evitar que atributos con mayores magnitudes tengan pesos muy diferentes al resto

Normalizacion: Min-Max
Funciona al ver cuanto mas grande es el valor actual del valor minimo feature y escala esta diferencia por el rango.

$$X^*_{mm} = \frac{X - min(X)}{range(X)} = \frac{X - min(X)}{max(X) - min(X)}$$

Los valores de la normalizacion van entre 0 y 1.
Normalizacion: Z-Score
Los valores de un atriuto se normalizan en base a su media y desvio estandar

$$Z - Score = \frac{X - mean(X)}{sd(X)}$$

Es util cuando el verdadero minimo y maximo del atributo no son conocidos, o cuando hay valores atipicos que dominan la normalizacion min-max.
Normalizacion: Decimal Scalling
Asegurar que cada valor normalizado este entre -1 y 1

$$X_{decimal} = \frac{X}{10^d}$$

Donde **d** representa el numero de digitos en los valores de la variable con el valor absoluto mas grande.

9.2. Transformaciones para lograr normalidad

Podemos reducir este sesgo a partir de transformaciones, por ejemplo:

- Raiz Cuadrada
- Logaritmos
- Inversa de la raiz cuadrada
- Transformaciones de Box-Cox

9.3. Discretizacion

Es una tecnica que permite dividir el rango de una variable continua en intervalos.
Se reducen los valores de una variable continua a un numero reducido de etiquetas.

9.3.1. Discretizacion: Binning

Se divide la variable en un numero especifico de Bins
Los criterios de agrupamiento pueden ser por ejemplo:

- Igual-Frecuencia: Las mismas observaciones en un bin.
- Igual-Ancho: Definimos rangos o intervalos de clases para bin.
- Cuantiles: Separar en intervalos utilizando Mediana, Cuartiles y Percentiles.

A su vez, para cada agrupamiento podemos hacer:

- Reemplazo por media o mediana
- Reemplazo por una etiqueta o valor entero

9.4. Variables Dummies - One Hot Encoding

Algunos metodos analiticos requieren que las variables predictoras sean numericas
Cuando tenemos categoricos, podemos recodificar la variable categorica en una o mas Variables Dummies.

9.5. Creacion de variables nuevas

Por ejemplo, Sumar fuentes de informacion para calcular la distancia desde un inmueble en venta al espacio verde mas cercano.

10. Arboles

10.1. ID3

- ID3: Iterative Dichotomiser 3 (Tree -¿Arbol)
- Genera un arbol de decisión a partir de un conjunto de ejemplos

La salida de un algoritmo ID3 se representa como un grafo en forma de arbol, cuyos componentes son:

- Un nodo de raíz.
- Es aciclico
- Nodos Hoja y Ramas

- Desc 1
- Desc 2
- Desc 3
- Desc 4

10.1.1. ID3 - Entropía de la información

La medida del desorden o la medida de pureza. Básicamente, es la medida de la impureza o la aleatoriedad en los datos.

10.1.2. ID3 - Ganancia de la información

La ganancia de la información se aplica para cuantificar **qué característica**, de un conjunto de datos dados, **proporciona la máxima información** sobre la clasificación.

10.1.3. ID3 - Algoritmo básico

- Calcular la entropía para todas las clases
- Calcular la entropía para cada valor posible de cada atributo
- Selecciona el mejor atributo basado en la reducción de la entropía **usando el cálculo de la ganancia de información**
- Iterar, para cada sub-nodo. Excluyendo el nodo raíz que ya fue utilizado.

10.2. Impureza de Gini

La impureza de Gini es una medida de cuan a menudo un elemento elegido aleatoriamente del conjunto sería etiquetado incorrectamente si fue etiquetado de manera aleatoria de acuerdo a la distribución de las etiquetas del subconjunto. Algunas implementaciones de arboles de decisión utilizan la impureza de Gini en lugar de la ganancia de información, ya que es más fácil de calcular (computacionalmente menos costosa)

10.3. C4.5

Se hicieron mejoras al ID3

- Campos númericos, rangos continuos

- Datos faltantes
- Poda

Datos faltantes:
Manejo de los datos de formación con valores de atributos faltantes - C4.5 permite que los valores de los atributos sean marcados como '?'. Los valores faltantes de los atributos simplemente no se utilizan en los calculos de la ganancia y entropía.
Campos continuos, o rangos continuos
Si un atributo A, tiene un rango continuo de valores, el algoritmo puede crear dinamicamente un campo booleano tal que si A ¡C, Ac=True sino Ac=False.
¿Como encontrar ese umbral C? Vamos a cortar el rango de forma que **C** nos quede con la mayor ganancia de información.

1. Ordenamos A, de menor a mayor.
2. Identificamos los valores adyacentes(de la clase que es nuestra salida)
3. Detectamos cuando hay un cambio de valor de salida, entonces en esos limites seguramente están nuestros C_i candidatos
4. Creamos varios C_i, que dividan en dos el rango. Para cada uno de estos rangos calculamos la ganancia de información. Nos quedaremos con el que mejor resultado de.

Poda
Consiste en:

- Generar el arbol.
- A continuación, analizar recursivamente, y desde las hojas, que preguntas (nodos interiores) se pueden eliminar sin que se incremente el error de clasficación con el conjunto de test.

Si hay ruido en el arbol tendrá un error, es decir cantidad de casos mal clasificados. $Error : \frac{Casos\ bien\ clasificados}{Casos\ totales}$
El método de poda consiste en:

1. Se elimina un nodo interior cuyos sucesores son todos nodos hoja
2. Se vuelve a calcular el error que se comete con este nuevo arb ol sobre el conjunto de test
3. Si este error es menor que el anterior, entonces se elimina el nodo y todos sus sucesores
4. Se vuelve al paso 1.

10.4. Random Forest

La idea general de Random Forest es "Muchos estimadores mediocres, promediados pueden ser muy buenos"

10.4.1. Bootstrap Aggretating

Es una técnica, o meta-algoritmo que dice lo siguiente: Dado un conjunto de entrenamiento D, de tamaño n, la técnica de **bagging** generará **m** nuevos conjuntos de entrenamiento D1,...Di,...,Dm cada uno de tamaño n' tomando muestras aleatorias de D. Y en general n' ¡n. Siendo n' aproximadamente 2/3 de n. Además de eso, en el proceso de los nuevos conjuntos generados, se tomarán solo algunos atributos de forma aleatoria también. Para saber cuantos atributos tendremos, tenemos que aplicar la raíz cuadrada del total de atributos.

11. Ensambles

- Entrenar varios modelos, cada uno sobre datos distintos
- Cada modelo sobre-ajusta de manera diferente
  - Cada modelo: bajo sesgo, alta varianza.
  - Por ejemplo: Arboles profundos

11.1. Bias

El error debido al Bias de un modelo es simplemente la diferencia entre el valor esperado del estimador (es decir, la prediccion media del modelo) y el valor real.
Cuando se dice que un modelo tiene un bias muy alto quiere decir que el **modelo es muy simple y no se ha ajustado a los datos de entrenamiento** (suele ser **underfitting**, por lo que produce un error alto en todas las muestras: entranamiento, validacion y test.

11.2. Variance

La varianza de un estimador es cuanto varia la prediccion segun los datos que utilicemos para el entrenamiento.
Un modelo con **varianza baja indica que cambiar los datos de entranamiento produce cambios pequenos en la estimacion**
Al contrario, un modelo con **varianza alta quiere decir que pequenios cambios en el dataset conlleva a grandes cambios en el output** (suele ser **overfitting**)

11.3. Como utilizamos los modelos mediocres

- **Votacion:** Para una nueva instancia, clasificarla con todos los modelos y devolver la clase mas elegida. La votacion reduce la varianza de la clasificacion. Si los modelos individuales devuelven una probabilidad, se puede hacer una probabilidad ponderadad

11.4. Bagging

Tenemos un dataset inicial y realizamos N particiones, y para cada uno de estos conjuntos, entrenamos un clasificador, por lo cual, tendremos N salidas y a partir de estas salidas se realiza una votacion para producir una salida final.
Es una tecnica que consiste en construir nuevos conjuntos de entrenamiento usando **bootstrap**(muestras aleatorias con reemplazo) para entrenar distintos modelos, y luego ordenarlos. **Algoritmo de Bagging**

1. Dividimos el conjunto de entrenamiento en distintos subconjuntos, obteniendo como resultado diferentes muestras aleatorias
   - Las muestras son uniformes (misma cantidad de individuos)
   - Son muestras con reemplazo (los individuos pueden repetirse en el mismo conjunto de datos)
2. Entrenamos un modelo con cada subconjunto
3. Construimos un unico modelo predictivo a partir de los anteriores

Caracteristicas

- 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 el voto ponderado

Problemas para usarlo con arboles

- Si pocos atributos son predictores fuertes, todos los arboles se van a aparecer entre si
- Esos atributos terminaran cerca de la raiz, para todos los conjuntos generados con **bootstrap**

11.5. Bagging: Random Forest

Igual a **bagging** tradicional, pero en cada nodo considerar solo un subconjunto de **m** atributos elegidos al azar.

11.6. Boosting

Entrenar modelos de forma secuencial, es una alternativa a Bagging, que entra modelos de forma paralela. Lo que hace es buscar modelos para las instacias mal clasificadas por los anteriores. **Algoritmo Boosting**

- Comenzar con un modelo (simple) entrenado sobre todos los datos: h0.
- En cada iteracion i, entrenar hi dando mayor importancia a los datos mal clasificados por las iteraciones anteriores
- Terminar al conseguir cierto cubrimiento, o luego de un numero de iteraciones.
- Clasificar nuevas instancias usando una votacion ponderada de todos los modelos construidos.

11.6.1. Modelos exitoso

- Adaboost
- Gradient boosting
- XGBoost: eXtreme Gradient Boosting

XGBoost vs Gradient Boosting

- La velocidad de XGBoost es mucho menor gracias a su implementacion y a estar mejor orientado al uso eficiente del hardware (GPU)
- El **Accuracy** tambien es mejor debido a que XGBoost maneja mejor el **overfitting** mediante regularizaciones

Resumen

- Necesita pesos
  - Debemos adaptar el algoritmo de aprendizaje
  - Y tomar muestras con reemplazo segun pesos
- Puede sobreajustar

11.6.2. AdaBoost

La técnica detrás de AdaBoost consiste en entrenar un predictor, un clasificador base (por ejemplo un árbol de decisión), verificar los errores que comete y entrenar luego otro predictor que corrija estos errores, (estas instancias mal clasificadas). AdaBoost repite este proceso hasta disminuir el error o encontrar un clasificador perfecto.
Utiliza para verificar el error el mismo conjunto de entrenamiento. Solo que, antes de entrenar el siguiente predictor, pondera las instancias mal clasificadas, aumentando su peso relativo. De esta manera el siguiente predictor entrenado estará focalizado en corregir los errores del primero.
Contra: El entrenamiento de AdaBoost no puede hacerse en paralelo y por lo tanto es poco escalable.
La implementacion mas comun es con arboles.
A diferencia de **Random forest**, donde tenemos N arboles completos (de distinta profundidad), en AdaBoost tenemos un bosque de Tocones (Stumps) **AdaBoost resumen**

- **AdaBoost** combina un monton de **weak learners** para hacer clasificaciones. Estos **Weak Learners**, son generalmente **stumps**
- Algunos stumps tienen mas pesos que otros en la votacion final, tienen mas que decir (amount of say)
- Cada uno de estos tocones esta construido teniendo en cuenta los errores del tocon anterior.
    - Lo podemos hacer usando una funcion de impureza de Gini ponderada, para cada ejemplo
    - O simplemente regenerando los datos como vimos en este ejemplo

11.6.3. Gradient Boost

Es un modelo utilizado en regresion tanto como clasificacion.

- Gradient Boost crea una cadena de arboles con una profundidad fija.
- Comienza con un solo valor, un nodo hoja. Y luego calcula arboles para calcular el error cometido por el anterior
- Cada arbol esta ponderado por un factor constante llamado **learning rate** o tasa de aprendizaje
    - Los arboles estan restringidos en su crecimiento

11.6.4. XGBoost

XGBoost fue diseñado para Big Data, es decir para conjuntos de datos grandes y complejos. Sin embargo a fines de entender el algoritmo principal lo usaremos con un conjunto de datos simple (y para el caso de regresión).

$$Similarity\ Score\ = \frac{(Suma\ de\ Residuos)^2}{Cantidad\ de\ Residuos + \lambda} \tag{12}$$

Regularizacion Imaginemos que tenemos una serie de datos y queremos encontrar una forma de predecir valores futuros. En este caso utilizariamos regresion lineal. Pero que pasa si tenemos solo dos puntos?
La linea ajusta perfecto. No hay residuos (errores). Pero el problema se da cuando utilicemos el conjunto de prueba y tendremos un problema de alta varianza dado que el residuo de los datos de prueba sera enorme. Una forma de solucionar este problema es utilizando una variacion de la regresion lineal llamada: **Ridge Regression**
La idea es encontrar una linea que no ajuste tan bien. Para ello se agrega un bias pequenio en los datos de entrenamiento.
Cuando usamos **Regresion Lineal**, es decir, cuadrados minimos, lo que estamos haciendo es minimizando **la suma de los cuadrado de los residuos**. En cambio, **Ridge Regression** estamos minimizando:
La suma de los cuadrados de los residuos + $\lambda * (pendiente)^2$ Donde:

- λ Determina que tan severa es esta penalizacion

11.6.5. Ensambles Hibridos

Completar

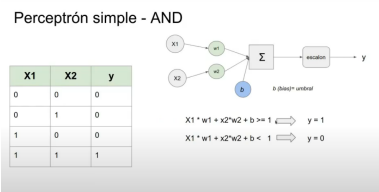
12. Redes Neuronales Artificiales

Se las conoce como Redes Neuronales Artificiales porque las redes neuronales son las que poseen los seres vivos y la idea detras de esto es replicar las redes neuronales del ser vivo. Cabe destacar que las redes neuronales son algoritmos supervisados, por lo cual necesitan de un conjunto de entrenamiento. Aunque existen algunas redes neuronales que son no supervisadas.

12.1. Perceptron

Un perceptron es una neurona artificial.Tenemos entradas X,Y,Z externas, donde cada una de ellas estara siendo multiplicada por un peso. Nosotros utilizaremos la funcion **Funcion escalón unitario**, que es una funcion escalon centrada en 0.

12.1.1. Perceptron Simple - AND



Necesitamos saber cuanto valen w_1, w_2 **y b(bias)**.
Se pueden setear los pesos de forma aleatoria inicialmente, pero encontrar los pesos correctos, es lo que se llama proceso de entrenamiento. **PROBLEMA LINEALMENTE SEPARABLE**

12.1.2. Perceptron Simple - Almacenamiento

Cuando el programa guarda los datos de la red neuronal en memoria, lo que realmente es hace es guardar en una matriz SOLAMENTE los pesos. Un modelo entrenado no es mas que una matriz de numeros flotantes.

12.2. Redes SOM(Kohonen)

Mapean un espacio de entrada a traves de los pesos en una capa de salida, que tiene varias neuronas de ancho y de alto. Es un algoritmo no supervisado, por lo tanto, lo que se hace es armar cluster, o sea, conjuntos de datos que estan relacionados por sus caracteristicas intrinsecas. Tenemos dos capas generalmente, la capa de entrada y las capas de salida que representan una matriz.

12.3. Backpropagation

Es el algoritmo que se utiliza para entrenar redes neuronales.

12.4. Implementacion de Redes Neuronales

12.4.1. Funciones de activacion a utilizar en la ultima capa

- Regresion -¿Sigmoidea, Lineal, ReLu, etc.
- Clasificacion de Clases excluyentes -¿Sigmoidea
- Clasificacion de N clases simultaneas -¿Softmax

Softmax
Funcion exponencial normalizada. Se utiliza como funcion de activacion de la capa de salida en modelos de clasificacion, interpretandola como **scoring**, segun el modelo, de pertenecer a dicha clase

12.4.2. Redes neuronales muy complejas

Si tenemos redes neuronales muy complejas, podriamos llegar un caso de overfitting, para solucionar esto, tenemos los siguientes metodos de regularizacion. Donde un metodo de regularizacion, ayuda a que el modelo funcione mejor con datos que nunca vio.

- Regularizacion L1 y L2
- Dropout
- Early Stopping
- Data augmentation

Regularizacion L1 y L2
Penalizacion el valor de los pesos de la red. Esto evita que se le de mas relevancia a una caracteristica que a otra. Se le agrega un termino en la funcion de costos proporcional a los pesos.
Si es proporcional al modulo se llama **Regularizacion L1**, si es proporcional al modulo cuadrado se le llama **Regularizacion L2**.
Dropout
Metodo de regularizacion que evita codependencias en las conexiones de la red. La idea es .ªpagar.ªctivaciones aleatoriamente durante el entrenamiento. Esto hace que el buen funcionamiento de la red no dependa de unas pocas neuronas. **Early Stopping**
La idea es evitar el sobreajuste parando el entrenamiento antes de que el error del set de validacion empieza a aumentar. Este metodo busca entonces quedarse con los pesos en la instancia optima.
Data Augmentation
La idea es agregar datos usando los datos que se tienen y aplicarles transformaciones que los conviertan en nuevos datos, de manera que sean verosimiles. Es especialmente util cuando se trabaja con imagenes. En general es dificil encontrar las transformaciones a aplicar.
Optimizadores Un optimizador es una implementacion concreta de un algoritmo de **Backpropagation**. Los optimizadores mas usados son:

- SGD: Stochastic gradient descent
- Momentum
- Nesterov
- RMSprop
- AdaGrad
- Adam

- Nadam

SGD
Backpropagation simple, sin ningun tipo de optimizacion, tal y como lo vimos en clase. Algoritmo de la decada de 1960.
Momentum
Propuesto por Boris Polyak en 1964. La idea principal: Imaginemos una pelota rodando por una colina, ira acelerando hasta que llegue a una velocidad constante debido al friccion y seguira su camino. A diferncia del **backpropagation** original en donde los pasos son regulares aqui son cada vez mas rapidos. El gradiente se utiliza para aceleracion y no para la velocidad
Nesterov
Propuesta por Yurii Nesterov en 1983. Variante de Momentum, en vez de calcular el gradiente del error en el punto actual, lo calcula un poco mas adelante (en la direccion del momento)
AdaGrad Presentado por John Duchi y otros en 2011. Uno de los problemas de los otros metodos es que en N dimensiones, el error descendera por la dimension con la pendiente mas empinada, que no necesariamente sera la que conduzca al minimo global. **AdaGrad** reduce el vector gradiente a lo largo de las dimensiones mas empinadas. En otras palabras, cuanto mas alto es el gradiente, mas empinada es la funcion de error en esa dimension. Ya que el gradiente mide "la pendiente".
Al dividir esa pendiente, por un valor proporcional a ella misma, se suaviza, y en todas las dimensiones el descenso por gradiente es similar, evitando caer en valles locales.
Pros:
Con frecuencia **AdaGrad** tiene un buen desempenio para problemas cuadraticos simples. Es bueno para tareas sencillas como regresion lineal.
Contras:
A menudo se detiene demasiado pronto cuando entrena redes neuronales. Muchas veces se detiene antes de alcanzar el minimo global. No deberia usarse para entrenar redes profundas.
RMSProp Creado por Geoffrey Hinton y Tijmen Tieleman en 2012 Soluciona el principal problema de AdaGrad al ir .ºlvidando"las pendientes anteriores, a medida que sigue avanzando. Es decir, solo acumula los gradientes de las iteraciones mas recientes. β es la tasa de decaimiento
Adam
Adam: Adaptative moment estimation. Fue presentado en 2014 por Diederik P.Kingma y Jimmy Ba. Combina las ideas de Momentum y RMSProp. Hace un seguimiento de una media de decaimiento exponencial de gradientes pasados y de gradientes cuadrados pasados.
AdaMax
AdaMax: Modificacion de Adam. En general Adam da mejores resultados, pero depende del conjunto de datos.
Nadam

Es Adam + Nesterov, asi que a menudo converge mas rapido que Adam
AdaDelta
Es una variacion de AdaGrad en la que en vez de calcular el escalado del factor de entrenamiento de cada dimension, teniendo en cuenta el gradiente acumulado desde el principio de la ejecucion, se restringe a una ventana de tamanio fijo de los ultimos n gradientes. Similar a RMSProp que va olvidando los gradientes

12.4.3. Numero de Capas

Para muchos problemas 1 oculta sera suficiente. En teoria un PMC con una sola capa oculta puede modelizar funciones complejas. Tendra que tener neuronas suficientes. Pero si estamos ante problemas mas complejos, las redes profundas tendran mejor desempenio, ya que pueden modelizar mejor con menos neuronas totales. Problemas como reconocimiento de imagenes o del discurso requieren decenas o cientos de capas, pero todas ellas conectadas como en PMC.

12.4.4. Numero de neuronas por capa

El numero de neuronas por capa de entrada y de salida esta determinado por el problema a resolver. Cada numero es una imagen de 28x28 por pixel = 784 neuronas de entrada. Los digitos a reconocer, son los del sistema decimal tradicional. Asi que son 10, del 0 al 9. 10 neuronas de salida. Lo habitual es hacer una piramide. Poniendo cada vez menos neuronas. Por ejemplo para MNIST, 3 capas ocultas podrian tener: 300, 200 y 100 neuronas cada una. Sin Embargo, ultimamente, se ha cuestionado esta tecnica, ya que a veces poner la misma cantidad de neuronas en todas da el mismo resultado o a veces mejor.

12.4.5. Hiperparametros

- Learning rate, es el mas importante, indica que tan rapido se va descendiendo en la funcion de costo. Valores entre E-01 a E-04.
- La cantidad de epocas(epochs) depende...

12.4.6. Entrenamiento de la red

Una vez que tenemos los ingredientes anteriores, podemos entrenar la red. (Arquitectura + Hiperparametros + Optimizador + Funcion de perdida + Funciones de activacion)

12.5. Notas sobre la practica de Implementacion de Redes Neuronales

13. Introduccion a las Redes de Aprendizaje Profundo

No hay una clara diferencia entre las redes neuronales superficiales y las redes neuronales profundas. Aca una lista de las redes profundas mas conocidas

- Restricted Boltzmann Machine
- Autoencoder
- Deep Belief Network
- Convulotional Net
- Recurrent Net
- Recursive Neural Tensor Nets (RNTN)

Lo que hacemos, es segmentar nuestro problema a analizar en capas, para que las distintas capas ataquen diferentes aspectos.

13.1. Usos de las redes de aprendizaje profundo

- Procesamiento de texto, donde se utiliza Recurrent Net (A nivel de caracter), RNTN, Convolutional Net (para analisis de sentimientos)
- Reconocimiento de imagenes, con Deep Belief Network, Convolutional Net
- Reconocimiento de Objetos, Convolutional Net, Recursive Neural Tensor Nets.
- Reconocimiento del Habla, Recurrent Net
- Clasificacion con Deep Belief Network y Perceptrones Multicapa con RELU
- Analisis de Series de Tiempo -¿Recurrent Net

13.2. Problemas del Entrenamiento de una Red Profunda

El entrenamiento de una red de aprendizaje profundo puede tardar meses, pero con Modernos GPU's puede llegar a tardar un dia. Una de las razones por las que las redes de aprendizaje profundo son bastante recientes, es por la falta de computo que esta casi resuelta con las GPU's. Tambien el **Desvanecimiento del Gradient**, es mucho mas lento con el deep learning, dado que en las ultimas capas, los pesos ni se actualizaban.

13.2.1. Restricted Boltzman Machine

Geoff Hilton, fue el primero en encontrar una solucion al problema de las redes neuronales de aprendizaje profundo. Es considerado el Padre del **Deep Learning**

1. Ejecutar una entrada en sentido directo
2. Hacer una ejecucion en sentido inverso
3. Comparar con **KL Divergence** y ajustar pesos y bias, hasta que las salidas de la red invertida coinciden con las de entradas, o se acercan lo mas posible.

13.2.2. Deep Belief Nets

Una red Deep Belief Nets, es exactamente igual a un **Perceptron Multicapa**, pero su metodo de entrenamiento es completamente diferente.

1. Capa capa aprende el input entero (Distinto a las redes convolucionales)
2. Cuando el entrenamiento concluye, la red aprendio a detectar patrones inherentes en los datos, pero aun no sabemos nada de esos patrones
3. Asi que hay una segunda parte de entrenamiento supervisado pero con una cantidad pequenia de datos para clasificacion

13.2.3. Autoencoders

- RBM Y DBN son autoencoders
- Un autoencoder aprende a producir una salida exactamente la misma informacion que recibe de entrada
- Entrada y salida poseen igual numero de neuronas

Usos:

- Compresores
- Reduccion de Dimensionalidad
- Eliminacion de Ruido

Entrenamiento: Se pueden entrenar con **BackPropagation**, pero utilizando una metrica particular llamada "**loss**"(Cantidad de informacion que perdio al tratar de reconstruir el input)

13.2.4. Convultional Neural Nets

- Dominan completamente la vision espacial
- Desarrolladas por **Yann Lecun** of New York University. Tambien es el director del departamento de **AI de Facebook**
- Despues que aunaron esfuerzos en 2015, Microsoft, Google y Baidu lograron que una computadora derrote a un humano en un concurso de reconomiento visual de objetos. La primera vez en toda la historia de IA.

Que es la convolucion?
Es una manera de combinar dos funciones en una nueva.

13.2.5. Redes Recurrentes

Son utiles cuando los patrones en los datos cambian con el tiempo. Estas redes existen hace mucho tiempo pero han ganado popularidad recientamente gracias a **Schmidhuber**, **Hochreiter** y **Graves** Pueden recibir una secuencia de valores como entrada y pueden tambien devolver una secuencia de valores de salida. Las redes recurrentes poseen un problema de entrenamiento, que es justamente el desvanecimiento del gradiente. 1 sola capa con 100 pasos temporales es como entrenar una red de 100 capas de alimentacion hacia delante.

13.2.6. Redes recurrentes Vs. Redes hacia delante

- Red hacia delante: Clasificacion o Regresion
- Red recurrente: Serie, prediccion, pronostico

13.2.7. Redes Neuronales de Tensores Recursivas

- Creadas por **Richard Socher** de MetalMind
- Para realizar tareas de analisis de sentimientos, tiene en cuenta el orden y la agrupacion sintactica
- Tienen estructura de arbol
- Neuronas Raiz
- Neuronas Hojas