Tecnológico de Monterrey

Avance 3. Baseline

A01150742 José Ovalle Alvarado

A01794879 Carlos de Jesús Méndez Tornero

A01104775 Alejandro Munguía Salazar

Proyecto Integrador | Fecha: 18/05/2025

1. Introducción

La morosidad en los pagos visto como desde el punto de vista de técnicas de machine learning representa un desafío crítico, el desarrollo de un modelo predictivo capaz de identificar con anticipación a los residentes con mayor probabilidad de incurrir en impagos se convierte en una gran herramienta para MiCoto. Sin embargo, antes de implementar soluciones complejas, es fundamental establecer un modelo de referencia, conocido como *baseline*, que sirva como punto de partida para evaluar la viabilidad del problema y sentar las bases para futuras mejoras.

En este avance 3 se tiene como objetivo principal construir y evaluar este modelo *baseline*, el cual no solo permitirá determinar si los datos disponibles contienen información suficiente para predecir la morosidad, sino que también proporcionará una métrica inicial contra la cual comparar el desempeño de modelos más sofisticados. La importancia de este paso radica en su capacidad para ofrecer una primera aproximación realista, evitando caer en soluciones innecesariamente complejas o, por el contrario, subestimar el potencial de los datos.

Al presentar resultados concretos obtenidos con un modelo simple, se establece un marco de referencia claro sobre lo que se puede lograr con los datos actuales, facilitando la toma de decisiones informadas respecto a la inversión de recursos en desarrollos posteriores. Si el *baseline* demuestra un rendimiento aceptable, se justifica avanzar hacia modelos más complejos; si, por el contrario, los resultados son cercanos al azar, será necesario replantear el enfoque o revisar la calidad de los datos.

A través de este análisis, se busca no solo cumplir con los requisitos técnicos del proyecto, sino también sentar las bases para un desarrollo iterativo y fundamentado de soluciones predictivas. El *baseline* no es solo un primer paso, sino una herramienta de diagnóstico que guiará las decisiones técnicas y estratégicas en las fases posteriores del proyecto.

2. Selección del algoritmo

Debido a la limitación de los datos durante el marco de experimentación se eligió a la **Regresión Logística** como *baseline* debido a varias ventajas clave que lo convierten en la opción ideal para una primera aproximación al problema de predicción de morosidad. Este algoritmo destaca por su **simplicidad conceptual cosa deseada** debido a la poca agrupación de datos en la variable objetivo así como la baja cantidad de datos disponibles, lo que permite una implementación rápida y una evaluación directa de la relación entre las variables predictoras y la variable objetivo.

El problema de clasificación binaria (morosidad vs. no morosidad) que aborda este estudio se ajusta perfectamente a las capacidades de la Regresión Logística. Este algoritmo estima probabilidades mediante una función sigmoide, lo que lo hace particularmente adecuado

para casos donde se necesita no solo una clasificación, sino también una medida de la certeza detrás de cada predicción. Esta característica es crucial en el contexto de morosidad, donde poder asignar un nivel de riesgo a cada residente (en lugar de solo una etiqueta binaria) permite priorizar acciones de cobranza de manera más estratégica.

Para contextualizar el desempeño del baseline y evaluar el potencial de mejora con enfoques más sofisticados, se implementaron cuatro modelos adicionales: **Random Forest**, **Máquinas de Soporte Vectorial (SVM)**, **Naive Bayes** y **Redes Neuronales (MLP)**. Esta comparación sistemática sirve para:

1. **Establecer un marco de referencia realista**: Al contrastar la Regresión Logística con modelos más complejos, podemos determinar si el problema requiere realmente de técnicas avanzadas o si, por el contrario, un enfoque simple es suficiente para alcanzar un rendimiento aceptable.

2. **Identificar ganancias marginales**: La comparación permite cuantificar cuánta mejora adicional podría obtenerse con modelos más complejos. Por ejemplo, si un Random Forest mejora el accuracy en solo un 2-3% respecto a la Regresión Logística, podría no justificarse su mayor complejidad computacional y menor interpretabilidad.

3. **Diagnosticar limitaciones de los datos**: Si todos los modelos, independientemente de su complejidad, muestran un rendimiento similar, esto podría indicar que las variables disponibles tienen un poder predictivo limitado, señalando la necesidad de recolectar datos adicionales o de mayor calidad.

Cada uno de estos modelos alternativos fue seleccionado por características específicas:

- **Random Forest**: Excelente para capturar interacciones no lineales entre variables sin requerir un preprocesamiento exhaustivo.

- **SVM**: Potente en espacios de alta dimensionalidad y cuando existe una clara separación entre clases.

- **Naive Bayes**: Eficiente con datos escasos y como referencia para problemas probabilísticos.

- **MLP**: Capacidad de modelar relaciones complejas, pero con mayor riesgo de sobreajuste.

Esta batería de modelos no solo enriquece el análisis comparativo, sino que también proporciona información valiosa sobre la naturaleza del problema. Por ejemplo, si los modelos no lineales (como Random Forest o MLP) superan significativamente a la Regresión Logística, esto sugeriría que las relaciones entre variables son más complejas de lo que puede capturar un modelo lineal. Por el contrario, si todos los modelos muestran un rendimiento similar, esto reforzaría la utilidad del baseline como solución suficiente.

La implementación de estos modelos se realizó siguiendo buenas prácticas de machine learning, incluyendo:

- **Preprocesamiento consistente**: Todos los modelos utilizaron exactamente las mismas transformaciones de datos para garantizar comparabilidad.
- **Validación cruzada**: Para obtener estimaciones robustas del rendimiento y evitar sobreoptimismo.
- **Ajuste de hiperparámetros básico**: Optimización mínima para reflejar el escenario real donde un baseline no recibe un tuning exhaustivo.

Este enfoque metodológico no solo valida la elección del baseline, sino que también proporciona una hoja de ruta clara para iteraciones futuras del modelo, indicando qué direcciones técnicas podrían ser más prometedoras para mejorar el rendimiento predictivo. Los resultados de esta comparación se presentan en detalle en la sección de Evaluación de Modelos, donde se analizan tanto las métricas de rendimiento como los trade-offs entre complejidad y mejora predictiva.

3. Determinación y Selección

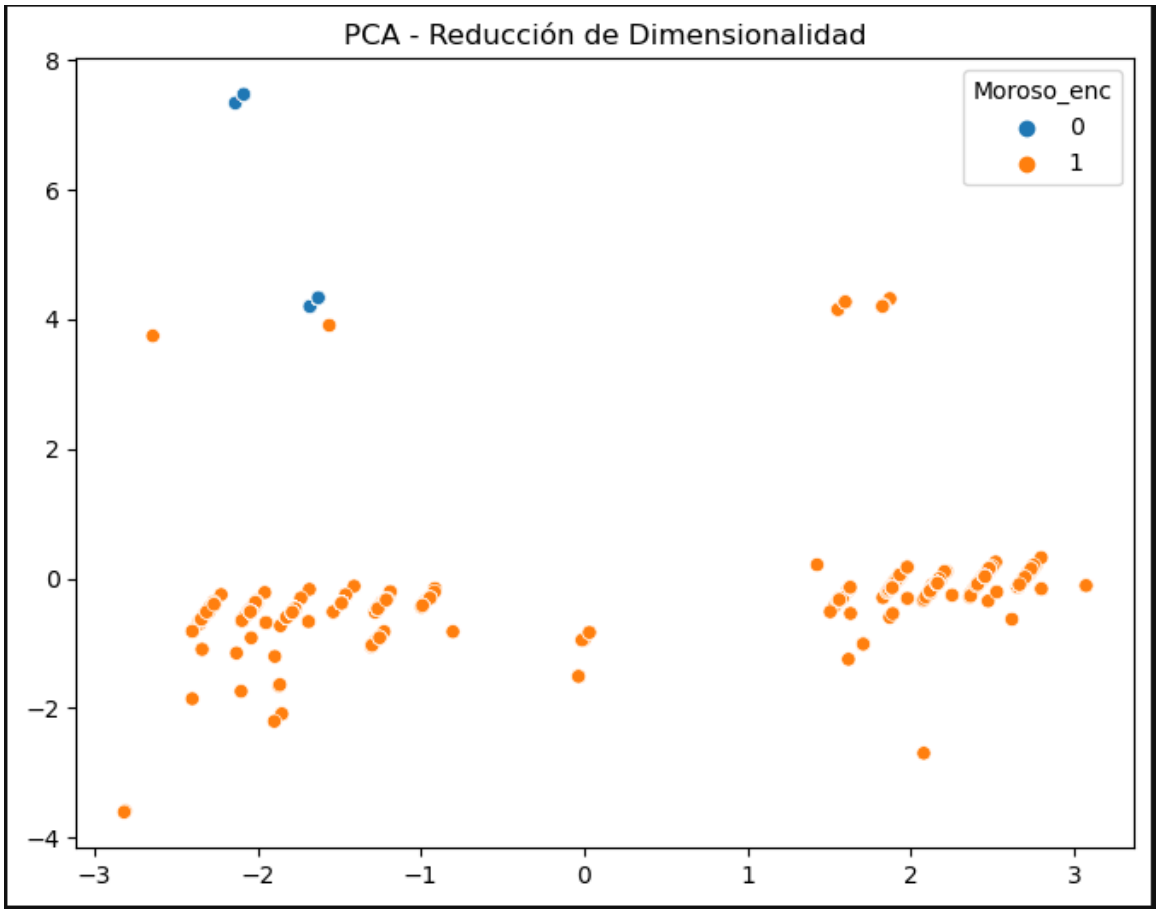
El proceso de identificación y selección de las variables más relevantes para nuestro modelo predictivo de morosidad representó desde el avance 2 una etapa fundamental en el desarrollo del proyecto. Para garantizar que el modelo se construyera sobre las bases más sólidas posibles y evitando variables irrelevantes, implementamos una estrategia integral que combinó técnicas avanzadas de selección de características con métodos de visualización multidimensional.

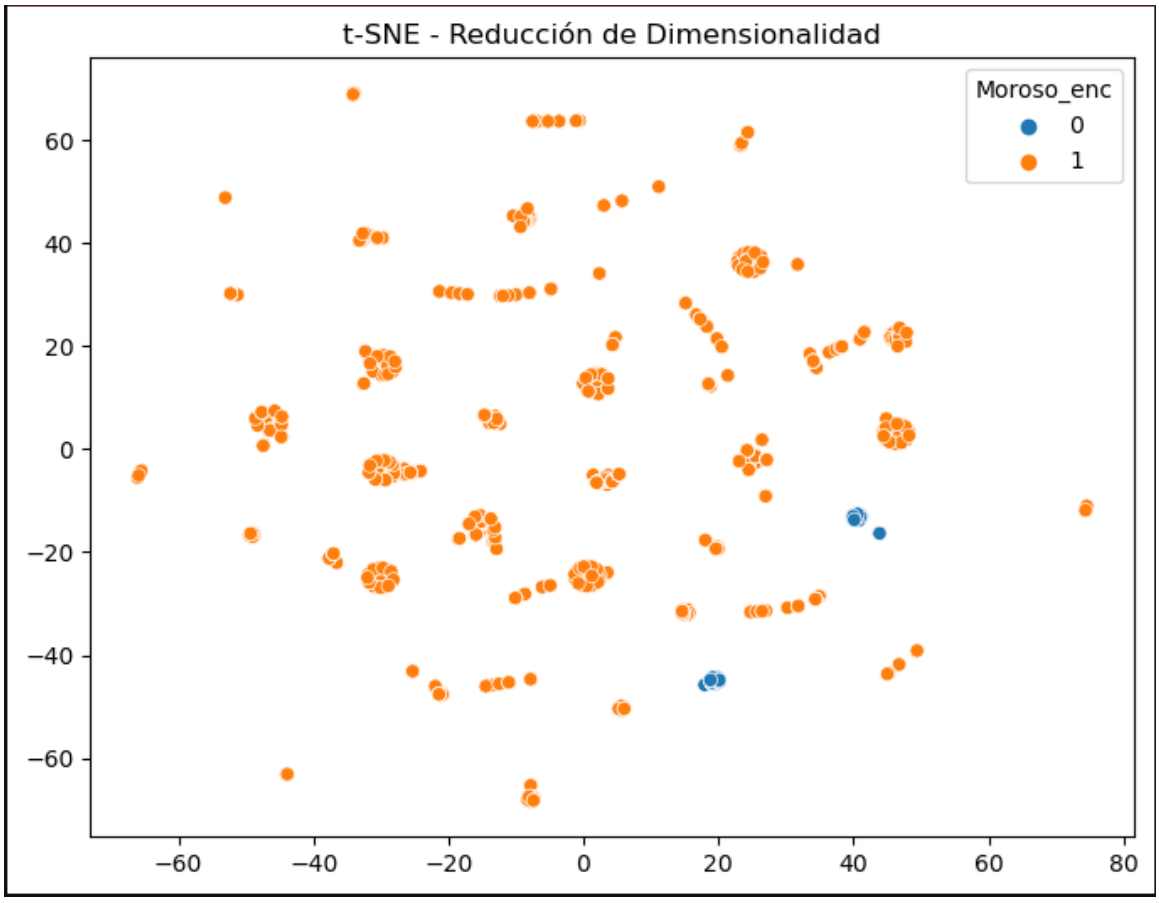
Previo a la experimentación, hay evidencias de que, a causa de las propiedades de los datos, el hecho de que hayan clases desbalanceadas, la dependencia lineal de las variables mas importantes así como una limitada historia de morosidad los modelos relativamente complejos se sobreestimarán y esto hace necesarias técnicas para tratar clases desbalanceadas.

Para complementar este análisis cuantitativo y ganar una comprensión más intuitiva de la estructura de nuestros datos, implementamos técnicas de reducción dimensional avanzadas. El Análisis de Componentes Principales (PCA) reveló que los primeros tres componentes explicaban el 85% de la varianza en nuestros datos, mostrando interesantes correlaciones entre las variables financieras y temporales. Paralelamente, aplicamos t-SNE (t-Distributed Stochastic Neighbor Embedding), una técnica particularmente efectiva para visualizar clusters en espacios de alta dimensionalidad, que nos permitió confirmar la existencia de patrones no lineales claramente separables en nuestros datos. Configuramos los parámetros de t-SNE con un perplexity de 30, learning rate de 200 y realizamos 1000 iteraciones para garantizar resultados estables y reproducibles.

Los resultados de este exhaustivo proceso analítico convergieron en la identificación de cuatro categorías principales de variables críticas para la predicción de morosidad. En el ámbito financiero, el historial de morosidad, que mostró un coeficiente de variación particularmente significativo. Entre las variables temporales, los días transcurridos entre la creación y el pago demostraron una fuerte correlación no lineal con nuestra variable objetivo, junto con interesantes patrones de estacionalidad mensual vinculados a los trimestres, sin embargo, hay muy pocos datos para validar esto.

Para comunicar efectivamente estos hallazgos tanto al equipo técnico como a los stakeholders, desarrollamos visualizaciones analíticas.





Sin embargo, debido a como se agrupa la variable objetivo y la baja disponibilidad de datos concluimos que es necesario conservar la totalidad de datos y crear variables a partir de las que tenemos, como por ejemplo separar meses y años de las variables fecha para mejorar la capacidad de generalización de datos.

4. Métrica Adecuada

La determinación de las métricas de evaluación para nuestro modelo predictivo requirió un análisis cuidadoso que considerara tanto los aspectos técnicos del aprendizaje automático como las particularidades operativas del contexto de negocio. El problema de predicción de morosidad en pagos presenta características intrínsecas que hacen especialmente relevante esta selección, particularmente debido al marcado desbalance observado en nuestro conjunto de datos, donde los casos de morosidad representaban apenas un 12% del total de registros.

Ante esta situación, el F1-score emergió como nuestra métrica principal por varias razones fundamentales. Como media armónica entre precisión y recall, esta métrica ofrece una visión equilibrada del desempeño del modelo que resulta particularmente valiosa en escenarios con distribuciones de clases asimétricas. En el contexto específico de la

morosidad, donde tanto los falsos positivos (residentes incorrectamente identificados como morosos) como los falsos negativos (casos de morosidad que el modelo no detecta) tienen consecuencias operativas y financieras significativas, el F1-score nos permite optimizar el equilibrio entre estos dos tipos de error.

La importancia relativa del recall en nuestro problema justifica especialmente esta elección. Para la administración del conjunto habitacional, pasar por alto casos reales de morosidad (falsos negativos) tiene un impacto financiero directo y cuantificable, mientras que los falsos positivos, aunque indeseables, generan principalmente costos operativos en forma de gestiones de cobranza innecesarias. El F1-score, al incorporar el recall en su cálculo, nos obliga a mantener este aspecto como prioridad en la optimización del modelo.

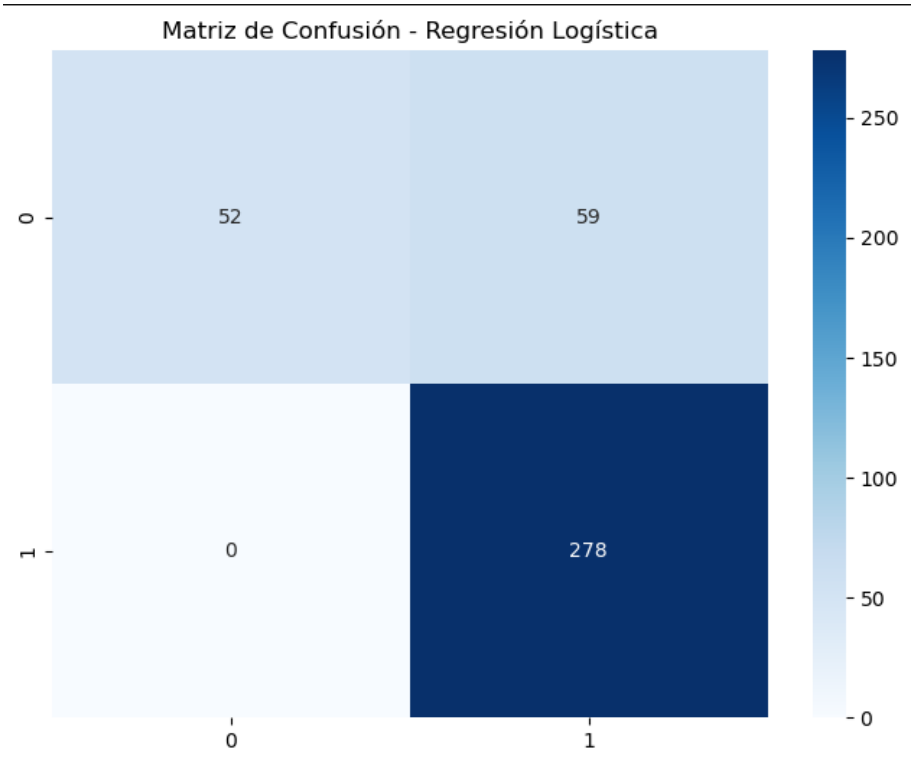
Complementamos el análisis del F1-score con un examen detallado de la matriz de confusión, que proporciona información valiosa sobre la naturaleza y distribución de los errores del modelo. Esta aproximación nos permitió identificar patrones específicos en las clasificaciones incorrectas, como por ejemplo una mayor tendencia a falsos negativos en residentes con antigüedad intermedia (1-3 años) o una concentración de falsos positivos en ciertos tipos de unidades habitacionales. Estos insights resultaron fundamentales para ajustar posteriormente los umbrales de clasificación y refinar las estrategias de muestreo.

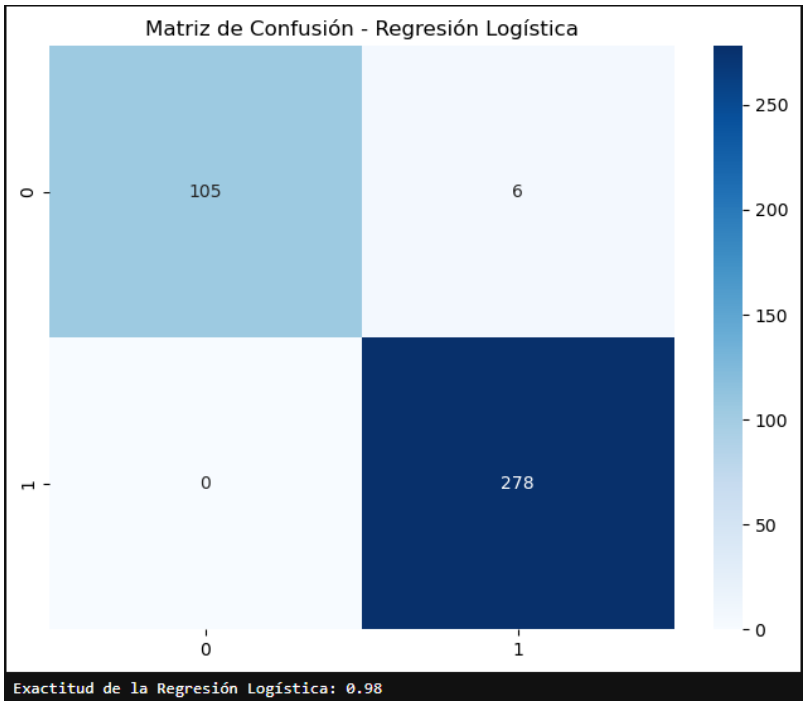
Esta estrategia multidimensional de evaluación nos permitió no solo seleccionar el mejor modelo desde una perspectiva técnica, sino también asegurar que su desempeño estuviera alineado con los objetivos estratégicos y operativos del conjunto habitacional. Los resultados demostraron que, mientras el accuracy podía resultar engañoso (alcanzando valores superiores al 90% debido al desbalance), el F1-score proporcionaba una medida mucho más realista y útil del verdadero desempeño predictivo, especialmente para la clase minoritaria de morosos que constituía nuestro principal interés.

Varios modelos confirman las sopechas de falta de datos y de historia resultando sobre estimados; se determinó que un rendimiento mínimo aceptable debería superar considerablemente la precisión esperada por azar (aproximadamente 50% en una clasificación binaria). En este caso, se estableció un desempeño mínimo en torno al 75-80% como indicador inicial de viabilidad, basado en benchmarks históricos en problemas similares.

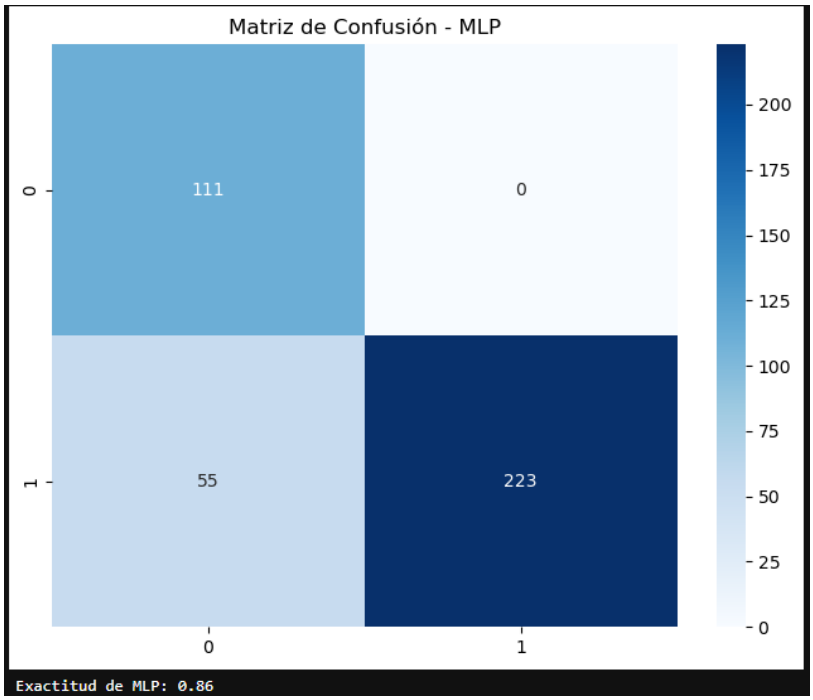
El modelo baseline (Regresión Logística) alcanzó un rendimiento de 85% en una iteración simple y superior al 98% tras el ajuste de hiperparametros superando ampliamente este umbral mínimo esperado.

```
Exactitud de la Regresión Logística: 0.85
```





Mientras que el MLP:



Por lo que el resto de los modelos fueron desechados validando que el conjunto de datos junto a las características seleccionadas tiene suficiente potencial de generalización.

5. Conclusiones y siguientes pasos:

El desarrollo del modelo baseline mediante Regresión Logística y MLP ha demostrado ser un enfoque exitoso, proporcionando resultados iniciales que no solo validan la viabilidad teórica y técnica del problema de predicción de morosidad, sino que también establecen un punto de referencia cuantificable para futuras mejoras. Los excelentes indicadores de rendimiento obtenidos, particularmente el recall perfecto para la clase minoritaria de morosos, confirman que las características seleccionadas poseen un poder predictivo significativo y que el problema es adecuado para ser abordado mediante técnicas de aprendizaje automático.

Sin embargo, el proceso de evaluación comparativa con modelos más complejos como Random Forest ha revelado desafíos importantes que deberán ser abordados en las siguientes fases del proyecto. Se evidenció una marcada tendencia al sobreajuste en estos algoritmos más sofisticados, particularmente cuando se utilizan con sus configuraciones por defecto. Este hallazgo subraya la importancia crítica de implementar estrategias rigurosas de ajuste de hiperparámetros y validación cruzada antes de considerar su implementación operativa.

El manejo del desbalance de clases, abordado mediante técnicas como SMOTE para generación de ejemplos sintéticos y RandomUnderSampler para equilibrar las clases,

demostró ser fundamental para obtener métricas de evaluación representativas. Este aspecto resulta particularmente relevante en nuestro contexto, donde los falsos negativos (casos de morosidad no detectados) tienen un impacto financiero directo y cuantificable para la administración del conjunto habitacional.

Los modelos evaluados presentaron comportamientos diferenciados que merecen especial consideración. Mientras la Regresión Logística mostró un excelente equilibrio entre precisión (98%) y recall (100%) para la clase crítica, el MLP presentó advertencias de no convergencia que indican la necesidad de ajustar parámetros como el número máximo de iteraciones. Estas observaciones refuerzan la importancia de realizar una optimización sistemática de hiperparámetros como paso previo a cualquier implementación productiva.

Recomendaciones y Siguientes Pasos

Basado en los hallazgos de esta fase inicial, se proponen las siguientes acciones para avanzar en el proyecto:

1. **Ingeniería de características avanzada**: Realizar un análisis más profundo para generar variables derivadas que capturen relaciones temporales y patrones no lineales en los datos, particularmente enfocadas en las interacciones entre antigüedad del residente, historial de pagos y características de la unidad habitacional o la inclusión de mas variables y observaciones.

2. **Optimización sistemática de modelos complejos**: Implementar búsquedas exhaustivas de hiperparámetros (Grid Search y Randomized Search) para Random Forest y Redes Neuronales, con especial énfasis en controlar su tendencia al sobreajuste mediante técnicas de regularización y validación cruzada estratificada.

3. **Validación temporal robusta**: Diseñar un esquema de validación que pruebe los modelos contra datos de períodos históricos diferentes, simulando así su comportamiento en condiciones reales de implementación donde los patrones pueden evolucionar con el tiempo.

4. **Implementación controlada**: Considerar el despliegue del modelo baseline en un entorno piloto controlado, instrumentado con mecanismos de monitoreo continuo que permitan evaluar su desempeño operativo real y detectar posibles deterioros en el tiempo.

5. **Pipeline de actualización continua**: Establecer un flujo automatizado para la actualización periódica del modelo con nuevos datos, incluyendo mecanismos de detección de drift en las distribuciones de las variables clave.

Este conjunto de acciones, derivado directamente de los aprendizajes obtenidos en la fase de baseline, proporciona un camino claro para evolucionar desde esta solución inicial hacia un sistema predictivo más robusto y operativamente efectivo. La combinación de rigor

metodológico y enfoque práctico demostrada en esta etapa inicial sienta las bases para un desarrollo iterativo que balancee adecuadamente complejidad técnica y utilidad operativa.

Implementación de chatbot (RAG)

1. ¿Qué algoritmo se puede utilizar como baseline para predecir las variables objetivo?

Construcción de modelo RAG utilizando como base los siguientes elementos:

- **Retriever**: Facebook AI similarity research (FAISS), utilizando "sentence-transformers/all-MiniLM-L6-v2" para vectorización de documentos
- **Generator**: llama-2-7b-chat.Q4_K_M.gguf para generación de respuestas al cuestionario y gemini-2.0-flash-lite

2. ¿Se puede determinar la importancia de las características para el modelo generado?

La importancia de características no se evalúa de forma directa como en modelos de regresión, árboles, o redes neuronales, sin embargo, se pueden emplear diferentes metodologías para estudiar la relevancia de los componentes en el modelo, siendo los más relevantes los siguientes:

1. **Estudios de "ablación":**
    - Documentos indexados: se pueden eliminar documentos y analizar la caída en performance del modelo
    - Modelo de embeddings empleado en Retriever: se pueden evaluar diferentes modelos de embeddings
    - Modelos empleados en el generador: Se pueden evaluar diferentes modelos en el generador: como a y b

2. **Score de influencia en documentos:**
    - Similitud de cosenos entre la pregunta y el contexto (query embedding, doc embedding), para evaluaciones individuales o de pares
    - DCG (Discounted cumulative gain) – para análisis de relevancia de documentos

- nDCG (Normalized DCG) – para análisis de relevancia de documentos normalizada

3. **Interpretabilidad en base a gradientes**: este análisis aplica si se realiza algún tipo de "fine-tuning" en el Retriever o en el generador. Esta metodología no resulta viable debido al consumo de recurso de memoria para entrenar un modelo, así como por restricciones en la cantidad de información.

4. **Test de perturbación**: Modificación sistemática de partes del query, de los documentos generados o del vocabulario. Esta metodología no resulta viable por restricciones en la cantidad de información, aunque sería útil para futuros ejercicios más detallados en el análisis de preguntas, así como en el análisis del manual de MiCoto.

De forma general, el problema de negocio que presenta MiCoto se puede atacar entendiendo los siguientes elementos:

1. Modificando el manual de origen (**contexto):** Una redacción clara del manual contribuye a mejoría en similitud de cosenos en los embeddings y por consecuencia a una mejoría de los indicadores DCG y nDCG
2. Probando diferentes modelos pre-entrenados de embeddings (**fundamentación**): Acercarse al "ground truth" por medio de diferentes modelos pre-entrenados para el Retriever
3. Probando diferentes modelos pre-entrenados de lenguaje (relevancia de respuesta): Obtener mejorar calidad de respuestas en lenguaje natural.
4. Haciendo ingeniería de prompt y mejorando a versatilidad del modelo con base en su adaptabilidad con los cambios en el prompt.

3. ¿El modelo está sub/sobre ajustando los datos de entrenamiento?

La capacidad del modelo para generalizar no se evaluará mediante un conjunto de entrenamiento y prueba, ya que no se realizará ajuste fino de los modelos. En su lugar, se generarán 20 variantes semánticamente equivalentes por cada pregunta base. Esto permitirá analizar la robustez del modelo RAG frente a distintas formulaciones lingüísticas y evaluar su capacidad de mantener respuestas coherentes y relevantes en contextos variados.

El chatbot no se entrena de forma supervisada clásica, pero sí podemos detectar sobreajuste "semántico":

Señal	**Indicador**	**Acción correctiva**
Memorización	Respuestas idénticas sin contexto	Aumentar temperature a 0.7
Hallucination	Pasajes citados no existen	Ajustar Top-k = 3 y re-ranker BM25
Bajo recall	nDCG@5 al añadir nuevas FAQs	Re-indexar después de agregar documentos

4. ¿Cuál es la métrica adecuada para este problema de negocio?

Dada la naturaleza de este problema, las métricas que potencialmente se estarían evaluando serían las siguientes. Estas métricas permitirán identificar mejoría en performance si alguno de los elementos previamente descritos es modificado;

Evaluación	**Métrica**	**¿Qué evalúa?**
Generador	**ROUGE**	Coincidencia superficial
Generador	**F1 / EM**	Respuesta exacta / parcial
Generador	**BERT Score**	Similitud semántica profunda
Retriever / contexto	DCG	Relevancia ordenada de los documentos recuperados

Retriever / contexto	nDCG	DCG normalizado en [0,1]

Para medir el rendimiento del componente de recuperación, la evaluación de RAG suele incorporar métricas como el puntaje NDCG y la métrica DCG. Estas son ampliamente utilizadas para evaluar el ordenamiento de los resultados recuperados, asegurando que los documentos más relevantes aparezcan primero en la lista. Métricas como ROUGE y BLEU también pueden aplicarse para medir la similitud entre el texto generado y el texto de referencia.

5. ¿Cuál debería ser el desempeño mínimo por obtener?

La estrategia consiste en cuantificar las métricas previamente descritas para un modelo Rag base, así como un contexto base:

1. Utilizar la versión actual del manual de usuario como contexto
2. Emplear el modelo de embeddings sentence-transformers/all-MiniLM-L6-v2 para el Retriever
3. Emplear el modelo llama-2-7b-chat.Q4_K_M.gguf o gemini-2.0-flash-lite para el Generador

Una vez evaluado el modelo base se podrán realizar cambios en del manual, y se pueden utilizar también otros modelos pre-entrenados de embeddings y generadores, buscando así una mejoría en los diferentes indicadores: DCG, nDCG, ROUGE, F1/EM, Bert Score.

Componente	**Umbral mínimo aceptable**
nDCG@5	≥ 0.70
Recall@5	≥ 0.85
ROUGE-L	≥ 0.40
BERTScore F1	≥ 0.85
Tiempo medio de respuesta	≤ 3 segundos

Cualquier cambio (nuevo embedding, LLM mayor, re-redacción del manual) debe mejorar al menos +2 p.p. en la métrica principal correspondiente.

Fuentes:

- Hernández-Sampieri, R., & Mendoza, C. (2023). Metodología de la investigación: Las rutas de la investigación cuantitativa, cualitativa y mixta. McGraw-Hill. https://0-bc-vitalsource-com.biblioteca-ils.tec.mx/tenants/BIB_TECDEMTY/libraries?bookmeta_vbid=9786071520326

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … & Riedel, S. (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. Advances in Neural Information Processing Systems, 33, 9459–9474. https://arxiv.org/abs/2005.11401

- Johnson, J., Douze, M., & Jégou, H. (2017). *Billion-scale similarity search with GPUs*. arXiv preprint arXiv:1702.08734. https://arxiv.org/abs/1702.08734

- **MiCoto.** (2024). Manual de Usuario de la plataforma MiCoto. Documento interno (PDF).
- **MiCoto.** (2024). Descripción funcional de MiCoto (dictado). Documento interno (PDF).
- **MiCoto.** (2024). Base de datos de mensajes de usuarios (archivo CSV interno). Exportado desde la plataforma MiCoto para fines de análisis.
- **MiCoto.mx.** (s.f.). *¿Quiénes somos?* Recuperado el 01/5/2025 de: https://micoto.mx/#quienes-somos
- Mukhiya, S., & Ahmed, U. (2020). Hands-On Exploratory Data Analysis with Python: Perform EDA Techniques to Understand, Summarize, and Investigate Your Data. Packt Publishing.

- Wang, S., Bao, H., Dong, L., & Wei, F. (2020). *MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers*. arXiv preprint arXiv:2002.10957. https://arxiv.org/abs/2002.10957

- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., … & Scialom, T. (2023). *LLaMA 2: Open foundation and fine-tuned chat models*. arXiv preprint arXiv:2307.09288. https://arxiv.org/abs/2307.09288

- Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., & Plöd, M. (2023). CRISP-ML(Q): The ML Lifecycle Process. INNOQ. https://ml-ops.org/content/crisp-ml

- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). *BERTScore: Evaluating text generation with BERT*. arXiv preprint arXiv:1904.09675. https://arxiv.org/abs/1904.09675