03/03/23 – Scikit-Learn Profesional

|  |  |
| --- | --- |
| Puntos clave | Notes |
| Los datos son todo: si tenemos correctamente analizados podemos llegar a modelos más efectivos.  Aprendizaje supervisado: por observación  Aprendizaje no supervisado: Descubrimiento.  Intentamos descubrir patrones o relaciones.  Aprendizaje por refuerzo: Prueba y error.  Opciones a ML: Algoritmos evolutivos(optimización), Lógica difusa(variables continuas), Agentes, Sistemas expertos(Sistema de reglas para responder preguntas concretas ) | **Como aprenden las maquinas**?  **Aprendizaje supervisado** ➜ si de los datos se puede extraer con anticipación información precisa del resultado que esperamos.  **Aprendizaje por refuerzo** ➜ Si de los datos no se puede sacar exactamente la información que queremos predecir, pero si podemos dejar que el modelo tome decisiones y evalué si estas decisiones son buenas o malas.  **Aprendizaje no supervisado** ➜ cuando no se tiene ninguna información adicional sobre lo que esperamos, sino que los datos por sí solos nos van a revelar información relevante sobre su propia naturaleza y estructura.  **Algoritmos evolutivos**: Realmente es una serie de algoritmos heurísticos, en donde en tu espacio de soluciones se explora los mejores candidatos, según se optimice cierta función de costo. Por ejemplo, se usa en la industria automotriz o de diseño aeroespacial para encontrar el mejor diseño que minimice, por ejemplo, la resistencia al aire.  **Lógica Difusa**: Es una generalización de la lógica clásica, pero en lugar de tener solo dos condiciones (verdadero, falso) [principio de tercero excluido], se tienen condiciones de verdad continuas. Por ejemplo, si 1 representa verdadero, y 0 representa falso, en la lógica difusa, el grado de verdad ahora puede tomar valores en el intervalo continuo de [0, 1]. Este enfoque tenía mucho auge en sistemas de control y robótica, antes del auge de las redes neuronales.  **Agentes**: Para sistemas cuyas propiedades se puede describir por la interacción de agentes, se utiliza este enfoque para encontrar o describir comportamientos en el colectivo. (Por ejemplo, los mercados financieros compuestos por agentes económicos, vendedores y compradores, etc.). La física estadística y los sistemas complejos también se ayudan de este enfoque.  Realmente son muy interesantes y divertidos cada enfoque. Te invito a investigar más al respecto.  **Problemas que podemos resolver con Scikit-learn**  Algunas limitaciones de Scikit-learn  1- No es una herramienta de Computer Vision.  2.- No se puede correr en GPUs. |
| Sumario: | |

1/4/23 - **Problemas que podemos resolver con Scikit-learn**

|  |  |
| --- | --- |
| Puntos clave | Notes |
| una variable discreta es una variable numérica que solo puede tomar valores enteros, por ejemplo la cantidad de personas en una habitación.  El tema es que por detrás, las variables categóricas se deben convertir en números para poder trabajar con scikit-learn por lo que se asignan valores enteros a cada categoría, convirtiéndolos en variables discretas. | 3.- No es una herramienta de estadística avanzada.  4.- No es muy flexible en temas de Deep Learning.  **Qué problemas podemos abordar con Scikit-learn?**  **Clasificaciones**: Necesitamos etiquetar nuestros datos para que encajen en alguna de ciertas categorías previamente definidas.  Ejemplos: ¿Es cáncer o no es cáncer?, ¿La imagen pertenece a un Ave, Perro o Gato?, ¿A qué segmento de clientes pertenece determinado usuario?  **Regresión**: Cuando necesitamos modelar el comportamiento de una variable continua, dadas otras variables correlaciones.  Ejemplos: Predecir el precio del dólar para el mes siguiente, el total de calorías de una comida dados sus ingredientes, la ubicación más probable de determinado objeto dentro de una imagen.  **Clustering**: Queremos descubrir subconjuntos de datos similares dentro del dataset. Queremos encontrar valores que se salen del comportamiento global.  Ejemplo: Identificar productos similares para un sistema de recomendación, descubrir el sitio ideal para ubicar paradas de buses según la densidad poblacional, segmentar imágenes según patrones de colores y geometrías.  **Importancia de las mates en el ML**  **La cortina de fondo**: Varias técnicas que usamos para que los computadores aprendan están inspiradas en el mundo natural.   * **Redes neuronales artificiales**: Están inspiradas en el cerebro humano. * **Aprendizaje por refuerzo**: Está inspirado en las teorías de la psicología conductual. * **Algoritmos evolutivos**: Las teorías de Charles Darwin.   **Temas matemáticos generales para repasar**:   * Funciones y trigonométrica. * Algebra lineal. * Optimización de funciones. * Calculo diferencial. |
| Sumario: | |

03/03/23

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | Temas de probabilidad a repasar  Probabilidad básica.  Combinaciones y permutaciones.  Variables aleatorias y distribuciones.  Teorema de Bayes.  Pruebas estadísticas.  **Datasets que usaremos**:  **World Happiness Report**: Es un dataset que desde el 2012 recolecta variables sobre diferentes países y las relaciona con el nivel de felicidad de sus habitantes. Nota: Este data set lo vamos a utilizar para temas de regresiones  **The Ultimate Halloween Candy Power Ranking**: Es un estudio online de 269 mil votos de más de 8371 IPs deferentes. Para 85 tipos de dulces diferentes se evaluaron tanto características del dulce como la opinión y satisfacción para generar comparaciones. Nota: Este dataset lo vamos a utilizar para temas de clustering  **Heart disease prediction**: Es un subconjunto de variables de un estudio que realizado en 1988 en diferentes regiones del planeta para predecir el riesgo a sufrir una enfermedad relacionada con el corazón. Nota: Este data set lo vamos a utilizar para temas de clasificación.  **¿Cómo afectan nuestros features a los modelos de Machine Learning?**  **¿Qué son los features?**   * Son los atributos de nuestro modelo que usamos para realizar una interferencia o predicción. Son las variables de entrada.   **Más features siempre es mejor, ¿verdad?**   * La respuesta corta es: NO   **En realidad si tenemos variables que son irrelevantes pasarán estas cosas**:   * Se le abrirá el paso al ruido. * Aumentará el costo computacional. * Si introducimos demasiados features y estos tienen valores faltantes, se harán sesgos muy significativos y vamos a perder esa capacidad de predicción. |
| Sumario: | |

03/03/23 -

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **Nota**: Hacer una buena selección de nuestro features, hará que nuestros algoritmos corran de una manera más eficiente.  **Una de las formas de saber que nuestros features han sido bien seleccionados es con el sesgo y la varianza**   * Una mala selección de nuestro features nos puede llevar a alguno de esos dos escenarios indeseados.     **Algo que debemos que recordar es que nuestro modelo de ML puede caer en uno de 2 escenarios que debemos evitar**:   1. Uno es el **Underfitting**: Significa que nuestro modelo es demasiado simple, en donde nuestro modelo no está captando los features y nuestra variable de salida, por lo cual debemos de investigar variables con más significado o combinaciones o transformaciones para poder llegar a nuestra variable de salida. 2. Por otro lado, está el **Overfitting**: Significa que nuestro modelo es demasiado complejo y nuestro algoritmo va a intentar ajustarse a los datos que tenemos, pero no se va a comportar bien con los datos del mundo real. Si tenemos **overfitting** lo mejor es intentar seleccionar los features de una manera más critica descartando aquellos que no aporten información o combinando algunos quedándonos con la información que verdaderamente importa. |
| Sumario: | |

03/04/23 - **Introducción al algoritmo PCA (Principal Component Analysis)**

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **¿Qué podemos hacer para solucionar estos problemas?**   * **Aplicar técnicas reducción de la dimensionalidad**: Utilizaremos el algoritmo de PCA. * **Aplicar la técnica de la regulación**: que consiste en penalizar aquellos features que no le estén aportando o que le estén restando información a nuestro modelo. * **Balanceo**: Se utilizará **Oversampling y Undersampling** en problemas de rendimiento donde tengamos un conjunto de datos que está desbalanceado. por ejemplo: en un problema de clasificación donde tenemos muchos ejemplos de una categoría y muy pocos de otra.   **¿PCA - Por qué usaríamos este algoritmo?**  Porque en machine learning es normal encontrarnos con problemas donde tengamos una enorme cantidad de features en donde hay relaciones complejas entre ellos y con la variable que queremos predecir.  **Pistas donde se puede utilizar un algoritmo PCA**:   * Nuestro dataset tiene un número alto de features y no todos sean significativos. * Hay una alta correlación entre los features. * Cuando hay overfiting. * Cuando implica un alto coste computacional. |
| Sumario: | |

03/04/23

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **¿En qué consiste el algoritmo PCA?**  **Básicamente en reducir la complejidad del problema**:  1.- Seleccionando solamente las variables relevantes.  2.- Combinándolas en nuevas variables que mantengan la información más importante (varianza de los features).    **¿Cuáles son pasos para llevar a cabo el algoritmo PCA?**   1. Calculamos la matriz de covarianza para expresar las relaciones entre nuestro features. 2. Hallamos los vectores y valores propios de esta matriz, para medir la fuerza y variabilidad de estas relaciones. 3. Ordenamos y escogemos los vectores propios con mayor variabilidad, esto es, aportan más información.   **¿Qué hacer si tenemos una PC de bajos recursos?**   * Si tenemos un dataset demasiado exigente, podemos usar una variación como IPCA. * Si nuestros datos no tienen una estructura separable linealmente, y encontramos un KERNEL que pueda mapearlos podemos usar KPCA. |
| Sumario: | |

03/04/23 - **Preparación de datos para PCA e IPCA**

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **¿Qué es normalizar los datos en ML?**  Para que funcionen mejor muchos algoritmos de Machine Learning usados en Data Science, hay que normalizar las variables de entrada al algoritmo.  **Normalizar** significa, en este caso, comprimir o extender los valores de la variable para que estén en un rango definido. Sin embargo, una mala aplicación de la normalización, o una elección descuidada del método de normalización puede arruinar tus datos, y con ello tu análisis.  Ejemplo visto en clase:  **Escalado estándar (Standard Scaler)**  Una alternativa al escalado de variables es usar otra técnica conocida como escalado estándar (a cada dato se le resta la media de la variable y se le divide por la desviación típica).    Este método funcionaría para normalizar la señal de la fibra óptica del ejemplo anterior, conservando su forma, pero  ¿qué pasará con otras señales?.  Los dos estadísticos que se usan (media y desviación típica) son muy sensibles a valores anómalos (muy grandes o pequeños con respecto al resto).  Imaginemos otro ejemplo; Vamos a medir cuánto se usa la palabra “resaca” en publicaciones de Facebook (datos reales). La frecuencia de uso de esta palabra tiene picos durante el fin de semana y valles entre semana. Los datos tienen valores anormalmente altos en fiestas como Halloween y Navidad. |
| Sumario:  **Un pequeño recordatorio**: Vamos a recordar que estamos estamos trabajando bajo el dataset de pacientes con riesgo a padecer una enfermedad cardiaca, con este dataset pretendemos que utilizando ciertas variables de los pacientes, por ejemplo su edad, su sexo, su presión sanguínea y un índice de dolor que pueden sentir al realizar un ejercicio física.  Vamos a intentar hacer una clasificación binaria, entre si el paciente tiene una enfermedad cardiaca o no la tiene, el objetivo de esta clase es hacer una clasificación básica, pero que nos dé una información relevante, maximizando la información de todos estos features. | |

03/04/23 - Implementación del algoritmo PCA e IPCA

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | Antes de normalizar, calculamos la media (5.55) y la desviación típica (10.53). Ya podemos ver que la media está en torno a 5, cuando nuestros datos sin anomalias no pasan de valores en torno al 4 (mala señal). Si aplicamos ahora la normalización estándar, tenemos lo siguiente.    Lo primero que vemos es que no hemos conseguido normalizar entre 0-1 con este método. Además ahora tenemos valores negativos, cuando antes no los teníamos. Por si esto fuera poco, nuestros valores pico y valle han quedado muy atenuados por culpa de las anomalías. Una solución a esto sería eliminar las anomalías antes de normalizar (tema para otro post).  Análisis similares se puede hacer para otros métodos de normalización: escalar sobre máximo, **normalizer**, escalado robusto, etc.  **Implementación del algoritmo PCA e IPCA**  **Un pequeño recordatorio**: Vamos a recordar que estamos trabajando bajo el dataset de pacientes con riesgo a padecer una enfermedad cardiaca, con este dataset pretendemos que utilizando ciertas variables de los pacientes, por ejemplo su edad, su sexo, su presión sanguínea y un índice de dolor que pueden sentir al realizar un ejercicio física.  Vamos a intentar hacer una clasificación binaria, entre si el paciente tiene una enfermedad cardiaca o no la tiene, el objetivo de esta clase es hacer una clasificación básica, pero que nos dé una información relevante, maximizando la información de todos estos features. |
| Sumario: **Implementación del algoritmo PCA e IPCA**   * El rendimiento de los dos algoritmos es casi exactamente el mismo, pero hay que considerar que nuestro dataset tenía 13 features originalmente para intentar predecir una clasificación binaria y utilizando PCA. * Solo tuvimos que utilizar 3 features artificiales que fueron los que nos devolvió PCA para llegar a un resultado coste computacional y estamos utilizando información que es realmente relevante para nuestro modelo. | |

03/04/23 - **Kernels y KPCA**

|  |  |
| --- | --- |
| Key Points | Notes |
|  | **Kernels y KPCA**  Ahora que ya sabemos para el algoritmo de PCA, ¿qué otras alternativas tenemos?  **Kernels**: es una función matemática que toma mediciones que se comportan de manera no lineal y las proyecta en un espacio dimensional más grande en donde son linealmente separables.  **Y, ¿esto para que puede servir?**    Sirve para casos en donde no son linealmente separables. En la primera imagen no es posible separarlos con una línea y en la imagen 2 si lo podemos hacer mediante Kernels. Lo que hace la función de Kernels es proyectar los puntos en otra dimensión y así volver los datos linealmente separables.  **¿Qué tipo de funciones para Kernels nos podemos encontrar?**    **Ejemplos de funciones de Kernels en datasets aplicados a un clasificador**: |
| Sumario: | |

03/04/23 - que es la regularización y como aplicarla?

|  |  |
| --- | --- |
| Key Points | Notes |
|  | Qué es la regularización y cómo aplicarla?  Esta técnica consiste en disminuir la complejidad de nuestro modelo a través de una penalización aplicada a sus variables más irrelevantes.    Como podemos apreciar en la gráfica 1, hay un sub-ajuste, ya que la línea roja se acopla muy bien para los datos de prueba, pero no para los datos de entrenamiento. La línea roja en los datos de prueba da una mala generalización, una mala aproximación.  Entonces, la regularización consiste en introducir un poco de sesgo para introducir la varianza de los datos.  Pero para poder aplicar regularización necesitamos un término adicional el concepto de perdida. El concepto de perdida nos dice que tan lejos están nuestras predicciones de los datos reales, esto quiere decir que entre menor sea la perdida mejor será nuestro modelo. |
| Sumario: | |