Evaluación del Trade-off entre Emisiones de Carbono y Rendimiento en Sistemas de Recomendación

Gabriel Catalán
Pontificia Universidad Católica de Chile
Santiago, Chile
gicatalan@uc.cl

Ilan San Martín
Pontificia Universidad Católica de Chile
Santiago, Chile
ilansanmartink@uc.cl

RESUMEN

Este trabajo evalúa el trade-off fundamental entre el rendimiento de las recomendaciones y las emisiones de carbono (CO_2-eq) asociadas a su entrenamiento. Se implementa una metodología experimental utilizando la librería Surprise y el dataset MovieLens 1M. Se entrenan y comparan varios modelos de filtrado colaborativo, incluyendo UserKNN, ItemKNN, SVD y LightFM, bajo distintas fracciones del dataset (10 %, 25 %, 50 %, 75 %, 100 %). La calidad de las recomendaciones se mide utilizando la métrica de ranking NDCG@10, mientras que el impacto ambiental se cuantifica mediante la herramienta CodeCarbon. Los resultados demuestran que LightFM es superior en calidad de ranking (NDCG@10 de 0.3116), mientras que SVD ofrece el balance más eficiente entre un bajo error de predicción y la huella de carbono más baja. Se concluye que la selección del modelo y el tamaño del dataset son cruciales para una estrategia de Green RecSys, ya que es posible obtener un rendimiento cercano al máximo con una fracción significativamente menor de las emisiones.

1. INTRODUCCIÓN

A lo largo de la última década, el desarrollo de modelos de aprendizaje automático ha experimentado un crecimiento exponencial, tanto en complejidad como en capacidad computacional. Esto ha permitido alcanzar mejoras significativas en diversas aplicaciones, como lo son los motores de búsqueda, las recomendaciones personalizadas, los asistentes virtuales, entre muchas otras. Sin embargo, estas mejoras han traído consigo un mayor consumo de recursos y energía, lo que se traduce en un aumento importante en las emisiones de carbono asociadas. Schwartz et al. [4] destacan esta tendencia y señalan que la gran mayoría de personas y empresas busca obtener buenos resultados computacionales a costa de mayores emisiones, lo que se conoce como *Red AI*. Como alternativa a esto, proponen un enfoque orientado a construir modelos más eficientes y sostenibles, que definen como *Green AI*.

Bajo esta línea, el presente trabajo busca estudiar la implementación de *Green RecSys*, que corresponde a una mirada más específica dentro del enfoque *Green AI*, centrada exclusivamente en los sistemas de recomendación. Más concretamente, se busca evaluar el *trade-off* entre las emisiones de carbono generadas al entrenar distintos modelos de recomendación y la calidad de las recomendaciones entregadas, pues mejores resultados suelen requerir un mayor uso de recursos, lo que se traduce en más emisiones. Para esto, se comparan distintos métodos entre sí, al mismo tiempo que se analiza cómo a medida que aumenta el tamaño del dataset, mejora la precisión, pero también crece el impacto ambiental. El objetivo entonces es identificar cómo se comportan distintos modelos frente a distintas cantidades de datos, entendiendo la relación entre rendimiento y emisiones en cada caso.

[0]Código fuente disponible en: github.com/gicatalan/RecSysProjectGroup15

2. REVISIÓN DEL ESTADO DEL ARTE

A continuación, se presentan distintos estudios de metodologías relacionadas con *Green RecSys*, fundamentales para la base del estudio realizado en este trabajo.

2.1. GreenRec

Uno de los trabajos más importantes en el área de recomendación sustentable es el de Liu et al. (2024) Liu et al. [3], quienes se enfocan en sistemas de recomendación de noticias y proponen un paradigma de entrenamiento llamado *Only-Encode-Once* (OLEO). Este método busca reducir drásticamente las emisiones de carbono al evitar el procesamiento repetido de los mismos artículos durante el entrenamiento, lo cual es una de las principales fuentes de ineficiencia en los modelos tradicionales end-to-end. En lugar de codificar múltiples veces cada noticia, OLEO entrena previamente un codificador de contenido y luego reutiliza las representaciones almacenadas, lo que permite reducir el costo computacional sin sacrificar demasiada precisión. Para medir la eficiencia, los autores introducen la métrica ApC (AUC por Emisión de Carbono), que relaciona el rendimiento del modelo con su huella ambiental. Según sus resultados, el enfoque OLEO logra una eficiencia hasta un 2992 % mayor que métodos end-to-end basados en modelos de lenguaje preentrenados.

Si bien en este trabajo no se experimenta directamente con OLEO, se considera que es una propuesta muy relevante dentro del enfoque *Green RecSys*, ya que demuestra que es posible alcanzar un buen desempeño a un costo ambiental mucho menor.

Dataset		MIND-small						MIND-large					
		NewsRec			CTR			NewsRec			CTR		
Method		NAML	LSTUR	NRMS	BST	DCN	DIN	NAML	LSTUR	NRMS	BST	DCN	DIN
ID-based	AUC	50.13	51.04	54.84	50.09	53.92	55.95	52.98	54.98	57.59	52.10	57.41	57.36
	MRR	23.01	22.90	26.53	22.13	25.18	25.88	24.52	25.99	27.41	24.81	26.76	26.70
	N@5	22.35	22.31	26.34	21.59	24.43	25.95	24.12	25.64	27.05	24.63	26.90	26.84
	$CO_2E\downarrow$	**19**	**20**	**28**	**38**	**60**	**84**	**294**	**353**	**471**	**555**	**926**	**1294**
	ApC	0.68	5.20	17.29	0.24	6.53	7.08	1.01	1.41	1.61	0.38	0.80	0.57
Text-based (End-to-end)	AUC	60.14	61.27	62.21	60.51	62.63	62.90	63.03	63.89	64.12	63.28	63.88	64.02
	MRR	28.93	29.64	30.19	28.59	29.73	30.06	30.40	31.24	31.77	30.73	31.65	31.98
	N@5	29.33	30.28	31.10	29.09	30.52	30.65	31.82	32.15	32.64	31.95	32.40	33.00
	$CO_2E\downarrow$	42	58	62	53	63	90	648	892	1010	1212	972	1386
	ApC	24.14	19.43	19.69	19.83	20.05	14.33	2.01	1.56	1.40	1.09	1.43	1.01
PLM-NR (End-to-end)	AUC	62.06	**63.64**	62.53	**64.40**	63.32	**63.26**	**65.19**	**65.73**	**65.57**	**66.03**	65.42	**65.31**
	MRR	**31.66**	**31.74**	30.74	**32.21**	32.00	**31.83**	**32.74**	**33.18**	**32.94**	**33.40**	32.85	32.68
	N@5	**32.25**	**32.72**	31.31	**33.34**	32.58	**32.40**	**33.77**	**34.26**	**34.13**	**34.70**	33.99	**33.70**
	$CO_2E\downarrow$	178	202	252	505	1,752	1,839	2,527	3,032	4,043	8,086	27,036	28,329
	ApC	6.78	6.75	4.97	2.85	0.76	0.72	0.60	0.52	0.39	0.20	0.06	0.05
BERT (OLEO)	AUC	60.62	61.09	60.94	60.81	62.65	62.40	63.02	63.62	63.40	62.94	64.29	63.75
	MRR	29.31	29.26	29.31	29.04	30.92	30.75	31.23	31.59	31.38	30.56	32.60	31.58
	N@5	29.71	29.60	29.65	29.38	31.37	32.44	31.79	32.30	32.16	31.83	33.63	32.43
	$CO_2E\downarrow$	22	23	33	38	62	86	353	404	505	640	956	956
	ApC	48.27	48.22	33.15	28.45	20.40	14.41	3.69	3.37	2.65	2.02	1.49	1.44
PREC (OLEO)	AUC	**62.95**	62.16	**62.95**	62.43	**64.57**	63.12	64.78	64.88	64.34	65.33	**65.44**	64.53
	MRR	31.26	31.00	**31.18**	30.42	**32.60**	31.28	32.64	32.94	32.93	33.29	**33.04**	**32.72**
	N@5	32.01	31.79	**32.10**	30.94	33.48	32.01	33.66	34.00	33.95	34.35	**34.03**	33.58
	$CO_2E\downarrow$	22	23	33	38	62	86	353	404	505	640	956	956
	ApC	**58.86**	**52.87**	**39.24**	**32.71**	**23.50**	**15.25**	**4.19**	**3.68**	**2.84**	**2.40**	**1.61**	**1.52**
	ApC Imp. (%)	768%	683%	690%	1048%	2992%	2018%	598%	608%	628%	1100%	2583%	2940%

Figura 1: Resultados de GreenRec, comparando la eficiencia ambiental (ApC) entre modelos tradicionales y OLEO.

2.2. Reducción del tamaño del dataset

Otro trabajo fundamental en el área es el de Arabzadeh et al. (2024) [1], que corresponde a la base del diseño experimental de este estudio. En su investigación, los autores analizan cómo afecta la reducción del tamaño del dataset al rendimiento de distintos modelos de sistemas de recomendación, utilizando cuatro conjuntos de datos: MovieLens (100K, 1M y 10M) y Amazon Toys & Games. A partir de múltiples experimentos, se agrupan los modelos en dos categorías según su sensibilidad a la reducción de datos. El Grupo 1 (que incluye UserKNN, ItemKNN, SVD y NMF) mostró una pérdida de rendimiento considerable al reducir en un 50 % el dataset de MovieLens, mientras que el Grupo 2 (que incluye FunkSVD, BiasedMF y Popularity), presentó pérdidas mucho menores, siendo más eficientes desde el punto de vista energético.

Cuadro 1: Pérdida de rendimiento (NDCG@10) al reducir 50 % del tamaño del dataset.

Grupo	MovieLens ↓50 %	Amazon ↓50 %
Grupo 1	↓ 50 %	no reportado
Grupo 2	↓ 23 %	↓ 13 %

Además de la caída en precisión, el trabajo destaca el impacto ambiental: se logró reducir el tiempo de entrenamiento en un 72 %, con un ahorro estimado de 27.4 kg de CO_2 al aplicar reducciones de tamaño. Esta comparación evidencia que no todos los modelos se ven afectados por igual al entrenar con menos datos, lo que abre la puerta a buscar configuraciones más sustentables sin sacrificar calidad.

Si bien nuestro estudio toma este enfoque como punto de partida, nuestro aporte se enfoca en entregar un análisis más detallado: experimentar con más puntos intermedios de reducción y expansión del dataset, y realizar una estimación más precisa y comparativa de las emisiones de carbono para cada modelo. De este modo, buscamos entender no solo cómo cambia la calidad de la recomendación, sino también cuál es el costo ambiental real asociado a cada decisión experimental.

2.3. CodeCarbon

Un pilar fundamental para cualquier estudio en Green AI es la capacidad de medir de forma fiable el impacto ambiental de los experimentos. En este contexto, el trabajo de Lacoste et al. (2019) que introduce CodeCarbon es de suma relevancia. CodeCarbon es una herramienta de software de código abierto diseñada para estimar las emisiones de CO_2 producidas por el uso de recursos computacionales. El paquete funciona monitoreando el consumo de energía de los componentes de hardware (CPU, GPU) durante la ejecución de un bloque de código. Luego, combina esta información con datos geográficos para determinar la intensidad de carbono de la red eléctrica local, la cual varía significativamente según la matriz energética de la región (e.g., predominancia de combustibles fósiles vs. energías renovables). El resultado es una estimación en kilogramos de CO_2 equivalentes (kg CO_2-eq). Para este estudio, CodeCarbon es la herramienta elegida por su facilidad de integración en Python y por proporcionar una metodología transparente y estandarizada para cuantificar la huella de carbono de nuestros modelos, permitiendo una comparación directa y objetiva de su eficiencia ambiental.

3. DATASET

Para este estudio se utiliza la versión del dataset MovieLens 1M, desarrollado por el GroupLens Research Group [2]. Estos conjuntos de datos son ampliamente utilizados en investigación de sistemas de recomendación debido a su estructura clara y a la disponibilidad pública de información anonimizada.

El conjunto MovieLens 1M cuenta con un millón de calificaciones de 6.040 usuarios sobre 3.952 películas. En donde cada usuario ha calificado al menos 20 películas. Los datos se organizan bajo 4 variables principales: ID del usuario (userId), ID de la película (movieId), y calificación entregada por el usuario en una escala de 1 a 5 (rating) y una marca temporal (timestamp).

En este trabajo se utilizará principalmente el dataset para probar distintos métodos base, y para validar escalabilidad y observar cómo cambian tanto la calidad de la recomendación como las emisiones de carbono. Además, se generarán subconjuntos del dataset aplicando reducciones sobre el conjunto original, manteniendo la distribución de calificaciones lo más equilibrada posible. Esto permitirá evaluar el impacto del tamaño del dataset en el rendimiento de cada modelo y su huella ambiental.

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931
5	1	70	3.0	964982400
6	1	101	5.0	964980868
7	1	110	4.0	964982176
8	1	151	5.0	964984041
9	1	157	5.0	964984100

Figura 2: Estructura del dataset MovieLens

4. METODOLOGÍA

La metodología empleada en este estudio fue simple y se dividió en pasos bien definidos. En primer lugar, se descargaron los archivos comprimidos correspondientes a los datasets MovieLens 1M, disponibles públicamente en el sitio de GroupLens. Posteriormente, se instalaron las dependencias necesarias para la ejecución de los experimentos, incluyendo las librerías `Surprise` para la implementación de modelos de recomendación clásicos, y `CodeCarbon`, que permite estimar las emisiones de carbono asociadas al uso computacional.

Una vez configurado el entorno, se procedió a cargar y preprocesar los datasets para adaptarlos al formato requerido por los algoritmos. Luego, se definió una función encargada de entrenar cada modelo, calcular su métrica de rendimiento (NDCG@10) y registrar las emisiones estimadas por CodeCarbon durante el proceso de entrenamiento.

Finalmente, se ejecutó esta función utilizando distintos algoritmos disponibles en Surprise, tanto con el dataset completo como con versiones reducidas de los datos. Todos los resultados fueron almacenados y posteriormente analizados para entender la relación entre la precisión de los modelos y sus respectivas emisiones de carbono.

5. ANÁLISIS DE RESULTADOS

En esta sección, se analizan los resultados obtenidos de la experimentación. Se evalúa el comportamiento de los modelos UserKNN, ItemKNN y SVD en tres ejes principales: la calidad del ranking (NDCG@10), el costo ambiental (emisiones de CO_2-eq) y la relación entre ambos. Inicialmente, se dispone con los siguientes resultados de rendimiento y emisiones para los distintos modelos y distintos tamaños:

Cuadro 2: Resultados de precisión (NDCG@10) y emisiones de CO_2-eq para distintos modelos y fracciones del dataset.

Modelo	Fracción	RMSE	NDCG@10	Duración (s)	CO_2-eq (kg)
SVD	0.10	0.950	0.0461	161.0	0.000004
UserKNN	0.10	1.037	0.0044	667.6	0.000007
ItemKNN	0.10	1.102	0.0043	347.5	0.000002
LightFM	0.10	—	0.0578	12.1	0.000013
SVD	0.25	0.933	0.0391	179.1	0.000005
UserKNN	0.25	0.996	0.0003	1596.1	0.000014
ItemKNN	0.25	1.053	0.0002	785.7	0.000006
LightFM	0.25	—	0.0992	18.4	0.000000
SVD	0.50	0.913	0.0529	193.8	0.000010
UserKNN	0.50	0.988	0.0001	2778.9	0.000038
ItemKNN	0.50	1.025	0.0008	1380.1	0.000014
LightFM	0.50	—	0.1466	27.8	0.000064
SVD	0.75	0.892	0.0669	205.2	0.000018
UserKNN	0.75	0.982	0.0001	4040.6	0.000078
ItemKNN	0.75	1.009	0.0096	1878.4	0.000029
LightFM	0.75	—	0.2081	37.0	0.000000
SVD	1.00	0.873	0.0882	229.2	0.000043
UserKNN	1.00	0.979	0.0001	5173.0	0.000167
ItemKNN	1.00	1.003	0.0265	2417.9	0.000062
LightFM	1.00	—	0.3116	46.3	0.000119

5.1. Comparación de Rendimiento (NDCG@10)

El primer análisis se centra en la calidad de las recomendaciones. Se espera que, a medida que aumenta la fracción del dataset utilizado para el entrenamiento, la métrica NDCG@10 mejore para todos los modelos, ya que disponen de más información para aprender las preferencias de los usuarios. Los resultados son los siguientes:

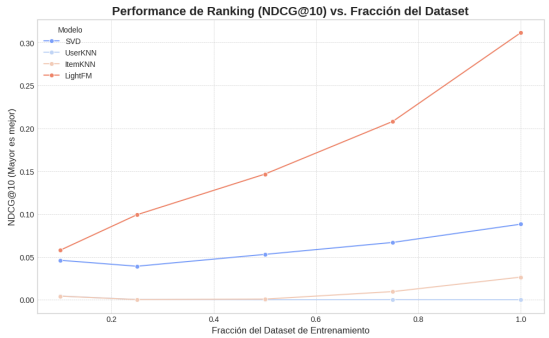

Figura 3: NDCG@10 según tamaño del dataset y modelo

Se puede notar que el modelo LightFM mejora su rendimiento de forma pronunciada y cuasi-lineal con el aumento del tamaño del dataset, alcanzando un valor aproximado de 0.3116 al utilizar el 100 % de los datos, en donde a medida que dispone de más interacciones usuario-item, el modelo puede ajustar mejor sus vectores latentes y capturar patrones más sutiles de preferencia, lo que se traduce en una mejora considerable en la calidad del ranking.

Luego, de forma mucho menos pronunciada, se observa un crecimiento en el modelo SVD, que escala desde aproximadamente 0.05 hasta alrededor de 0.088 en NDCG@10. Si bien también se basa en

representaciones latentes, su estructura es más rígida, lo que limita su capacidad para capturar relaciones complejas. A pesar de esto, el aumento en la cantidad de datos permite estabilizar mejor los factores latentes y generalizar con mayor precisión, lo que justifica la mejora observada.

Por último, los modelos ItemKNN y UserKNN muestran rendimientos bajos y planos. Esto se debe a que ambos modelos se basan exclusivamente en la similitud entre ítems o usuarios, en donde al tener datos dispersos, la incorporación de nuevas observaciones no aporta mejoras sustanciales, ya que la calidad de las recomendaciones depende más de la densidad local que del volumen global de datos.

5.2. Análisis de Emisiones de Carbono (CO_2-eq)

Las emisiones de carbono, directamente ligadas al tiempo de ejecución (Gráfico **??**), revelan enormes diferencias en la eficiencia computacional. Con el 100 % del dataset, el modelo más costoso fue **UserKNN** con emisiones de **0.000167 kg CO_2-eq**, mientras que el más eficiente fue **SVD** con solo **0.000043 kg CO_2-eq**. Esto representa una diferencia del **288 %**, demostrando que la elección del algoritmo tiene un impacto ambiental drástico. LightFM, a pesar de su alto rendimiento, se mantiene eficiente en tiempo (46.3 segundos), aunque su huella de carbono (0.000119 kg) es mayor que la de SVD.

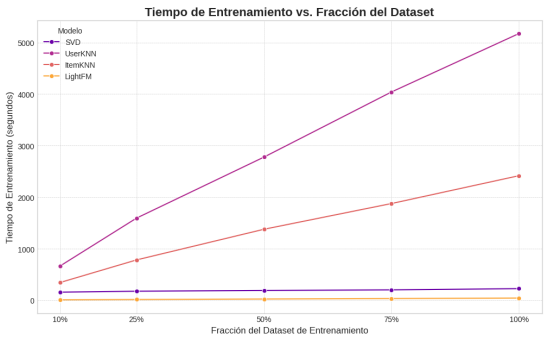

Figura 4: Tiempo de entrenamiento vs Tamaño del dataset

5.3. Análisis del Trade-off: NDCG@10 vs. Emisiones

El núcleo de este estudio es visualizar el balance entre el costo y el beneficio. La Figura 5 muestra este trade-off.

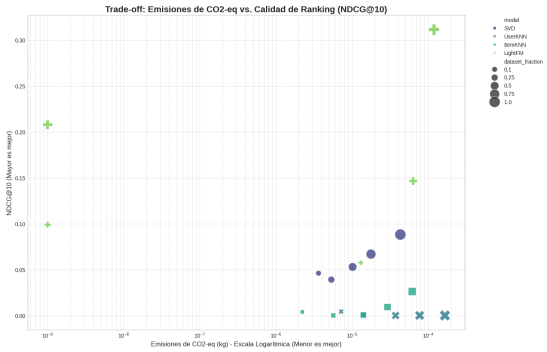

Figura 5: Trade-off entre emisiones y NDCG@10

Se evidencia un trade-off claro entre ambos factores. El modelo SVD, por ejemplo, logra un NDCG@10 cercano a 0.053 utilizando solo el 50 % del dataset, con una emisión estimada de 0.000010 kg de CO_2-eq. Sin embargo, al incrementar la fracción de datos al 100 %, su rendimiento sube marginalmente a 0.0882, mientras que sus emisiones aumentan considerablemente a 0.000043 kg, lo que representa un incremento de más del 300 % en emisiones para una mejora relativa de apenas un 66 % en precisión.

Este fenómeno se repite, con mayor notoriedad, en modelos como LightFM, donde el uso del 75 % del dataset ya entrega un NDCG@10 de 0.2081 con emisiones nulas estimadas, mientras que escalar al 100 % eleva la métrica a 0.3116, pero con un costo de 0.000119 kg de CO_2-eq. Esta diferencia sugiere que las mejoras marginales en precisión con datasets completos implican aumentos significativos en emisiones.

A partir de estos resultados, se identifica un posible punto óptimo en términos de eficiencia ambiental y rendimiento: entrenar el modelo SVD con un 50 % del dataset o LightFM con un 75 %, ya que ofrecen un buen equilibrio entre calidad de recomendación y emisiones generadas. Superar estos niveles entrega beneficios en precisión, pero a costa de un crecimiento desproporcionado en el impacto ambiental.

6. CONCLUSIONES

Este trabajo ha evaluado de forma empírica el trade-off entre el rendimiento y la huella de carbono en sistemas de recomendación. Mediante la experimentación con modelos de filtrado colaborativo sobre fracciones crecientes del dataset MovieLens 1M, se han obtenido las siguientes conclusiones clave:

1. El Trade-off es Real y Medible: Se confirma la existencia de una relación directa pero no siempre lineal entre la cantidad de datos, la precisión del modelo (NDCG@10) y las emisiones de CO_2-eq. Incrementar el tamaño del dataset para buscar mejoras marginales en la precisión resulta en un costo ambiental desproporcionado.
2. La Elección del Modelo es Crucial: No todos los algoritmos escalan de la misma manera. En nuestros experimentos, **SVD** demostró ser el más equilibrado, mientras que **UserKNN** fue significativamente menos eficiente, ya sea en términos de

rendimiento por emisión o por su alto costo computacional absoluto.

3. Identificación de un "Punto Verde": Los resultados sugieren que no siempre es necesario utilizar el 100 % de los datos disponibles. Para este escenario, entrenar **SVD con un 50 %** del dataset permite alcanzar aproximadamente**60 % del rendimiento máximo**, pero con solo una fracción de las emisiones, presentando una estrategia de Green RecSys viable.

6.1. Limitaciones y Trabajo Futuro

Este estudio se limitó a modelos clásicos de filtrado colaborativo y al dataset MovieLens 1M. Como pasos futuros, sería valioso extender este análisis para incluir modelos basados en redes neuronales profundas (como NCF o VAE) y arquitecturas más complejas. Asimismo, medir el ciclo de vida completo, incluyendo las emisiones generadas durante la inferencia, proporcionaría una visión aún más completa del impacto real de estos sistemas.

REFERENCIAS

[1] Ardalan Arabzadeh, Tobias Vente, and Joeran Beel. 2024. Green Recommender Systems: Optimizing Dataset Size for Energy-Efficient Algorithm Performance. arXiv:2410.09359 [cs.IR] https://arxiv.org/abs/2410.09359

[2] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2015), 1–19. https://doi.org/10.1145/2827872

[3] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiao-Ming Wu. 2024. Benchmarking News Recommendation in the Era of Green AI. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion)*. ACM, Singapore. https://doi.org/10.1145/3589335.3651472

[4] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. *arXiv preprint arXiv:1907.10597* (2019). https://arxiv.org/abs/1907.10597