

Matthew Espinosa – TS7A

Parcial 3 - Ciencia de datos

1. Google Teachable Machine

1. Describe el proceso que seguiste para generar y entrenar el modelo.

- Ingresar al sitio web <https://teachablemachine.withgoogle.com/>
- Crear un nuevo proyecto de imagen
- Seleccionar el modelo de imagen estándar
- Crear 2 clases, en este caso fue una de carros y otra de motos
- Adjuntar las imágenes correspondientes en cada clase
- Preparar el modelo
- Probar el modelo y verificar que esté funcionando correctamente
- Exportar el proyecto

2. Exporta el modelo e intégralo en un proyecto con p5.js. Adjunta el código.

- **Nota:** Código adjunto como p5.js

3. ¿Qué ventajas tiene esta herramienta para prototipado rápido?

- **Entrenamiento en tiempo real:** Permite visualizar los resultados del entrenamiento de forma instantánea, lo que facilita ajustes rápidos.
- **Sin necesidad de grandes conjuntos de datos:** Es posible crear modelos efectivos con pocas imágenes, ideal para pruebas rápidas.
- **Interfaz intuitiva:** Su diseño es amigable y no requiere conocimientos avanzados de programación o herramientas complejas.
- **Integración directa con aplicaciones web y móviles:** Los modelos exportados son fáciles de usar en proyectos con TensorFlow.js, Android, entre otros.
- **Uso gratuito y accesible en la nube:** No requiere instalación de software y se puede usar desde cualquier navegador.

2. Modelado con Kaggle

- **Nota:** Código adjunto como data_analysis.py

3. Complejidad y Paradigma P y NP

1. El paradigma de P vs NP con un ejemplo de cada uno.

P es el conjunto de problemas que pueden ser resueltos en tiempo polinómico. Es decir, su complejidad crece de forma razonable a medida que aumenta el tamaño de la entrada.

- **Ejemplo de P:** El **problema de la suma de subconjuntos** en su versión más sencilla, donde se verifica si un subconjunto de números suma un valor objetivo. Este problema se puede resolver eficientemente con algoritmos como la programación dinámica.

NP es el conjunto de problemas cuya solución, si se proporciona, puede ser verificada en tiempo polinómico, pero no necesariamente se sabe si la solución puede ser encontrada rápidamente (en tiempo polinómico).

- **Ejemplo de NP:** El **Problema del Viajante de Comercio (TSP)**, que consiste en encontrar el recorrido más corto que pasa por un conjunto de ciudades. Si se nos da una posible solución (un recorrido), podemos verificar rápidamente su validez, pero encontrar la solución óptima es muy difícil.

2. ¿Cómo afecta este paradigma a la solución de problemas en ciencias de datos?

El paradigma **P vs NP** impacta profundamente en la ciencia de datos, ya que muchos problemas son NP-completos o NP-difíciles, lo que implica que encontrar soluciones óptimas puede ser muy costoso computacionalmente.

- **Optimización en Machine Learning:** Muchos problemas en aprendizaje automático, como el ajuste de hiperparámetros o la selección de características, son NP-difíciles. Los algoritmos exactos pueden ser inviables para grandes volúmenes de datos, por lo que se utilizan enfoques aproximados o heurísticos, como el descenso de gradiente.
- **Problemas combinatorios:** Tareas como el análisis de grandes redes o el clustering pueden implicar problemas NP-completos, lo que hace que se prefieran soluciones aproximadas debido a la alta complejidad de obtener la solución óptima.
- **Escalabilidad:** En Big Data, donde los datos son masivos, resolver problemas NP-completos de manera exacta se vuelve impracticable. Por lo tanto, se recurre a algoritmos que encuentran buenas soluciones sin garantizar la

óptima, lo cual es una necesidad en entornos de procesamiento de grandes volúmenes de datos.

En resumen, aunque muchos problemas en ciencia de datos están relacionados con problemas NP-difíciles, se utilizan algoritmos aproximados o heurísticas para obtener soluciones en tiempos razonables, lo que es esencial para la práctica en el análisis de datos.

4. Algoritmos de Optimización

- **Nota:** Código adjunto como tsp_algorithms.py

```
➡ 2-SAT
Resultado 2-SAT: [True, True, False]

TSP
Ruta optimizada: [0, 1, 4, 3, 2]
Distancia total: 11.31370849898476

Recocido Simulado
Mejor solución encontrada: 1.998411418816953
Valor mínimo de energía: 2.523590175051993e-06
```

5. Importancia de la Optimización en las Organizaciones

En la era de la información, donde los datos crecen de manera exponencial, optimizar algoritmos y su manejo se ha convertido en un factor clave para el éxito organizacional. Las empresas que gestionan grandes volúmenes de información enfrentan desafíos como el procesamiento eficiente, la toma de decisiones en tiempo real y la reducción de costos operativos. Sin estas optimizaciones, los procesos pueden volverse lentos e ineficaces, afectando tanto la competitividad como la experiencia del cliente.

Optimizar algoritmos significa utilizar de manera eficiente los recursos disponibles, como el tiempo de procesamiento, la memoria y la capacidad de almacenamiento. En la industria financiera, por ejemplo, los algoritmos optimizados son esenciales para procesar millones de transacciones en tiempo real, detectar fraudes y realizar análisis predictivos que guíen estrategias de inversión. Incluso unos pocos segundos de retraso pueden generar pérdidas significativas.

En el sector logístico, la optimización de rutas mediante algoritmos como el TSP (Traveling Salesman Problem) permite a empresas como Amazon o DHL reducir costos y garantizar entregas más rápidas, incluso en períodos de alta demanda.

El sector salud también se beneficia enormemente. Los algoritmos optimizados procesan imágenes médicas, analizan datos genómicos y agilizan el acceso a registros electrónicos, mejorando la precisión de los diagnósticos y salvando vidas.

En resumen, la optimización no es solo un desafío técnico, sino una herramienta estratégica que permite a las organizaciones mantenerse competitivas, innovar y adaptarse a un entorno cada vez más dinámico y exigente. Desde la logística hasta la salud, esta capacidad define el éxito en la era digital.