Instituto Politécnico Nacional

Escuela Superior de Cómputo

Práctica 5. Análisis de Sentimientos

R E P O R T E

Materia:

Tecnologías de Lenguaje Natural

presenta:

Luis Fernando Rodríguez-Domínguez

Profesor:

Ituriel Enrique Flores Estrada

Índice general

Capítulo 1

Introducción

El presente reporte documenta la realización de la Práctica 5 correspondiente a la asignatura de Tecnologías de Lenguaje Natural. El objetivo principal de esta práctica es implementar y evaluar un sistema para realizar análisis de sentimientos, también conocido como análisis de polaridad o de opinión. Para ello, se aplicarán tres enfoques metodológicos distintos: análisis basado en diccionarios (lexicons), aprendizaje de máquina clásico y redes neuronales profundas con capas de embeddings.

Para llevar a cabo este estudio comparativo, se utilizará el conjunto de datos público "Amazon Fine Food Reviews" [1], obtenido a través de la plataforma Kaggle [2]. Este dataset, que contiene miles de reseñas de productos alimenticios, proporciona un corpus rico y desafiante para la clasificación de sentimientos.

Antes de la fase de modelado, se realizará un exhaustivo proceso de preprocesamiento de datos. Este comenzará con un Análisis Exploratorio de Datos (EDA) para comprender la estructura del dataset. Posteriormente, se aplicarán técnicas de limpieza y transformación, como la conversión de las calificaciones numéricas (1 a 5 estrellas) en etiquetas de sentimiento categóricas (Negativo, Neutral y Positivo) y el balanceo de clases mediante submuestreo para mitigar el sesgo en el entrenamiento de los modelos. Adicionalmente, se aplicarán técnicas de normalización de texto, justificando su uso para cada uno de los enfoques a evaluar.

Se implementarán y compararán los siguientes enfoques y técnicas:

- **Análisis Basado en Diccionarios:** Se asignará una polaridad a cada reseña sin necesidad de entrenamiento, utilizando tres léxicos predefinidos: **Opinion Lexicon** y **SentiWordNet**, provistos por la biblioteca NLTK [3], y el léxico **Harvard IV-4**, accedido a través de la librería `pysentiment2`.

- **Análisis con Aprendizaje de Máquina Clásico:** Se utilizará la representación vectorial **TF-IDF** para convertir el texto en características numéricas. Sobre esta representación, se entrenarán y evaluarán tres algoritmos supervisados de la biblioteca Scikit-learn [4]: **Regresión Logística (LR)**, **Árboles de Decisión (DT)** y **Máquinas de Soporte Vectorial (SVM)**.

- **Análisis con Redes Neuronales:** Se implementará una arquitectura de red neuronal para la clasificación, explorando dos estrategias para la capa de embeddings

utilizando el framework TensorFlow/Keras [5]:

1. Una capa de embeddings pre-entrenada con vectores de **GloVe** [6].
2. Una capa de embeddings que aprende sus propias representaciones vectoriales desde cero a partir del corpus de reseñas.

Finalmente, se realizará un análisis comparativo del rendimiento de los modelos de aprendizaje supervisado, utilizando métricas como la exactitud (accuracy) y la validación cruzada (k-fold cross-validation), para derivar conclusiones sobre la efectividad, ventajas y desventajas de cada enfoque en la tarea de análisis de sentimientos.

Capítulo 2

Desarrollo

En este capítulo se detalla la implementación y ejecución de los diferentes módulos del programa desarrollado para el análisis de sentimientos. Se presentarán las metodologías empleadas y las evidencias de su funcionamiento, abarcando la adquisición de datos, el preprocesamiento, y los tres enfoques de clasificación: basado en diccionarios, con aprendizaje de máquina clásico y mediante redes neuronales.

2.1. Adquisición y Análisis Exploratorio de Datos (EDA)

El primer paso fue la adquisición del conjunto de datos "Amazon Fine Food Reviews" [1] desde la plataforma Kaggle, utilizando la librería `kagglehub`. Una vez descargado, se cargó en un DataFrame de pandas para su análisis. La Figura 2.1 (correspondiente a la celda de ejecución [3] del notebook) muestra la confirmación de la carga y una inspección inicial, incluyendo las dimensiones del dataset, los nombres de las columnas y una muestra de las primeras filas. El EDA reveló un dataset con 568,454 reseñas y 10 columnas, de las cuales `Score` y `Text` son las más relevantes para esta práctica.

```
--- Inspección Inicial del Dataset ---
Dimensiones del dataset: 568454 filas y 10 columnas.

Nombres de las columnas: ['Id', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator', 'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text']

--- Primeras 3 filas del dataset: ---
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...

```
--- Caracterización de las Dimensiones ---
Columna: 'Id', Tipo de Dato: int64, Valores Únicos: 568454
Columna: 'ProductId', Tipo de Dato: object, Valores Únicos: 74258
Columna: 'UserId', Tipo de Dato: object, Valores Únicos: 256059
Columna: 'ProfileName', Tipo de Dato: object, Valores Únicos: 218415
Columna: 'HelpfulnessNumerator', Tipo de Dato: int64, Valores Únicos: 231
Columna: 'HelpfulnessDenominator', Tipo de Dato: int64, Valores Únicos: 234
Columna: 'Score', Tipo de Dato: int64, Valores Únicos: 5
Columna: 'Time', Tipo de Dato: int64, Valores Únicos: 3168
Columna: 'Summary', Tipo de Dato: object, Valores Únicos: 295742
Columna: 'Text', Tipo de Dato: object, Valores Únicos: 393579
```

Figura 2.1: Confirmación de la carga e inspección inicial del dataset.

Un análisis más detallado de las columnas, también mostrado en la Figura 2.1, propor-

cionó información sobre el tipo de dato y la cantidad de valores únicos por cada dimensión, cumpliendo con el objetivo de caracterizar el conjunto de datos.

2.2. Preprocesamiento de Datos

Con base en el EDA, se procedió a preprocesar los datos. Se justificó la necesidad de utilizar únicamente las columnas `Score` y `Text`, descartando las demás por no ser relevantes para la clasificación de sentimientos. Se eliminaron las filas con valores nulos en estas dos columnas.

A continuación, se convirtió la calificación numérica (`Score`) en una etiqueta de sentimiento categórica (Negativo, Neutral, Positivo) según las reglas especificadas. Al validar la distribución de estas nuevas clases, se observó un fuerte desbalance, con una mayoría de reseñas positivas (443,777) en comparación con las negativas (82,037) y neutrales (42,640). Para evitar que los modelos de aprendizaje se sesgaran hacia la clase mayoritaria, se aplicó una técnica de **submuestreo (undersampling)**, reduciendo el número de muestras de cada clase al tamaño de la clase minoritaria (42,640). La Figura 2.2 (salida de la celda [4]) muestra la distribución de clases antes y después de este proceso, resultando en un dataset balanceado de 127,920 reseñas en total.

```
Seleccionando columnas 'Score' y 'Text' y eliminando filas con valores nulos...
Dimensiones después de seleccionar y limpiar nulos: (568454, 2)

Creando la columna 'Sentimiento' a partir de 'Score'...

--- Distribución de Clases Original (después de crear sentimiento) ---
Sentimiento
Positivo    443777
Negativo     82037
Neutral      42640
Name: count, dtype: int64

Se observa un fuerte desbalance de clases, con una mayoría de reseñas positivas.

Balanceando clases mediante submuestreo...

--- Distribución de Clases Después del Balanceo ---
Cada clase tendrá 42640 muestras.
Sentimiento
Negativo    42640
Neutral     42640
Positivo    42640
Name: count, dtype: int64

Dimensiones del DataFrame balanceado: (127920, 3)
```

Figura 2.2: Distribución de las clases de sentimiento antes y después del balanceo.

2.3. Normalización de Texto

Se definió una estrategia de normalización de texto diferenciada. Para los modelos de Machine Learning y Redes Neuronales, se aplicó una limpieza agresiva para reducir el ruido, que incluyó conversión a minúsculas, y eliminación de URLs, etiquetas HTML, números, puntuación y stopwords. Esta función se encapsuló en `limpiar_texto_agresivo`.

En contraste, para los métodos basados en diccionarios, se decidió utilizar el texto original para no eliminar palabras contextuales clave como las de negación (ej. "not").

La Figura 2.3 (celda [6]) muestra una comparación entre un texto original y su versión normalizada, evidenciando el resultado del proceso de limpieza.

```
Ejemplo de texto post-normalización:
Texto original: 'This review is NOT good at all! It cost $10.00. Check out my blog at http://my-reviews.com 😊'
Texto normalizado (limpieza agresiva): 'review good cost check blog'
```

Figura 2.3: Ejemplo de una reseña original y su versión tras la normalización agresiva.

2.4. Análisis con Diccionarios (Lexicons)

Este enfoque no supervisado asignó una polaridad a cada reseña basándose en léxicos preexistentes. Se evaluó el rendimiento de cada léxico comparando su clasificación con la etiqueta de sentimiento real derivada del ‘Score‘. La Figura 2.5 (celda [7]) presenta los reportes de clasificación y las matrices de confusión para cada uno de los tres diccionarios utilizados:

- **Opinion Lexicon:** Mostró una exactitud del 46
- **SentiWordNet:** Alcanzó una exactitud del 43
- **Harvard IV-4:** Su rendimiento fue el más bajo, con una exactitud del 35

Estos resultados establecen un baseline y confirman las limitaciones de los métodos basados en reglas que no consideran el contexto.

```
--- Evaluación de Opinion Lexicon ---
Reporte de Clasificación:
              precision    recall  f1-score   support

    Negativo       0.64      0.37      0.47     42640
     Neutral       0.34      0.13      0.18     42640
    Positivo       0.43      0.87      0.57     42640

    accuracy                           0.46    127920
   macro avg       0.47      0.46      0.41    127920
weighted avg       0.47      0.46      0.41    127920

Accuracy: 0.4554799874921826
Matriz de Confusión:
[[15678  7493 19469]
 [ 6705  5381 30554]
 [ 2254  3180 37206]]

--- Evaluación de SentiWordNet ---
Reporte de Clasificación:
              precision    recall  f1-score   support

    Negativo       0.50      0.42      0.46     42640
     Neutral       0.34      0.03      0.06     42640
    Positivo       0.41      0.84      0.55     42640

    accuracy                           0.43    127920
   macro avg       0.42      0.43      0.36    127920
weighted avg       0.42      0.43      0.36    127920
```

Figura 2.4: Reportes de clasificación y matrices de confusión para los tres léxicos.

```
Accuracy: 0.43187929956222637
Matriz de Confusión:
[[18086  1641 22913]
 [12202  1375 29063]
 [ 5799  1056 35785]]

--- Evaluación de Harvard IV-4 ---
Reporte de Clasificación:
              precision    recall  f1-score   support

    Negativo       0.40      0.18      0.25     42640
     Neutral       0.32      0.16      0.21     42640
    Positivo       0.35      0.71      0.47     42640

    accuracy                           0.35    127920
   macro avg       0.36      0.35      0.31    127920
weighted avg       0.36      0.35      0.31    127920

Accuracy: 0.35136022514071297
Matriz de Confusión:
[[ 7630  7564 27446]
 [ 6352  6883 29405]
 [ 5002  7205 30433]]
```

Figura 2.5: Reportes de clasificación y matrices de confusión para los tres léxicos.

2.5. Análisis con Aprendizaje de Máquina Clásico

Para este enfoque supervisado, se utilizó el texto normalizado y se convirtió en una representación numérica mediante **TF-IDF**, limitando el vocabulario a las 5,000 características más importantes. El dataset se dividió en 80 % para entrenamiento y 20 % para prueba. Se evaluaron tres algoritmos: Regresión Logística, Árbol de Decisión y Linear SVM.

Para cada modelo, se realizó una validación cruzada de 5 pliegues sobre el conjunto de entrenamiento para obtener una estimación robusta de su rendimiento. Posteriormente, el modelo se entrenó con todos los datos de entrenamiento y se evaluó en el conjunto de prueba. La Figura 2.8 (salida de la celda [8]) muestra los resultados detallados, incluyendo

la exactitud media en validación cruzada y el reporte de clasificación final. La **Regresión Logística** y el **Linear SVM** mostraron un rendimiento muy sólido y similar (exactitud de 72 %), mientras que el **Árbol de Decisión** fue inferior (66 %).

```
--- Evaluando Modelo: Regresión Logística ---
Realizando validación cruzada (5-fold)...
  Exactitud (Accuracy) Media en CV: 0.7192 (+/- 0.0027)
Entrenando en el conjunto de entrenamiento completo...

Evaluando en el conjunto de prueba...
  Exactitud (Accuracy) en Prueba: 0.7215
  Reporte de Clasificación en Prueba:
              precision    recall  f1-score   support

    Negativo       0.73      0.73      0.73      8528
     Neutral       0.65      0.65      0.65      8528
    Positivo       0.78      0.79      0.79      8528

    accuracy                           0.72     25584
   macro avg       0.72      0.72      0.72     25584
weighted avg       0.72      0.72      0.72     25584
```

Figura 2.6: Resultados de la validación cruzada y evaluación en prueba para los modelos de Machine Learning clásico.

```
--- Evaluando Modelo: Árbol de Decisión ---
Realizando validación cruzada (5-fold)...
  Exactitud (Accuracy) Media en CV: 0.6431 (+/- 0.0006)
Entrenando en el conjunto de entrenamiento completo...

Evaluando en el conjunto de prueba...
  Exactitud (Accuracy) en Prueba: 0.6588
  Reporte de Clasificación en Prueba:
              precision    recall  f1-score   support

    Negativo       0.67      0.68      0.67      8528
     Neutral       0.63      0.64      0.63      8528
    Positivo       0.68      0.66      0.67      8528

    accuracy                           0.66     25584
   macro avg       0.66      0.66      0.66     25584
weighted avg       0.66      0.66      0.66     25584
```

Figura 2.7: Resultados de la validación cruzada y evaluación en prueba para los modelos de Machine Learning clásico.

```
--- Evaluando Modelo: Linear SVM ---
Realizando validación cruzada (5-fold)...
  Exactitud (Accuracy) Media en CV: 0.7133 (+/- 0.0038)
Entrenando en el conjunto de entrenamiento completo...

Evaluando en el conjunto de prueba...
  Exactitud (Accuracy) en Prueba: 0.7179
  Reporte de Clasificación en Prueba:
              precision    recall  f1-score   support

    Negativo       0.72      0.73      0.73      8528
     Neutral       0.66      0.63      0.64      8528
    Positivo       0.77      0.80      0.78      8528

    accuracy                           0.72     25584
   macro avg       0.72      0.72      0.72     25584
weighted avg       0.72      0.72      0.72     25584


Guardando modelos de ML entrenados en 'models_ml.pkl'...
[SUCCESS] Modelos guardados.
```

Figura 2.8: Resultados de la validación cruzada y evaluación en prueba para los modelos de Machine Learning clásico.

2.6. Análisis con Redes Neuronales

Finalmente, se exploró un enfoque de Deep Learning utilizando una red neuronal simple con dos estrategias de embeddings. Se aplicó validación cruzada y ‘EarlyStopping‘ para asegurar la robustez y evitar el sobreajuste.

2.6.1. Embeddings Pre-entrenados (GloVe)

Se utilizó una capa de Embedding inicializada con vectores GloVe de 200 dimensiones, manteniendo estos pesos congelados durante el entrenamiento. Como se observa en la Figura 2.11 (salida de la celda [13]), este modelo alcanzó una **exactitud en prueba del 63.8 %**. Las curvas de entrenamiento (Figura 2.9) muestran una brecha entre la exactitud de entrenamiento y la de validación, sugiriendo que el conocimiento general de GloVe no se transfirió de manera óptima a este dominio específico.

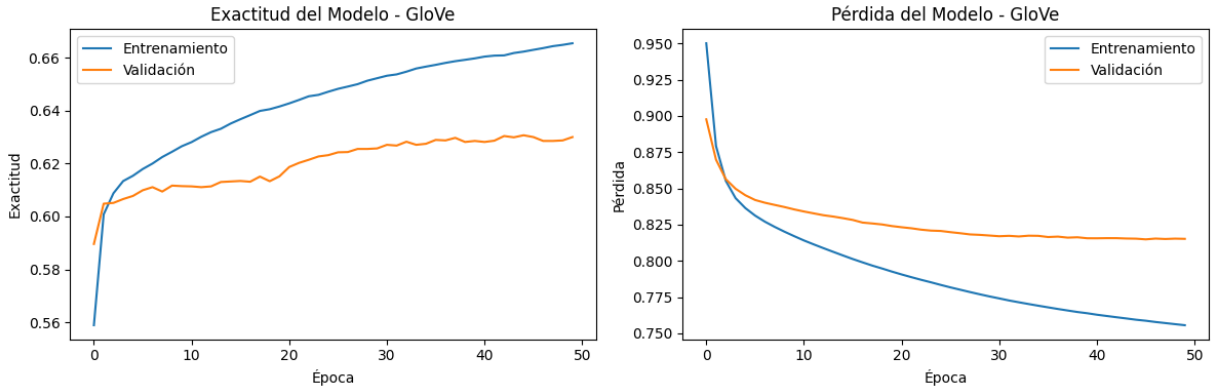

Figura 2.9: Curvas de entrenamiento y validación para el modelo con embeddings GloVe.

2.6.2. Embeddings Aprendidos

En este caso, la capa de Embedding se entrenó desde cero junto con el resto de la red. Este enfoque permitió que el modelo creara representaciones vectoriales adaptadas al vocabulario de las reseñas. El modelo obtuvo el mejor rendimiento de este apartado, con una **exactitud en prueba del 72.4 %**. Las curvas de entrenamiento (Figura 2.10) muestran una convergencia más estable y un mejor ajuste a los datos de validación en comparación con el modelo GloVe.

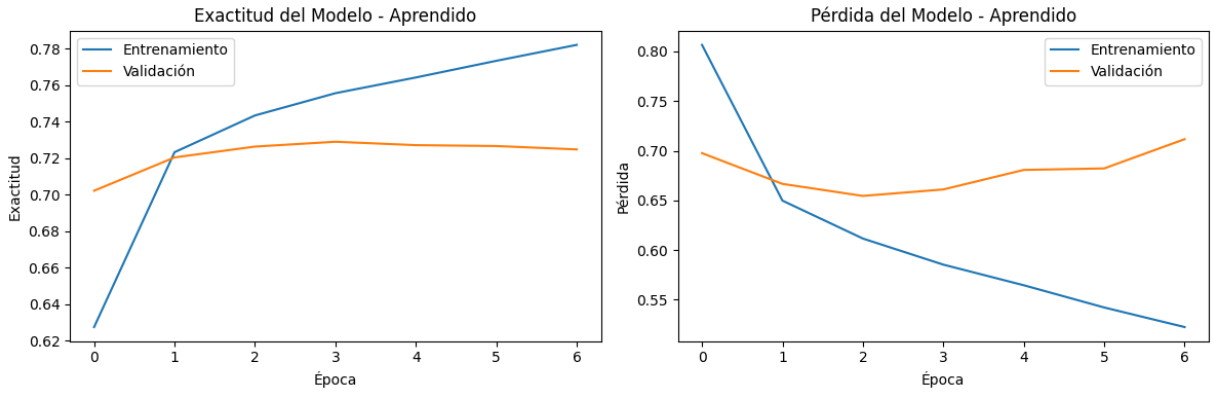

Figura 2.10: Curvas de entrenamiento y validación para el modelo con embeddings aprendidos.

La Figura 2.11 también presenta los reportes de clasificación detallados para ambos modelos de redes neuronales, confirmando el rendimiento superior del modelo con embeddings aprendidos en esta tarea.

```
--- Reporte de Clasificación (Modelo Pre-entrenado GloVe) ---
              precision    recall  f1-score   support

    Negativo       0.66      0.65      0.65      8528
     Neutral       0.56      0.56      0.56      8528
    Positivo       0.70      0.70      0.70      8528

    accuracy                           0.64     25584
   macro avg       0.64      0.64      0.64     25584
weighted avg       0.64      0.64      0.64     25584

800/800 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step

--- Reporte de Clasificación (Modelo con Embeddings Aprendidos) ---
              precision    recall  f1-score   support

    Negativo       0.74      0.72      0.73      8528
     Neutral       0.63      0.70      0.66      8528
    Positivo       0.82      0.75      0.78      8528

    accuracy                           0.72     25584
   macro avg       0.73      0.72      0.73     25584
weighted avg       0.73      0.72      0.73     25584
```

Figura 2.11: Reportes de clasificación en prueba para los modelos de redes neuronales.

Capítulo 3

Conclusiones Generales

Este experimento se centró en la comparación de tres enfoques distintos para el análisis de sentimientos sobre un corpus de reseñas de productos de alimentos: métodos basados en diccionarios, algoritmos de aprendizaje de máquina clásico y redes neuronales con capas de embeddings. La evaluación y comparación de estas metodologías, utilizando el dataset "Amazon Fine Food Reviews" [1], permitió obtener una visión clara sobre las fortalezas, debilidades y la aplicabilidad de cada enfoque en un escenario práctico.

3.1. Análisis de Resultados

3.1.1. Análisis Basado en Diccionarios (Lexicons)

Este enfoque, que no requiere entrenamiento, sirvió como un baseline inicial. Su rendimiento, sin embargo, fue el más bajo de todos los métodos evaluados, con exactitudes que no superaron el 46 %.

- **Opinion Lexicon** y **SentiWordNet**, aunque diseñados para análisis de sentimientos general, mostraron una capacidad limitada. Su principal deficiencia es la falta de sensibilidad al contexto, la incapacidad para manejar adecuadamente la negación (ej., "not good") y el vocabulario específico del dominio de alimentos.
- **Harvard IV-4**, orientado a textos financieros, fue completamente ineficaz para esta tarea, obteniendo la exactitud más baja (35 %). Este resultado subraya una de las mayores limitaciones de los métodos basados en léxicos: su rendimiento es altamente dependiente del dominio para el que fueron creados.

En resumen, aunque son rápidos de implementar, estos métodos son insuficientes para tareas que requieren un análisis de sentimiento preciso y matizado.

3.1.2. Análisis con Aprendizaje de Máquina Clásico

Utilizando una representación **TF-IDF** del texto, los modelos de Machine Learning clásico establecieron un estándar de rendimiento muy sólido y competitivo.

- **Regresión Logística** y **Linear SVM** demostraron ser muy efectivos, alcanzando exactitudes de prueba alrededor del **72.1 %** y **71.8 %**, respectivamente. Estos modelos lineales son eficientes y robustos, funcionando muy bien con datos de texto dispersos y de alta dimensionalidad.

- El **Árbol de Decisión** tuvo un rendimiento notablemente inferior, con una exactitud de prueba del **65.9 %**. Sin un ajuste cuidadoso de hiperparámetros (poda), los árboles de decisión son propensos a sobreajustarse y no generalizan tan bien en este tipo de tareas.

Estos resultados confirman que los modelos de ML clásico, en combinación con TF-IDF, son una opción extremadamente fuerte, ofreciendo un excelente balance entre rendimiento y coste computacional.

3.1.3. Análisis con Redes Neuronales y Embeddings

Este enfoque exploró el uso de representaciones vectoriales densas (embeddings) para capturar el significado semántico de las palabras.

- El modelo con **embeddings aprendidos** desde cero obtuvo el mejor rendimiento global del experimento, con una exactitud en prueba del **72.4 %**. Al entrenar los vectores de palabras directamente sobre el corpus de reseñas, el modelo fue capaz de crear representaciones semánticas adaptadas al vocabulario y contexto específico del dominio, superando, aunque por un margen muy estrecho, a los modelos de ML clásico.

- El modelo con **embeddings pre-entrenados de GloVe** tuvo un rendimiento inesperadamente bajo, con una exactitud del **63.8 %**, siendo superado incluso por el Árbol de Decisión. Este resultado sugiere que el conocimiento lingüístico general capturado por GloVe (entrenado en Wikipedia y noticias) no se transfirió eficazmente al dominio más coloquial y específico de las reseñas de alimentos.

3.1.4. Comparativa y Desafíos

La Figura 3.1 presenta una comparación visual de la exactitud obtenida por los cinco modelos de aprendizaje supervisado en el conjunto de prueba.

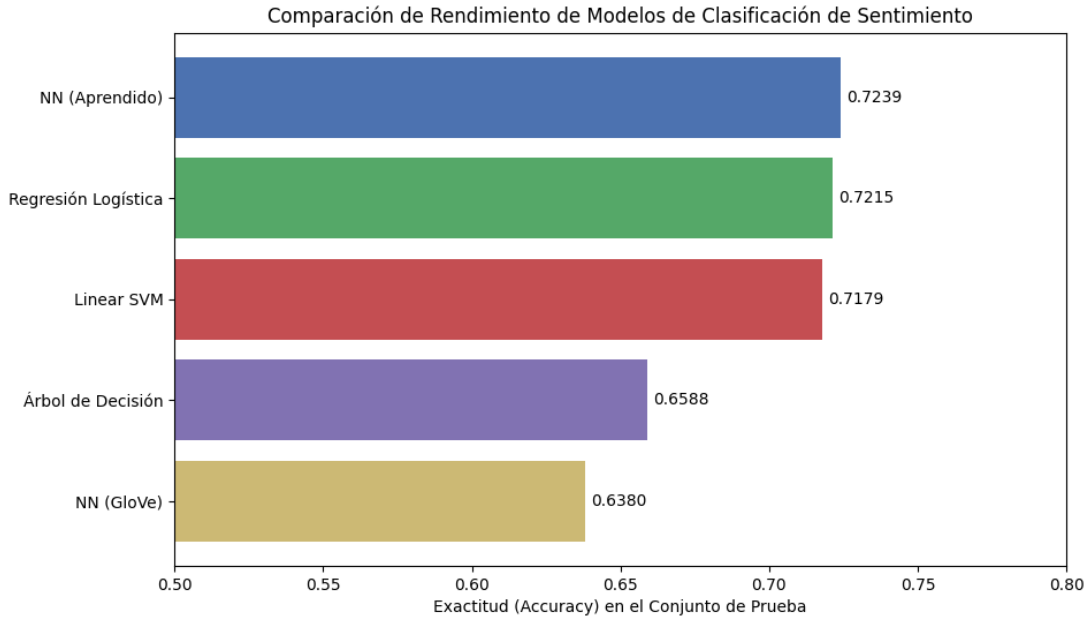

Figura 3.1: Comparación de la exactitud (Accuracy) de los modelos de ML y NN en el conjunto de prueba.

Un desafío clave en esta práctica fue el **balanceo de clases**, que resultó ser fundamental para evitar que los modelos se sesgaran hacia la clase mayoritaria (Positivo). La estrategia de submuestreo fue efectiva para crear un entorno de entrenamiento equitativo. Otro aspecto importante fue la **estrategia de normalización**; la limpieza agresiva fue beneficiosa para los modelos de aprendizaje, pero habría sido perjudicial para los métodos basados en diccionarios al eliminar palabras de negación.

3.2. Conclusiones Finales

El análisis comparativo de las diferentes metodologías para la clasificación de sentimientos en reseñas de alimentos permite extraer varias conclusiones clave:

- Los **métodos de aprendizaje supervisado** (tanto ML clásico como redes neuronales) son drásticamente superiores a los enfoques no supervisados basados en diccionarios para esta tarea. La capacidad de aprender patrones a partir de los datos es crucial para lograr un rendimiento robusto.
- Los modelos de **Machine Learning clásico**, específicamente la **Regresión Logística** y el **Linear SVM** con representación TF-IDF, demostraron ser una opción altamente eficaz y eficiente. Ofrecen un rendimiento excelente (72 %) con una menor complejidad computacional y de implementación en comparación con las redes neuronales, lo que los convierte en un baseline muy fuerte o incluso en la solución preferida en muchos escenarios prácticos.

- El modelo de **red neuronal con embeddings aprendidos** logró el mejor rendimiento del análisis (72.4 %), aunque por un margen muy pequeño. Esto sugiere que, para un dataset suficientemente grande y específico como este, permitir que el modelo cree sus propias representaciones de palabras es la estrategia más prometedora, ya que se adapta a los matices del dominio. Sin embargo, este ligero aumento en la exactitud viene acompañado de un mayor coste computacional en el entrenamiento.

- El bajo rendimiento del modelo con embeddings **GloVe** (63.8 %) es una lección importante sobre la transferencia de aprendizaje. Los embeddings pre-entrenados no son una solución garantizada; su éxito depende de la similitud entre el corpus de pre-entrenamiento y el dominio de la tarea final. En este caso, el conocimiento general de GloVe fue menos útil que las representaciones específicas generadas por TF-IDF o aprendidas desde cero.

En conclusión, para la tarea de análisis de sentimientos sobre reseñas de Amazon, una estrategia de Machine Learning clásica y bien ejecutada es casi tan buena como un enfoque de Deep Learning más complejo. Para obtener mejoras marginales en el rendimiento, las redes neuronales con embeddings entrenados específicamente para el dominio son el camino a seguir, pero la elección final dependerá siempre del balance entre la precisión deseada y los recursos computacionales disponibles.

Bibliografía

[1] J. Leskovec and A. Krevl, "SNAP Datasets: Amazon fine food reviews." http://snap.stanford.edu/data/web-FineFoods.html, June 2014.

[2] Kaggle Inc., "Kaggle: Your machine learning and data science community." https://www.kaggle.com, 2024.

[3] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2009.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, (Savannah, GA, USA), pp. 265–283, 2016.

[6] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, 2014.