Análisis de Sentimiento sobre Criptomonedas en la Comunidad Paraguaya de Reddit

Este trabajo presenta un análisis de sentimiento aplicado a comentarios en redes sociales relacionados con criptomonedas, con foco en la comunidad r/Paraguay en Reddit. El objetivo fue clasificar los comentarios en positivos, negativos y neutrales mediante técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje automático. Se exploraron varios modelos de clasificación con resultados moderados, limitados principalmente por el desbalance de clases y la escasez de muestras positivas.

Introducción

En los últimos años, el interés por las criptomonedas ha crecido significativamente en Paraguay, generando debates sobre su viabilidad, riesgos y potencial de adopción. Frente a este contexto, surgió la inquietud de explorar cuál es la percepción pública de los paraguayos sobre esta industria.

En los últimos años, las criptomonedas se han convertido en un tema recurrente en diversas plataformas digitales utilizadas en Paraguay, generando debates sobre su viabilidad, riesgos y potencial de adopción. Frente a este contexto, surgió la inquietud de explorar cuál es la percepción pública de los paraguayos sobre esta industria. El presente trabajo se propone analizar comentarios en línea para identificar patrones de sentimiento en torno a las criptomonedas.

Para ello, se eligió Reddit como fuente principal de datos, considerando que la comunidad oficial de Paraguay en esta plataforma es particularmente activa y refleja de la forma de hablar habitual en el país.

El objetivo del presente trabajo fue clasificar los comentarios relacionados con criptomonedas en tres categorías de sentimiento: positivo, negativo y neutral, aplicando

técnicas de procesamiento de lenguaje natural (NLP) y modelos clásicos de aprendizaje automático supervisado.

Metodología

Recolección de Datos

Los datos se extrajeron principalmente de Reddit, utilizando la librería *praw* en Python. Se buscaron publicaciones y comentarios que incluyen términos clave relacionados a criptomonedas:

```
# Definir palabras clave relacionadas a criptomonedas
CRYPTO_KEYWORDS = [
    "cripto", "criptomonedas", "bitcoin", "ethereum", "eth",
    "btc", "blockchain", "web3", "nft", "defi", "crypto", "minería", "minar",
    "p2p"
]
```

Dado el marcado desbalance en la distribución de clases, ya que la mayoría de los comentarios eran negativos y neutrales, se intentó aumentar la cantidad de ejemplos positivos incorporando datos adicionales provenientes de plataformas como X, y Facebook. Sin embargo, el sesgo en la representación de clases se mantuvo.

Limpieza y Preprocesamiento

Se eliminaron menciones, URLs, emojis, y se aplicó una tokenización con spaCy en español. También se removieron stopwords para reducir ruido textual:

```python
def clean_text(text):
    # 1. Eliminar menciones, URLs y caracteres especiales innecesarios
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)  # @usuario
    text = re.sub(r'https?://\S+', '', text)  # URLs
    text = re.sub(r'\n', ' ', text)  # Saltos de línea
    text = re.sub(r'[^A-Za-zÁÉÍÓÚÑáéíóúñ0-9\s\$]', '', text)  # Elimina emojis y símbolos

    # 2. Pasar a minúsculas
    text = text.lower()

    # 3. Tokenizar con spaCy
    doc = nlp(text)

    # 4. Filtrar tokens
    tokens = [
        token.text
        for token in doc
        if token.text.lower() not in stopwords_es  # eliminar stopwords
        and not token.is_punct
        and not token.is_space
    ]

    return " ".join(tokens)
```

Etiquetado del Sentimiento

Los comentarios fueron etiquetados manualmente en tres categorías de acuerdo a los siguientes criterios:

Etiqueta	Descripción
0 Positivo	Entusiasmo o experiencias favorables
1 Negativo	Desconfianza o críticas
2 Neutral	Consultas, hechos o menciones sin carga emocional

La distribución resultante fue la que sigue:

```
df["sentimiento"].value_counts()
```

```
sentimiento
NEUTRAL     189
NEGATIVO    104
POSITIVO     70
Name: count, dtype: int64
```

Vectorización y Modelado

Los textos fueron vectorizados utilizando la técnica TF-IDF mediante la clase TfidfVectorizer. A continuación, se entrenaron modelos clásicos de clasificación supervisada, seleccionados por su efectividad en tareas de procesamiento de lenguaje natural: Logistic Regression, Random Forest y Linear SVC.

Resultados

Los resultados de cada modelo se detallan a continuación:

Logistic Regression

```
# Predecir y evaluar
y_pred = model.predict(X_test_tfidf)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.6712328767123288
              precision    recall  f1-score   support

           0       0.45      0.36      0.40        14
           1       0.63      0.81      0.71        21
           2       0.77      0.71      0.74        38

    accuracy                           0.67        73
   macro avg       0.62      0.63      0.62        73
weighted avg       0.67      0.67      0.67        73
```

Linear SVC

```
[52]: # Predecir y evaluar
y_pred = model.predict(X_test_tfidf)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.6712328767123288
              precision    recall  f1-score   support

           0       0.71      0.36      0.48        14
           1       0.57      0.76      0.65        21
           2       0.74      0.74      0.74        38

    accuracy                           0.67        73
   macro avg       0.67      0.62      0.62        73
weighted avg       0.68      0.67      0.66        73
```

Random Forest

```
# Predecir y evaluar
y_pred = model.predict(X_test_tfidf)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.6027397260273972
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        14
           1       0.75      0.29      0.41        21
           2       0.58      1.00      0.74        38

    accuracy                           0.60        73
   macro avg       0.44      0.43      0.38        73
weighted avg       0.52      0.60      0.50        73
```

Los resultados muestran que los modelos lineales (Logistic Regression y Linear SVC) se desempeñan mejor en contextos con clases desbalanceadas, al menos en comparación con

Random Forest. El principal problema es el desbalance de las clases, todos los modelos tuvieron inconvenientes para detectar los comentarios positivos,

Conclusiones

Este estudio demuestra la viabilidad de aplicar modelos clásicos de aprendizaje automático al análisis de sentimiento en español en contextos específicos como Paraguay. Sin embargo, la limitada cantidad de datos y el sesgo en la distribución de clases restringen la precisión.

Para mejorar los resultados, se propone:

- Aumentar la recolección de muestras positivas (p. ej. campañas dirigidas, foros especializados).
- Aplicar técnicas de balanceo.
- Ajustar hiperparámetros y usar validación cruzada más extensa.