



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y
CIENCIAS AMBIENTALES
TALLER GRUPAL



1. Datos Informativos

1.1. Módulo: 4

1.2. Nivel: Séptimo

1.3. Fecha: 11/5/2025

1.4. Nombres: Aupas Antony, Baraja Cristian, Basantes Geovanny

1.5. Tema: Métricas de productividad del desarrollador de sistemas de NLP

2. Objetivos

- Comprender qué son las métricas de productividad en desarrollo de software.
- Identificar métricas útiles y riesgosas.
- Reflexionar sobre el uso ético y estratégico de estas métricas.
- Analizar casos reales y construir criterios de aplicación.

3. Contenido

3.1. Definición de métricas de productividad

Las métricas de productividad en el ámbito del desarrollo de software son parámetros usados para medir el desempeño, la calidad y la eficacia de los grupos o programadores individuales. Estas métricas pueden fundamentarse en datos cuantitativos (como el número de líneas de código) o cualitativos (como el valor aportado al usuario).

3.2. Tipos de métricas

3.2.1. Métricas útiles vs. métricas engañosas

Las métricas beneficiosas son aquellas que facilitan la mejora, mientras que las engañosas pueden ocasionar incentivos erróneos o crear presión innecesaria. Un ejemplo es el medir solo líneas de código, lo que podría motivar a producir código superfluo.

De acuerdo con Kerzner (2017), las métricas efectivas deben ser comprensibles, accionables y estar alineadas con los objetivos estratégicos del equipo.

3.2.2. Buenas prácticas

La incorporación de indicadores tanto numéricos como descriptivos nos ofrece una perspectiva más integral del desempeño de los programadores, puesto que no todo puede expresarse en cifras. Es crucial que estas métricas se utilicen como un recurso para el avance, en lugar de como métodos para sancionar o ejercer presión. Asimismo, cada resultado debe ser analizado en su entorno, tomando en cuenta aspectos como la dificultad del proyecto o los retos superados. Fomentar la claridad y la interpretación compartida de la información permite que cada integrante del equipo se sienta incluido en el proceso y promueve un clima de mejora continua en lugar de rivalidad excesiva.

3.2.3. Herramientas disponibles

Hay varias herramientas que permiten registrar y analizar las métricas de productividad de un desarrollador y se describen a continuación:

- **GitHub/GitLab:** Para commits, problemas y solicitudes de extracción.

- **Jira:** Para medir velocidad y seguimiento de tareas.
- **SonarQube:** Para evaluar la calidad del código.
- **CodeClimate:** Para análisis de mantenibilidad y cobertura de pruebas.

3.3. Análisis de Caso

En el desarrollo de sistemas basados en modelos de Procesamiento de Lenguaje Natural (NLP), como los orientados a la explicabilidad de decisiones automáticas, medir la productividad del equipo de desarrollo representa un desafío técnico y ético. ¿Cómo evaluar el desempeño de quienes trabajan en modelos que no solo deben funcionar, sino también ser comprensibles para los usuarios? La necesidad de métricas claras y bien definidas es clave no solo para gestionar proyectos de IA, sino para asegurar su transparencia, responsabilidad y valor social.

Este taller se propone reflexionar sobre qué significa ser productivo en el contexto de un proyecto de explicabilidad en NLP. A través de una revisión crítica de métricas comunes —como líneas de código, velocidad de entrega, número de experimentos realizados o mejoras en interpretabilidad—, se busca distinguir entre indicadores útiles y aquellos que pueden ser engañosos o contraproducentes. Además, se analizarán casos reales y se diseñarán propuestas de métricas adaptadas a equipos que trabajan en inteligencia artificial explicable, con énfasis en su uso estratégico y ético.

El objetivo es que los participantes desarrollen criterios sólidos para medir la productividad de forma coherente con los valores del proyecto, considerando no solo la eficiencia técnica, sino también el impacto, la calidad del código, la

colaboración y el aporte al entendimiento de los modelos NLP por parte de los usuarios finales.

3.3.1. Aplicación de Métricas de Productividad en Proyectos de NLP Explicable

Las contribuciones realizadas de manera individual al proyecto contabilizando los cambios realizados al repositorio remoto anclado en github.



3.3.2. Métricas utilizadas:

Precisión: Mide la proporción de predicciones correctas sobre el total de predicciones realizadas.

- Exhaustividad (Recall): Mide la proporción de predicciones correctas sobre el total de instancias positivas.
- Puntaje F1: Métrica que combina precisión y exhaustividad en una sola medida.
- Área bajo la curva ROC (AUC-ROC): Mide la capacidad del modelo para distinguir entre clases positivas y negativas.

3.3.3. Problemas y beneficios identificados:

- Dificultad para interpretar las decisiones del modelo, debido a la naturaleza compleja de las redes neuronales.

- Posible sesgo en los datos de entrenamiento, lo que puede llevar a predicciones sesgadas.
- Necesidad de grandes conjuntos de datos etiquetados para un buen desempeño.

3.3.4. Beneficios:

- Alta capacidad de aprendizaje y adaptación a tareas complejas de procesamiento de lenguaje natural.
- Posibilidad de realizar predicciones y clasificaciones con alta precisión en diversas aplicaciones.
- Potencial para automatizar tareas que antes requerían intervención humana.

3.3.5. Mejoras:

Implementar técnicas de explicabilidad del modelo, como SHAP o LIME, para comprender mejor las decisiones. Realizar un análisis exhaustivo de los datos de entrenamiento y aplicar técnicas de balanceo y limpieza.

Explorar arquitecturas de redes neuronales más interpretables, como las redes neuronales convolucionales. Combinar el modelo de NPL con reglas de negocio o conocimiento de expertos para mejorar la confiabilidad.

3.3.6. Construcción de una métrica ideal:

Una métrica ideal para evaluar la explicabilidad de un modelo de NPL debería considerar los siguientes aspectos:

1. **Facilidad de interpretación:** Medir qué tan fácil es para los usuarios finales comprender las decisiones y razonamientos del modelo. Fidelidad al modelo:

Evaluar qué tan bien la explicación captura el comportamiento real del modelo, sin simplificaciones excesivas. Robustez a sesgos: Considerar si la explicación es consistente y no se ve afectada por factores externos o sesgos en los datos. Granularidad de la explicación: Permitir analizar tanto las predicciones a nivel de instancia como las tendencias generales del modelo.

2. Integración con el flujo de trabajo: Valorar qué tan bien la explicación se alinea con los procesos y necesidades del negocio.

3. Escalabilidad: Evaluar si la métrica puede aplicarse de manera eficiente a modelos de gran tamaño y complejidad.

4. Conclusiones

El análisis de la explicabilidad de los modelos de Procesamiento de Lenguaje Natural (NPL) es un aspecto fundamental para garantizar la confiabilidad y aceptación de estas tecnologías en aplicaciones del mundo real. Las métricas comúnmente utilizadas, como precisión, exhaustividad y puntaje F1, proporcionan una evaluación cuantitativa del desempeño del modelo, pero no abordan la cuestión de la interpretabilidad y transparencia de las decisiones. Para lograr una métrica ideal de explicabilidad, es necesario considerar diversos factores, como la facilidad de interpretación para los usuarios finales, la fidelidad al comportamiento real del modelo, la robustez a sesgos, la granularidad de la explicación, la integración con los flujos de trabajo del negocio y la escalabilidad a modelos de gran complejidad. Una métrica que combine estos aspectos en una puntuación general permitiría a las organizaciones seleccionar los modelos de NPL más



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y
CIENCIAS AMBIENTALES



explicables y alineados con sus necesidades, fomentando la confianza y la adopción de estas tecnologías en aplicaciones críticas.



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y
CIENCIAS AMBIENTALES



Taller: Métricas de rendimiento del software de sistemas de NLP

1. Datos Informativos

1.1. Módulo: 4

1.2. Nivel: Séptimo

1.3. Fecha: 11/5/2025

1.4. Nombres: Aupas Antony, Baraja Cristian, Basantes Geovanny

1.5. Tema: Métricas de rendimiento del software de sistemas de NLP

2. Objetivos

- Comprender los conceptos clave del rendimiento del software orientado a sistemas de NLP.
- Identificar y aplicar métricas de rendimiento específicas en proyectos de procesamiento de lenguaje natural.
- Analizar resultados y tomar decisiones basadas en métricas para mejorar modelos y servicios NLP.
- Explorar herramientas de medición del rendimiento enfocadas en servicios de inteligencia artificial y NLP.

3. Contenido

3.1. Introducción

3.1.1. ¿Qué entendemos por rendimiento del software en sistemas NLP?

El rendimiento en un sistema de NLP se refiere a su capacidad para procesar y analizar texto en lenguaje natural de forma eficiente, precisa y escalable. Esto incluye el tiempo que tarda un modelo en inferir, la precisión de sus respuestas, la latencia en sistemas de

chatbot o búsqueda semántica, y el consumo de recursos computacionales (GPU, RAM).

3.1.2. ¿Velocidad siempre significa buen rendimiento?

No necesariamente. En NLP, una inferencia rápida puede sacrificar precisión o comprensión contextual. El rendimiento debe equilibrar velocidad, exactitud del modelo, consumo de recursos y experiencia del usuario, especialmente en aplicaciones como asistentes virtuales, analizadores de sentimientos o motores de búsqueda inteligente.

3.2. Métricas clave de rendimiento en sistemas NLP

En lugar de analizar un sitio web, aplicamos las métricas al contexto de un prototipo NLP para una tienda en línea, que utiliza un chatbot para atención al cliente y un motor de recomendación semántico. Métricas para considerar:

- **Tiempo de inferencia:** Tiempo que tarda el modelo NLP (por ejemplo, BERT o GPT) en generar una respuesta.
- **Precisión / Recall / F1-score:** Métricas esenciales para tareas de clasificación de texto o detección de entidades.
- **Latencia del sistema:** Tiempo total desde que el usuario hace la pregunta hasta que recibe la respuesta.
- **Uso de CPU/GPU y memoria RAM:** Monitoreo del consumo de recursos por el modelo en producción.
- **Throughput:** Cantidad de peticiones procesadas por segundo, relevante para sistemas NLP en tiempo real.

- **Tasa de error de predicción:** Número de respuestas erróneas o irrelevantes en relación al total.
- **Escalabilidad:** Capacidad del sistema NLP de adaptarse a un mayor número de usuarios o datos.

3.3. Herramientas para medir rendimiento en sistemas NLP

- **Lighthouse** puede seguir utilizándose para evaluar el frontend donde se despliega el sistema NLP (por ejemplo, chatbot web).
- Para el backend de NLP se recomiendan:
- **TensorBoard:** para visualizar métricas durante el entrenamiento de modelos.
- **Prometheus + Grafana:** para monitorear el rendimiento en producción (CPU, RAM, latencia).
- **Locust** o **JMeter:** para pruebas de carga y estrés de API de inferencia NLP.
- **Hugging Face Evaluate:** para métricas específicas como accuracy, BLEU, ROUGE, etc.

3.4. Evaluación de un sistema NLP prototipo

Analizamos el sistema "NPL-ShopBot", un chatbot que responde preguntas de productos. Se recopilan las siguientes métricas:

- **Tiempo de inferencia promedio:** 1.5 segundos.
- **Precisión en respuestas sobre disponibilidad de productos:** 87%.
- **Uso de CPU/GPU:** 45% CPU, 70% GPU en cargas altas.
- **Latencia total con red:** 2.2 segundos.
- **Tasa de satisfacción del usuario (pruebas):** 82% positiva.

Discusión:

El rendimiento general es aceptable, pero el tiempo de inferencia podría mejorarse

usando modelos optimizados o destilados. Además, se sugiere el uso de técnicas de batch inference y caching para respuestas comunes.

3.5. Buenas prácticas para mejorar el rendimiento en NLP

- **Destilación de modelos (model distillation):** Usar versiones ligeras de modelos pesados (ej. DistilBERT).
- **Uso de batch inference:** Procesar múltiples consultas a la vez.
- **Optimización con ONNX / TensorRT:** Para mejorar la velocidad de inferencia en GPU.
- **Uso de embeddings precomputados:** Para respuestas frecuentes o ranking de productos.
- **Despliegue con servidores optimizados (FastAPI, TorchServe, etc.).**

5. Práctico – Evaluación de rendimiento de un sistema NLP de ejemplo (como TEMU pero con chatbot)

5.1 Rendimiento (Performance):

Puntaje simulado de 76. La latencia en hora pico es de 3s, lo que afecta la fluidez del chatbot. Se recomienda usar técnicas de optimización de inferencia y compresión del modelo.

5.2 Accesibilidad:

El frontend del sistema tiene un puntaje de 88 en accesibilidad (etiquetas semánticas, navegación por teclado), pero requiere ajustes en el contraste y textos alternativos.

5.3 Buenas prácticas:

Puntaje de 96. Se cumplen estándares como HTTPS, API segura, separación de componentes y manejo de errores.

5.4 Optimización SEO:

Aunque el foco del sistema es NLP, el sitio tiene un SEO de 92, útil para la página que presenta el sistema. Se sugiere mejorar descripciones de contenido.

6. Conclusión

La medición del rendimiento en sistemas de NLP es fundamental para garantizar respuestas rápidas, precisas y eficientes. A lo largo del taller se comprendió cómo las métricas tradicionales del software pueden adaptarse al contexto NLP, y cómo herramientas específicas como TensorBoard o Prometheus permiten monitorear modelos en producción.

Evaluamos un sistema de atención al cliente basado en NLP y descubrimos que factores como el tiempo de inferencia, el uso de recursos y la precisión de las respuestas afectan directamente la experiencia del usuario. Se evidenció que no basta con tener un modelo entrenado, sino que su comportamiento en producción debe optimizarse continuamente.

Finalmente, aplicar buenas prácticas como distilación de modelos, uso de infraestructura adecuada y monitoreo constante contribuye a sistemas NLP sostenibles, escalables y centrados en el usuario, fundamentales para su adopción en entornos reales.

7. Cierre y reflexión

Después de analizar los datos proporcionados por la herramienta PageSpeed Insights, se puede concluir que el sitio web de americanstore04.shop tiene un rendimiento general aceptable, pero existen varias oportunidades de mejora que

podrían tener un impacto significativo en la experiencia del usuario y en la efectividad del sitio web para lograr los objetivos de negocio. Los principales aspectos a mejorar son el tiempo de carga de la página, la accesibilidad, el seguimiento de las mejores prácticas de desarrollo web y la optimización de SEO. El análisis del rendimiento de la página web es un ejercicio crucial para entender el estado actual del sitio y poder identificar áreas de mejora. Estos datos proporcionan información valiosa que puede guiar los esfuerzos de optimización y mejorar la experiencia general del usuario. Es importante tener en cuenta que el rendimiento del software es un aspecto que debe ser monitoreado y mejorado constantemente, ya que a medida que las tecnologías y las expectativas de los usuarios evolucionan, es necesario mantener un proceso de revisión y actualización para asegurar que el sitio web siga siendo eficiente, accesible y competitivo en el mercado.

. Referencias bibliográficas

Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean*



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y
CIENCIAS AMBIENTALES



Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution.

Kerzner, H. (2017). *Project Management Metrics, KPIs, and Dashboards: A Guide to Measuring and Monitoring Project Performance* (3rd ed.). Wiley.

Smith, T. (2020). *Productivity in Software Engineering: Beyond the Numbers*. IEEE

Software, 37(2), 13–17. <https://doi.org/10.1109/MS.2020.2967194>

Wiggins, A. (2022). *Ethical Use of Metrics in Software Development*.

Communications of the ACM, 65(4), 24–27. <https://doi.org/10.1145/3514188>