Ciencia de datos

References:
Alex wang (python)
https://github.com/EngAndres/ud-public/tree/main/courses/data-science-introduction

Palabras clave:

- datos sesgados
- Stakeholders
- forecasting
- Business intelligent
- Data Lake
- data Warehouse
- ETL: Extract, Transform, Load
- insight
- Data pipelines (Procesos)
- casting
- EDA

Que es:
Usa método científico, procesos, algoritmos par extraer conocimiento

Scrum, metodologías ágiles,

Datos: Exploración, limpieza (datos correctos), balanceados, no sesgados, análisis, visualización (colores), modelos productivos (machine learning), testing (corregir errores, cada vez menos errores), despliegue (DevOps, MLOps, LLMOps, montar en servidores/ nube/ Docker)

Tomar decisiones: OF, a través de todo el análisis, tomar el mejor camino, la mejor decisión, pivotear, para que la empresa siga siendo rentable

Data oscura: Datos recopilados pero no usados para la toma de decisiones

Proyecto
Kaggle -> contest: escoger un problema a resolver

Definir problema / entender problema: representar como un sistema

Obtener los datos: apis, web scraping

Calidad de datos: limpiar ejemplos malos, formato de datos, prepararlos, ver info balanceada y relevante.

Explorar datos

Construir modelos: Base de datos, datos extraídos de la data Warehouse.

Evaluar modelos: Contruir solución, machine learning

Comunicar resultados: Stakeholders

Data Análisis:
Que paso, porque paso, que puede pasar y que es lo mejor que puede seguir pasando

1. Descriptivo: hechos, situaciones, factos, que pasó

2. Diagnóstico: Entender los datos, el porque suceden cosas, correlaciones (una cosa afecta otra)

3. Predictivo: jugar con variables, what if, forecasting (predecir valores futuros basados en datos históricos), que es lo más probable?

4. Prescriptivo: Bussines intelligent, generar recomendaciones (tinder),

Ej: Tinder:
- Info personal, gustos, preferencias
- porque le dió me gusta
- probabilidad de que le gusten algunos perfiles
- sugerir personas con la misma descripción a la cual hizo match

Big data
Uber maneja 17 petabytes a la semana

Datos extremadamente grandes par ser analizados

Sistemas de datos: almacenar, recuperar y enviar en tiempo real. Tecnologías. Streaming, kafka. Apache.

3Vs: Volumen (cantidad), Velocidad (En milisegundos), Variety (tipos, fuentes de datos)

Sistemas para big data: SQl, No SQL(MongoDB, Cassandra), sistemas distribuidos (Hadoop, Spark)

Data Lake
Repositorio, almacenamiento en formato RAW (crudo, sin procesar, nativo), cualquier tipo de información tal cual como llegue

Data Warehouse (BD relacional)
Sistema enfocado al reporte y análisis de datos para inteligencia de negocios.
La data del data lake se formatea, limpia y transforma para ponerle en la data Warehouse

ETL: Extract, Transform, Load

Data lake (E) -> preprocesamiento (T) -> data warehouse (L)

Postgres

Bases de Datos
Mongo es lento

Tablas relacionales
Tablas orientadas a Grafos
Vectores (planos cartesianos): Entrenan IA

OLAP: Se tiene todo en memoria para entregar info en milisegundos

IA
AI: Simular inteligencia humana

Machine learning: Proveer al sistema la habilidad de aprender automáticamente mediante datos y que sepa cómo y qué problemas puntuales resolver. Redes neuronales.
Today: Specific IA, Future: General IA.
Problema de la interpretabilidad: no saber como la máquina llegó a la solución. No saber cómo aprendió lo que aprendió.

Deep learning: division de machine learning.
Red neuronal: Función polinómica que interprete algo.
Millones de neuronas: Capas de neuronas hasta llegar al resultado.

LLMs Landscape: Language Largue Models (Modelos de Lenguaje de Gran Escala). SLMs (Modelos mas pequeños)
Ej: GPT4: 10B tokens, 170B neuronas en la primera capa, 3 Trillones de ejemplos, 20Mil procesadores, durante 3 semanas

Data y MetaData
Data: raw, unprocessed, unorganized facts de datos que por sí solos no tienen sentido

insight: La opción más rentable a tomar

MetaData: data about data. Información que se puede sacar de los datos. Who, what, where, when, why, how of data
- Ayuda a descubrir, organizar e interpretar
- Ayuda a manejar, governance, cataloging, data lineage (cómo llegaron a dónde están)
- *Permite atacar menos a la base de datos*.

Usos de data science
Data science usada en industrias para tomar decisiones y optimizar procesos
- Salud para predecir planes de tratamiento, patient outcomes
- Finanzas para detectar fraudes, automatic trading, algorithmic trading: bots que predicen en trading
- Venta al menor para optimizar precios, pronóstico de demanda, personalizar marketing
- Manufactura para predecir cuando una máquina se va dañar, mejorar control de calidad
- Transporte para optimizar rutas

Reinforcement learning: Q-learning

Tipos de roles (data)
- Ingeniero de datos: Bases de datos, carga datos
- Analista de datos: Entrega de resultados, recomendaciones, métricas, saber interpretar datos, PowerBI, Tableu, Dashboards
- Ingenieros de machine learning: Desarrollar, desplegar
- Científico de datos: Experimentos, analizar datos, decir órdenes a la gente, integra todo, <u>hacer de todo</u>
- Steward de datos: Evita que los datos se dañen

Responsabilidades
- Colectar conjuntos grandes de datos
- limpiar y validar datos
- Analizar datos
- Interpretar datos
- Desarrollar machine learning models
- Comunicar recomendaciones
- Mantenerse actualizado

Data pipelines (Procesos)
Serie de pasos para transformar la información
Raw (datos sin procesar) → ETL (Enviar a warehouse) → Warehouse (Formateo de raw) → Apps (Analizar datos) → PowerBI (Representar info)

ETL (Extract Transform Load)**,** Dagger, apache airflow
Extraer datos automáticamente y repetidamente
Transforma los datos en un formato para analizarlo

Python
Creado por guido van rossum
- Alto nivel: Diseñado para ser fácil de leer y escribir
- Interpretado: Ejecutado línea por línea. Cada vez que se ejecuta se lee la línea y se traduce a código assembly.
Código no se compila directamente en lenguaje
Se usa un intérprete para leer y ejecutar el código línea por línea en tiempo de ejecución
Crear código → compilado → genera .exe . Pre compila a C para pasar a assembly (es rápido)
- Tipado débil: No specific data
- Multiparadigma: Soporta orientado a objetos
- Snake case: var_name
- Propósito general,

—
Instalar jupyter
pip install -r <>.txt
.txt: mkdocs==<version>
mejor poetry

—
- Módulos: Conjunto de funciones, clases, variables: from <> import <>. Organizan código
- Packages (carpetas): Organizan módulos. Agregar __init__,py
- Dependencias: Librerías externas, conflictos de versiones
- Poetry: Se encarga de decidir qué versiones son las mas estables según las dependencias que se le pasen

—
Jupyter Notebook (Notebook git)

—
- Apuntadores: Apunta a una dirección en memoria (x → 0x00) id(x)
- Asignación por valor: Copia el valor

x → 0x00
y → 0x23

- Asignación por referencia: Uno o varios objetos apuntan al mismo espacio en memoria
  obj1 = Class()
  obj2 = obj1 (Si se cambia obj1, se cambia obj2)

—

Casting: cambiar el tipo de dato
Conjunto/Set: {a,b,c} Colección desordenada y no indexada, sin repetir

List comprehension: camino más cortó para crear una lista

Variadic functions: Recibe algunos parámetros obligatorios y otros opcionales

Iterators:

- map: Aplicar una función a cada uno de los ítems de una lista (plano cartesiano f(x) → y)
- filter: selecciona ítems que cumplan una condición

Lambda: Funciones anónimas con una línea de código, reduce memoria, legibilidad más rápida, menos código.

Numpy
Core library: Lo más importante
Paquete para manejar arreglos, paquete fundamental para scientific computing.
Propósito general
High-performance multidimensional array object (alto performance para matrices)
Creado por travis oliphant, open-source

outlier: anomalía de la desviación estándar (muy desviado). Al eliminarlas se hace limpieza de datos

Vectorización
Hacer menos costosa la operación, consume mas memoria pero menos procesamiento (tiempo)
Reemplazar ciclos con operaciones de arrays

Si la longitud del vector es muy pequeña, la diferencia entre la vectorización y los ciclos no va a tener gran diferencia o ventaja

boolean mask: Retornar arrays con una condición adentro. Ej: array[array % 2 == 0]

Strings
Son inmutables, ordenadas, iterables, indexable, slicable (slice). Python usa unicode (lexicográfico: comparar caracter por caracter)

Dates: Todas las fechas se guardan con hora suru

Regular expressions (ReGex): Para buscar patrones dentro de una cadena de caracteres

---

Pandas
Top para análisis y ciencia de datos
Manipulación de datos de alto nivel
Creada por Westcol McKinney
Construido sobre numpy

Series object (core of pandas): Un arreglo de una sola dimensión que contiene un arreglo con labels indicando sus índices

Data Frames:
Es una estructura de tabla de datos (excel / SQLtable).
Core data structure.
Contenedor de series
Se puede indexar el DF con una columna de nombres

timestamp: Cantidad de segundos que han ocurrido desde 1970

variables = columnas

feature engineering: Generar una nueva columna a partir de los datos ya existentes

---

1ffsGKIrEFq9bATLujx1NwnBJcWUiJxXlnTgIAaZ

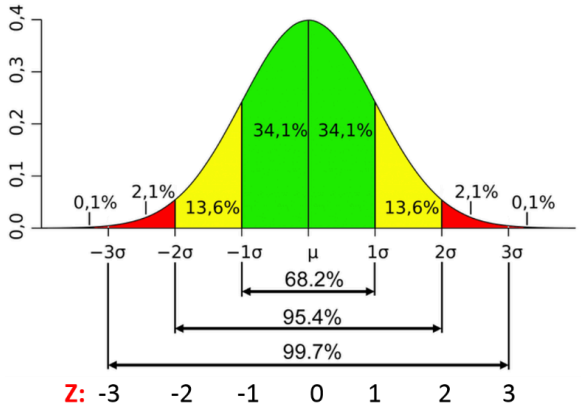
Data streaming: Transferir datos continuamente en tiempo real
Data profiling: Analizar datos para entender su estructura, calidad y contenido
Data cleaning: Detectar y corregir errores
Correlación: Determinar redundancias, ver si al cambiar variables que tanto cambian otras. Ver cual es la correlación más alta entre dos variables entre el rango (-1, 1)

Análisis exploratorio de datos (EDA):
Analizar los datos para resumir sus características principales

- Es el primer paso en el análisis de datos
- Generar insights e hipótesis

Outlier:
Valores demasiado lejos de la tendencia que generan ruido al calcular descripciones. Se pueden detectar o quitar usando el z-score (value-mean/std)
Los outlier se pueden eliminar cuando estan en una distribución normal (normalizados)
En una distribución normal:

En el rango *primer sigma* los datos son muy confiables 68.9%
En el *segundo sigma* son un poco menos 84%
En el *tercero* son algo confiables 99.5%
En el *tercero en adelante* son anomalías, nada confiables



Diseño

Business insight:

- What?
- For what?
- Impact?

Rueda de visualización:

- Hemisferio superior: Profundizar en explicación pero más difícil de entender
- Hemisferio inferior: Más superficial pero más fácil de entender

heurísticas:

- Algoritmos de aprendizaje
- Redes neuronales

Matplotlib:

Visualizaciones interactivas
Construido sobre arrays de numpy
Diseñado para trabajar con datos en 2d y con broader SciPy stack
Tiene 3 capas: Backend (No se toca), Artist (Components, colors), Scripting (Se escribe el código)

Reglas para mejores figuras / gráficos:

- Conocer a la audiencia
- Identificar mensaje
- Adaptar la figura a donde la vaya a transmitir (Tamaños, formatos)
- Colocar el texto necesario
- No confiar en las configuraciones por defecto
- Usar los colores correctamente
- No confundir al lector
- Evitar “Chartjunk”: Ser minimalista (fondos innecesarios, grillas innecesarias)
- Mensajes agradables
- Obtener la herramienta correcta

Scatter plot:

- Son usados para observar relaciones entre variables

line plot

- Observar tendencias en series de tiempo

subplot:
pequeños grupos de gráficas existiendo en una sola imagen

Histograma
Muestra frecuencias de valores (cuantas veces ocurre cada valor)
gráfico desde el mínimo valor encontrado en la serie de valores lo deja al inicio y el mayor al final

Box Plots:
Muestra la dispersión y la tendencia central de valores en un dataset
Los datos se distribuyen de forma normal

Heatmap
Muestra la intensidad de los datos, la correlación
Mapa de calor

Widgets
Interfaz con botones y seleccionables para interactuar con las gráficas

SeaBorn
Construida sobre matplotlib

Spurious correlations
Para conocer la correlación entre dos variables
Muestra una diagonal si hay una fuerte correlación

Ética
Mal uso de los datos:
Usar datos desbalanceados para tomar decisiones

Compartir datos:
Los datos anónimos que se llenan en formularios se pueden hallar de-identificando haciendo las preguntas correctas, como haciendo Joins o merges

Privacidad de datos
Filtración de datos

Selfies nudes
Incluso con partes distorsionadas se puede obtener información sobre tatuajes o color de pelo

deep fakes, deep voices

Training data
Los actuales modelos de ia a.k.a LLMs son los casos preferidos de los abogados
A quien le exigimos si nadie es dueño de nada en la ia

Machine learning
Metodo de analisis de datos que automatiza modelos analiticos
Rama de la inteligencia artificial basado en la idea de que un sistema puede aprender de datos, identificar problemas y tomar decisiones con la mínima intervención humana
Se usa la librería de Scikit-learn

Code: scaler: Para que todas las categorías queden en un rango fijo

- Supervised learning: Model trained on labeled data (datos etiquetados). Es el más usado
- Unsupervised learning: Algoritmos que no tienen una salida esperada (datos no etiquetada). Preprocesamiento
- Reinforcement learning: Aprende a través de interactuar con el entorno. Requiere más experiencia y más tiempo, recibe recompensas por resultados

Procesamiento de data

- Clasificación: Predecir etiquetas (ganadores, perdedores, llueve o no,)
- Regresión: Predecir una variable continua: Números, cantidades. (El valor de una acción, la temperatura en el lapso de un tiempo)
- Clustering: Agrupar puntos similares de datos,
- Reducir la dimensionalidad: Reducir el número de features/datos (feature engineering), eliminar datos redundantes
- Deteccion de anomalias: Identificar datos inusuales o anormales
- Asociación de reglas: Identificar relaciones entre variables

Trabajo de flujo:

Data collection: Reunir los datos,
Data processing: Clearing and preparing de data
Feature engineering: Clearing de features
Model selection: Elegir el mejor modela
Model training: Entrenar el modelo sobre la data
Model evaluation: Asesorar el rendimiento de los modelos
Model deployment: Poner el modelo en producción

Examinar los datos:

Data Exploration: Understanding de data
Data cleaning: Preparing de data
Feature engineering: Creating new features
Feature Selection: Selecting the most important data (Correlaciones)
Data preprocessing: Preparing the data for modeling

K-Nearest Neighbors: (Dime con quien andas y te diré quien eres)

Algoritmo que encuentra los valores más cercanos a un nuevo valor para poderlo clasificar
Al escoger los valores más cercanos (k), el nuevo valor sera la clasificacion que mas se repite

Euclidiano: $\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$

Castigo: Entre más grande la distancia más grande, Si es entre 0 y 1 la distancia será más pequeña

Para clasificación, la respuesta es la etiqueta
Para Regresión, la respuesta es el promedio a los valores más cercanos

Algorithmic Bias (sesgo):

Son errores sistemáticos en un modelo que resulta en resultados dispares (a veces sale un resultado y a veces otro)

Overfitting:

Ocurre cuando el modelo está entrenado muy bien pero no sabe responder a cosas nuevas, (Ej: Aprender mucho de python pero poco de logica de programacion, y no sabremos defendernos en nuevos lenguajes)

Underfitting:

Cuando un modelo es muy simple (pocos datos) y no se entrena bien

Supervised learning

Training dataset: El 70% de datos para enseñar al modelo, el 30% se deja para hacer pruebas
Validation dataset: El 10% de los Datos restantes son para tunear el modelo con hyper parametros (parametros de configuración del algoritmo, Ej: 'k' del k-Nearest)
Test Dataset: El 20% de datos para evaluar el desempeño del modelo. Si hay poco porcentaje de error quiere decir que por lo menos no hay underfitting

Regression:

Clasifica una columna y saca un promedio para

Linear regression:

Consiste en encontrar los valores correctos de una ecuación lineal para encontrar una línea que mejor represente el comportamiento de datos
Utiliza el algoritmo de los mínimos cuadrados:
ax + b: a(peso), b(bias)

Ridge Regression

Es un tipo de regresión lineal pero agrega penalización a los w muy grandes para evitar overfitting.
Añade regularización a la función de mínimos cuadrados
Determina si hubo un potencial overfitting o no.

Lasso Regression

Lazo permite que haya algunos w más grandes que otros
Se utiliza para determinar cuales features/columnas, brindan mayor información. Al trabajar con muchas columnas determina cuales son las más importantes

Polynomial Regression:

Util para capturar operaciones no lineales, donde la relación entre las variables dependientes e independientes son de grado n polinómica
Hay overfitting cuando los grados son muy grandes
Hay underfitting cuando los grados son muy pequeños

Logistic regression

Se usa para clasificación binaria
Lo que esté por debajo de una línea es una clase y lo que esté por encima es otra diferente

Cross validation

- Técnica para asesorar el rendimiento de un modelo
- Evitar overfitting
- Disminuye el bias(sesgo)
- k-fold: Divisiones para entrenar |1|2|3|4|5|, Si se prueba con 2, se entrena con 1, 3, 4 y 5

One-Hot encoding

- Técnicas para convertir variables categóricas a numéricas
- Genera vectores binarios para cada categoría
- Cada categoría se convierte en columna

Data Leakage

- Data duplicada
- Ocurre cuando una parte del dataset se utiliza sin darse cuenta en el conjunto de entrenamiento (Ej: Data repetida, los mismos valores de prueba que en los de entrenamiento)

Support vector machines

- Trabaja buscando un hiperplano que separa la data en diferentes clases, debido a que una tabla por lo general tiene más de una columna. En vez de dividir las features en líneas, se separan en planos

Árboles de decisión

- Puede ser usada tanto para tareas de clasificación como de regresión
- Permite particionar la data y hace conjuntos de preguntas basadas en los valores de las features
- Se quita el problema de que columnas son más importantes que otras
- Son muy rápidos

Naive Bayes Classifier

- Ingenuo
- Clasificador basado en el teorema de bayes: Probabilidades condicionales
- Asume que las features son condicionalmente independientes dado etiquetas de clases
- Se hace árbol de probabilidades

Modelos que utilizan probabilidades para predecir comportamientos: Estocásticos

Random Forest

- Es uno de los modelos más fuertes
- Múltiples árboles de decisión aleatorios promediando sus predicciones quitando el overfitting
- Profundidad: Cuántas preguntas se harán hasta abajo del árbol
- Para cada uno de los árboles saca los principales árboles con el mejor accuracy, mirando la feature que aparece en la raíz y los que más se repiten en ellos

Gradient Boosted Decision trees

- Modelo predictivo fuerte
- Es un aprendizaje ensamblado que combinas varios árboles de decisión y gradientes
- Genera árboles y corrige errores usando derivadas
- Construye árboles secuencialmente cada vez corrigiendo los errores de los anteriores arboles
- Se necesita optimizacion de parametros

Neural Networks

Son un tipo de modelo de machine learning inspirado en el cerebro humano
Una serie de neuronas que sacan una función polinómica para clasificar
Una neurona saca una clasificación según las features pasadas, genera una línea de regresión
Ya no tenemos una sola regresión lineal sino varias
Backpropagation: A través de derivadas parciales se asignan pesos a cada neurona (?)

Model evaluation

Asesorar el rendimiento del modelo
Elegir el mejor modelo para la tarea
Sacar múltiples métricas para sacar una medición de cual modelo se comporta mejor que los otros

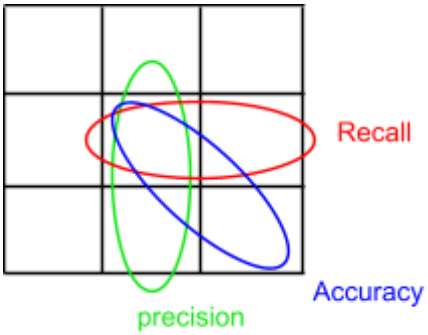
Matrices de confusión

Es una matriz que resume el rendimiento de los modelos, una por modelo
Predicted matrix:

	C1	**C2**
C1	True Positive	False Negative
C2	Falso Positive	True Negative

Basic evaluation metrics

Accuracy: Proporción de las predicciones correctas (TP + TN)/(TP+FN+FP+TN)
Precisión: Proporción de true positives entre todas las predicciones positivas (TP)/(TP+FP)
Recall: Proporción de de los true positives correctos (TP/TP+FN) : (1:Expected true is true, 0:Expected true is False)
F1-Score: Promedio armónico de precisión y recall ((precision*recall)/(precision + recall)). Si se acerca a 1 los falsos tanto negativos como positivos se estan disminuyendo



Classifier Decision metrics:

ROC Curve: Entre mas curvada hacia arriba mejor, habran mas True
Precision Recall-curve:
AUC-ROC
AUC-PR

Multi-Class Evaluation

Macro averaging:
Micro averaging:
weighted averaging:
one vs all:

Model Calibration

Proceso de ajustar el resultado para que coincida con la verdadera probabilidad de distribución
Ajustar al modelo para que maximice algunos resultados, como el accuracy, f1-score, etc.

proyecto: Decisiones que tomamos, para q es útil,

POSTER cientifico ieee
contexto
soluciones previas
Como se resolvio: método con el q se resolvio , como se implementó
resultados
conclusiones

PAPER ieee ~ 5 pag. (el profe los pasa en un evaluador de formato ieee)

abstract: cual es el problema, resumen como lo resolvieron otras personas, como lo hicimos nosotros

Background: (contexto) Pagina donde se pone el contexto del problema, (hablar de cocina), tecnicas en el pasado para resolver problema

Metodos: Desarrollo, Solucion q hicimos 2 pag. (tabla, diagrama, tantos nulos, anomalías, esquema, ecuaciones, metricas de error, metricas de lasso, modelos) que se hizo

2pag:
Resultados: Cosas estadisticas, metricas de error, probando con columnas, exp sobre datos (quitando o dejando nulos),

discusion: Ver si los resultados cumplen con el problema, a partir del metodo que propuse y los resultados resuelven problema

conclusion: Media pag. que analisis, algoritmos fueron correctos, que paso con la data, que resultados fueron relevantes

Biblio:

INFORME
Collecting: Análisis descriptivo y diagnostico como recopilamos la data, de donde la obtuvimos, estructura, cuantas filas, columnas, Detalle completo de la recolección de datos

Preprocessing: Que limpiamos, PORQUE limpiamos, porq lo hicimos, porque hicimos todos los cambios,

Feature engineering: Porq creamos columnas, ej: porq cambiamos a categorical, a partir de los países sacamos región porque era más útil,

Model selection: Sr escogieron tales features, probamos modelos y escogimos el mejor

Model training: Cambio de parametros

Model evaluation:

Si se convierte el target a categorical se tendrá un mejor resultado