Chatbot de atención al cliente con análisis de sentimientos: comparación de modelos de aprendizaje automático

Paula Llanos López
Universidad EAFIT
Medellín, Colombia
pillanosl@eafit.edu.co

Samuel Rivero Urribarrí
Universidad EAFIT
Medellín, Colombia
seriverou@eafit.edu.co

Sara López Marín
Universidad EAFIT
Medellín, Colombia
slopezm13@eafit.edu.co

***Abstract*—Los chatbots de atención al cliente a menudo no reconocen las emociones de los usuarios, lo que genera experiencias frustrantes cuando estos se muestran molestos o confundidos. Este proyecto propone el desarrollo y la evaluación de un chatbot que detectará emociones en los mensajes de los clientes y adaptará sus respuestas en consecuencia. Compararemos tres modelos de aprendizaje automático (Regresión Logística, Máquina de Vectores de Soporte y Naive Bayes) que serán entrenados con conversaciones de atención al cliente etiquetadas. El chatbot clasificará los mensajes de los usuarios en tres categorías emocionales: positivas, neutrales y negativas. Cuando se detecte un sentimiento negativo, el sistema proporcionará respuestas empáticas o sugerirá escalar el caso a soporte humano. Implementaremos una interfaz web con Python y Streamlit, que permitirá la interacción en tiempo real y la visualización de sentimientos. Esperamos que la Regresión Logística logre un rendimiento cercano al 85% de precisión, con un tiempo de respuesta promedio inferior a 2 segundos. Este trabajo demostrará que se pueden crear chatbots eficaces que reconocen las emociones mediante técnicas sencillas de aprendizaje automático y recursos públicos, lo que facilita su implementación a pequeña escala.**

***Index Terms*—Chatbot, análisis de sentimientos, atención al cliente, aprendizaje automático, detección de emociones, procesamiento del lenguaje natural.**

I. Introducción

Los chatbots de servicio al cliente están presentes en casi todos los sitios web y aplicaciones, pero la mayoría tienen una limitación crítica: no entienden las emociones del usuario. Un cliente frustrado recibe las mismas respuestas genéricas que uno satisfecho, lo que puede empeorar la situación y dañar la relación con el cliente.

Considérese este escenario: un cliente está molesto porque su pedido llegó tarde. Escribe al chatbot: "¡Ya pasaron 3 días y mi paquete NO ha llegado! ¿Qué está pasando?" Un chatbot tradicional responde: "Por favor proporciona tu número de orden." Esta respuesta, aunque técnicamente correcta, ignora completamente la frustración del cliente, haciéndolo sentir ignorado.

Un chatbot con análisis de sentimientos podría detectar la emoción negativa y responder: "Lamento mucho los problemas con tu entrega. Entiendo tu frustración. Déjame ayudarte inmediatamente con esto. ¿Podrías proporcionarme tu número de orden?" Esta pequeña diferencia puede cambiar completamente la experiencia del cliente.

A. Objetivos del Proyecto

Este proyecto abordará este problema desarrollando un chatbot que:

- **Detectará emociones** en los mensajes de los clientes (positivo, neutral, negativo).
- **Adaptará sus respuestas** según la emoción detectada.
- **Recomendará escalamiento** a un agente humano cuando detecte frustración alta.
- **Funcionará en tiempo real** con interfaz web fácil de usar.

B. Importancia del Proyecto

Para las empresas:

- Clientes más satisfechos = más retención y ventas.
- Detectar problemas antes de que escalen.
- Reducir carga de trabajo de agentes humanos.
- Datos sobre cómo se sienten los clientes.

Para nosotros como estudiantes:

- Aprender técnicas reales de Machine Learning aplicadas.
- Desarrollar un prototipo funcional y evaluable.
- Entender procesamiento de lenguaje natural (NLP).
- Generar un proyecto para portafolio profesional.

C. Diferenciación

Nuestro chatbot NO intentará:

- No aborda preguntas abiertas de alta complejidad fuera del dominio.
- No busca competir con modelos fundacionales o sistemas avanzados.

Nuestro chatbot SÍ:

- Detección del estado emocional del usuario (positivo, neutral, negativo).
- Responderá apropiadamente según la emoción.
- Proporcionará respuestas útiles a preguntas frecuentes.
- Criterios para escalar a un agente humano cuando corresponda.

D. Alcance

ALCANCE:

- Chatbot funcional con análisis de sentimientos.
- Comparación de tres modelos de ML.
- Conjunto de entrenamiento de $\sim 14{,}000$ ejemplos.
- Interfaz web simple pero funcional.
- Respuestas a 10-15 preguntas frecuentes.
- Sistema de escalamiento básico a agente humano.

II. Revisión de Literatura y Estado del Arte

A. Tipos de Chatbots

Un chatbot es un programa que simula una conversación con humanos. Los hay de varios tipos:

1. Chatbots basados en reglas: Funcionan como un árbol de decisiones. Si el usuario dice "X", responde "Y". Ejemplo: "Si el usuario dice 'hola', responder 'Hola, ¿en qué puedo ayudarte?"' Son muy predecibles y fáciles de programar, pero no entienden variaciones y son rígidos.

2. Chatbots con Machine Learning (nuestro enfoque): Aprenden patrones de ejemplos y pueden generalizar a mensajes no vistos antes. Ejemplo: Aprenden que "estoy furioso", "estoy molesto", "qué rabia" son emociones negativas. Son más flexibles y manejan variaciones, pero necesitan datos de entrenamiento.

3. Chatbots con IA avanzada: Usan redes neuronales gigantes y pueden conversar sobre casi cualquier tema, pero requieren recursos masivos. No es nuestro objetivo (demasiado complejo).

B. Análisis de Sentimientos

El análisis de sentimientos es enseñarle a una computadora a detectar emociones en texto. Es como cuando tú lees un mensaje y sabes si la persona está feliz, enojada o neutral, pero automatizado.

Ejemplos:

- "¡Excelente servicio, muy rápido!" $\rightarrow$ **POSITIVO**
- "¿Cuál es el horario de atención?" $\rightarrow$ **NEUTRAL**
- "Pésimo servicio, nunca vuelvo" $\rightarrow$ **NEGATIVO**

Proceso:

1) **E**ntrenamiento SVM
2) **E**ntrenamiento Naive Bayes
3) **L**ógica del Chatbot
4) **I**nterfaz Streamlit

C. Plan de Experimentos

TABLE I
Plan de Experimentos

Exp.	Qué probaremos	Métrica
E1	Logistic Regression default	Accuracy, tiempo
E2	SVM kernel linear	Accuracy vs E1
E3	Naive Bayes	Velocidad
E4	Dataset 14,000 ejemplos	¿Afecta accuracy?
E5	Tamaños vocabulario TF-IDF	Features óptimas

III. Resultados Esperados

A. Predicciones de Rendimiento

TABLE II
Resultados Esperados por Modelo

Modelo	Acc.	F1	Train	Pred.
Log. Reg.	85-88%	0.84-0.87	30-60s	$<$1s
SVM	87-90%	0.86-0.89	2-5min	1-2s
Naive Bayes	82-85%	0.80-0.83	10-20s	$<$0.5s

B. Análisis de Trade-offs

Escenario 1: Máxima precisión

- Usaremos: SVM.
- Razón: Mejor accuracy aunque sea más lento.
- Ideal para: Análisis offline, reportes.

Escenario 2: Velocidad máxima

- Usaremos: Naive Bayes.
- Razón: Mucho más rápido.
- Ideal para: Aplicaciones de alto volumen.

Escenario 3: Mejor balance

- Usaremos: Logistic Regression.
- Razón: 85-88% accuracy con velocidad rápida.
- Ideal para: Nuestro chatbot.

C. Entregables Finales

1) **Código Fuente:**
   - Repositorio GitHub organizado (Notebooks).
   - Interfaz sencilla.
   - Archivos de modelos entrenados.
   - README con instrucciones de instalación.
2) **Documentación:**
   - Este paper IEEE completo.
   - Documentación del código.
   - Guía de usuario para la interfaz.
3) **Resultados:**
   - Tabla comparativa de modelos.
   - Matrices de confusión.
   - Gráficas de rendimiento.
   - Análisis de errores.

IV. Consideraciones Técnicas

A. Problemas Comunes y Soluciones

1) Data Leakage: **Problema:** Información del conjunto de prueba filtrándose al entrenamiento.

Solución: Separaremos train/test ANTES de cualquier procesamiento. Nunca tocaremos el conjunto de prueba hasta evaluación final.

2) Overfitting: **Problema:** Modelo memoriza datos de entrenamiento pero falla con datos nuevos.

Señal: 95% accuracy en entrenamiento pero 70% en prueba.

Solución:

- Validación cruzada.
- Regularización (parámetro C en Logistic Regression y SVM).
- Más datos de entrenamiento.
- Reducir complejidad del modelo.

3) Desbalance de Clases: **Problema:** Mucho más ejemplos de una clase que otra.

¿Por qué es problema? Modelo aprende a siempre predecir la clase mayoritaria.

Solución:

- Balancearemos el dataset (igual cantidad por clase).
- Usaremos class_weight='balanced' en modelos.
- Consideraremos técnicas de oversampling/undersampling.

B. Optimización de Hiperparámetros

Logistic Regression:

- C: Inverso de la fuerza de regularización.
- max_iter: Número máximo de iteraciones.
- solver: Algoritmo de optimización.

SVM:

- C: Parámetro de regularización.
- kernel: Tipo de kernel (linear, rbf, poly).
- gamma: Coeficiente del kernel.

Naive Bayes:

- alpha: Parámetro de suavizado.

C. Validación Cruzada

Implementaremos validación cruzada k-fold (k=5) para obtener estimaciones más robustas del rendimiento.

V. Evaluación de Calidad

A. Métricas de Evaluación Detalladas

Para cada modelo calcularemos:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Donde:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

B. Análisis de Errores

Analizaremos los casos donde el modelo falle:

1) **Sarcasmo:** "Oh, genial, otro retraso" (negativo, pero parece positivo).
2) **Contexto:** "Nada mal" (positivo, pero contiene palabra negativa).
3) **Neutralidad:** Mensajes informativos mal clasificados.
4) **Lenguaje informal:** Abreviaciones, emojis, slang.

C. Pruebas de Usuario

Realizaremos pruebas con 20 usuarios reales:

- 10 mensajes positivos.
- 5 mensajes neutrales.
- 10 mensajes negativos.
- Mediremos satisfacción del usuario (escala 1-5).
- Recopilaremos feedback cualitativo.

VI. Impacto y Aplicaciones

A. Aplicaciones Prácticas

Empresa:

- Detectar clientes insatisfechos antes de que abandonen.
- Priorizar casos urgentes.
- Mejorar experiencia de compra.

Servicios Financieros:

- Identificar clientes frustrados con transacciones.
- Prevenir quejas escaladas.
- Mejorar retención de clientes.

Telecomunicaciones:

- Detectar problemas de servicio temprano.
- Reducir la pérdida de clientes.
- Optimizar soporte técnico.

B. Beneficios Esperados

Para las empresas:

- Potencial reducción en escalamientos innecesarios.
- Mejora en el nivel de satisfacción del cliente.
- Posible ahorro en costos de soporte.
- Identificación de problemas en tiempo real.

Para los clientes:

- Respuestas más empáticas.
- Resolución más rápida de problemas.
- Mejor experiencia general.
- Sensación de ser comprendidos.

VII. Datos y análisis preliminar

A. Conjunto y procedencia.

Se utiliza el conjunto público *Twitter US Airline Sentiment* (Kaggle), con $\approx$ 14,640 tuits en inglés sobre aerolíneas de EE. UU. El dominio (quejas, consultas y agradecimientos) es afín a nuestro chatbot de soporte.

B. Variables

- $\mathbf{y}$: airline_sentiment $\in \{\text{negative}, \text{neutral}, \text{positive}\}$.
- $\mathbf{X}$: text (contenido del tuit). Para evitar fugas, se excluyen de X columnas asociadas a la etiqueta (airline_sentiment_confidence, negativereason, negativereason_confidence) y metadatos no textuales.

C. Tamaño y muestreo

Partición **estratificada** 80 % *train* (con **CV** k=5 estratificada) y 20 % *test* (*hold-out*); `random_state` fijo. Si se requiere, se usa una **muestra estratificada**.

D. Análisis exploratorio de datos inicial

- **Distribución de clases:** desbalance hacia *negative* ($\approx$ 62–63%); *neutral* $\approx 21\%$, *positive* $\approx 16\%$.
- **Texto:** se revisan longitud, URLs, menciones, #hashtags y emojis. Aplicamos una limpieza ligera.

E. Prevención de fugas

Split **antes** de vectorizar; TF–IDF ajustado solo con *train*; exclusión de columnas reveladoras; evitar (cuando sea identificable) que tuits del mismo hilo/usuario caigan en particiones distintas.

F. Métrica de evaluación

Por el desbalance y la necesidad de rendimiento homogéneo, la métrica principal es **F1 macro**, también **accuracy** y **matrices de confusión**.

VIII. Limitaciones y Trabajo Futuro

A. Limitaciones Declaradas

1) **Idioma:** Se trabajará inicialmente en inglés por disponibilidad de datos públicos.
2) **Clasificación simple:** Solo 3 categorías de emoción (positivo, neutral, negativo).
3) **Preguntas limitadas:** Solo responderá a preguntas comunes que programemos.
4) **Modelos simples:** Usaremos ML clásico.
5) **Contexto limitado:** No mantendrá historial de conversación.
6) **Sarcasmo:** Tendrá dificultad para detectar sarcasmo.
7) **FAQs limitados:** Solo 10-15 preguntas predefinidas.
8) **Dataset moderado:** $\sim$ 14,000 ejemplos es suficiente para prototipo, no para producción industrial.

B. Mejoras Futuras

- Expandir a más idiomas (español).
- Aumentar dataset a mayor cantidad de ejemplos.
- Agregar más categorías emocionales (enojo, alegría, miedo, sorpresa).
- Implementar detección de sarcasmo.
- Ampliar FAQs a 50+ preguntas.
- Mantener contexto de conversación

C. Investigación Adicional

Áreas interesantes para investigar en el futuro:

1) **Transfer Learning:** ¿Puede un modelo entrenado en un dominio (aerolinea) funcionar en otro (e-commerce)?
2) **Explicabilidad:** ¿Cómo explicar las predicciones del modelo a usuarios no técnicos?
3) **Privacidad:** ¿Cómo manejar datos sensibles en conversaciones?
4) **Sesgo:** ¿El modelo tiene sesgos hacia ciertos grupos demográficos?
5) **Adversarial attacks:** ¿Pueden usuarios engañar al sistema intencionalmente?

IX. Conclusiones Esperadas

Este proyecto busca demostrar que es posible construir un chatbot efectivo con análisis de sentimientos usando técnicas de machine learning clásicas y recursos públicos limitados.

1) **Viabilidad técnica:** Esperamos que los tres modelos evaluados (Logistic Regression, SVM, Naive Bayes) logren accuracy superior al 80%, suficiente para aplicaciones prácticas.
2) **Balance óptimo:** Anticipamos que Logistic Regression ofrecerá el mejor balance entre precisión (85-88% accuracy) y velocidad ($<$1 segundo), haciéndolo ideal para chatbots en tiempo real.
3) **Accesibilidad:** Con $\sim$ 14,000 ejemplos de entrenamiento, hardware modesto, y herramientas gratuitas, demostraremos que estudiantes y pequeñas empresas pueden implementar esta tecnología.
4) **Impacto real:** La detección de emociones permitirá respuestas más empáticas, reduciendo la frustración del cliente y mejorando la satisfacción.
5) **Escalabilidad:** El sistema estará diseñado para crecer: más idiomas, más datos, más emociones, más canales.

A. Contribuciones Esperadas

Este trabajo contribuirá:

- Comparación justa de 3 modelos con mismo dataset y métricas.
- Código abierto completo y bien documentado.
- Aplicación práctica funcional (no solo teoría).
- Documentación detallada para estudiantes.

B. Reflexión Final

La emoción en el servicio al cliente importa. Un chatbot que responde correctamente pero sin empatía puede ser tan frustrante como uno que se equivoca. Este proyecto demostrará que agregar inteligencia emocional a chatbots no requiere tecnología compleja ni recursos masivos, es accesible y tendrá impacto real en la experiencia del usuario.

El futuro de los chatbots no es solo responder preguntas correctamente, sino hacerlo de manera que los usuarios se sientan comprendidos y valorados. Este trabajo será un pequeño paso en esa dirección.

REFERENCES

[1] Liu, B. (2012). *Sentiment analysis and opinion mining*. Synthesis lectures on human language technologies, 5(1), 1-167.

[2] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1-135.

[3] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12), 2009.

[4] Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-based systems*, 89, 14-46.

[5] Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), 82-89.

[6] Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational intelligence*, 29(3), 436-465.

[7] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12, 2825-2830.

[8] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

[9] Jurafsky, D., & Martin, J. H. (2020). *Speech and language processing* (3rd ed. draft). Pearson.

[10] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1253.

[11] Socher, R., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of EMNLP*, 1631-1642.

[12] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of EMNLP*, 1746-1751.

[13] Devlin, J., et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171-4186.

[14] Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A new chatbot for customer service on social media. *Proceedings of CHI*, 3506-3510.

[15] Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *interactions*, 24(4), 38-42.