Análisis de Sentimiento, Arboles de decisión, Lexicones

Felipe Olivares
Jose Escobar
Steven Garcia
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia

Palabras cláve:—Arboles de decisión, Análisis de sentimiento, Adaboost

1. INTRODUCCIÓN

El análisis de sentimiento, también se conoce como minería de opinión, en palabras simples se trata de una tarea de clasificación a gran escala y automáticamente, que se centra en catalogar los documentos o frases suministradas en función de la connotación positiva, negativa o neutral del mismo. En el desarrollo del presente documento, se evidencia una implementación de tres modelos distintos para el análisis de sentimiento, Modelo Diferencial, Modelo No Supervisado y el Modelo de implementación normal usando arboles de decisión, optamos por presentar recursos visuales que provienen de los ejemplos suministrados en clase con el fin de lograr un mayor entendimiento de los algoritmos utilizados.

	X		y
Outlook	Humidity Nominal	Windy	Play
overcast	high	FALSE	yes
overcast	normal	TRUE	yes
overcast	high	TRUE	yes
overcast	normal	FALSE	yes
rainy	high	FALSE	yes
rainy	normal	FALSE	yes
rainy	normal	TRUE	no
rainy	normal	FALSE	yes
rainy	high	TRUE	no
sunny	high	FALSE	no
sunny	high	TRUE	no
sunny	high	FALSE	no
sunny	normal	FALSE	yes
sunny	normal	TRUE	yes

Figura 1: Ejemplo Árbol decisión

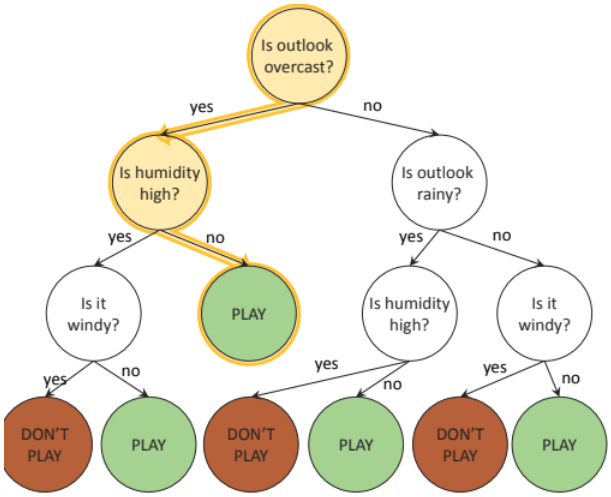


Figura 2: Ejemplo Árbol de decisión

Funciona minimizando la función de error, solo se actualizan los pesos cuando, la predicción es diferente a la etiqueta real, no obstante, se debe tener en cuenta que los pesos que fallaron en la predicción, no se contemplan en la sumatoria.

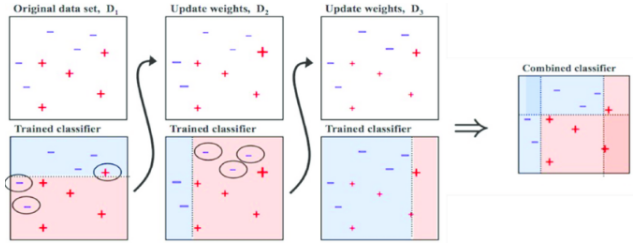


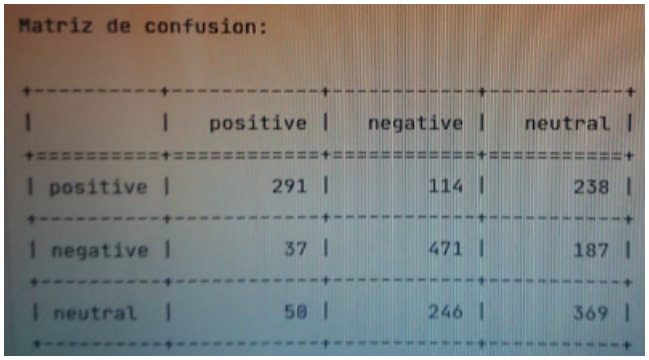
Figura 3: Adaboost

Resultados luego de la división del dataframe original balanceando.

```
Tweets en el dataframe original: 16140
Tweets en el dataframe balanceado: 8010
Numero de datos para la prueba: 2003
Precision promedio: 0.6307800448680917
Recall promedio: 0.6518261187829495
```

Figura 4: Separación

Resultados obtenidos luego de implementar arboles de decisión.



Matriz de confusion:

	positive	negative	neutral
positive	291	114	238
negative	37	471	187
neutral	50	246	369

Figura 5: Separación

Se opta por mejorar la técnica de arboles de decisión añadiendo el algoritmo Adaboost.

Matriz de confusion:

	positive	negative	neutral
positive	379	65	199
negative	53	430	212
neutral	75	135	455

Metricas de desempeño:

	precision	recall
positive	0.589425	0.747535
negative	0.618705	0.68254
neutral	0.684211	0.525404

Figura 6: Arboles de decisión junto al algoritmo Adaboost

Con el fin de hacer uso de los Lexicones, se opto por investigar en la WEB, implementaciones de Lexicones en Ingles populares, como resultado se determino utilizar SentiWordNet que hacer parte de la biblioteca NLTK ”nltk/corpus/reader/sentiwordnet.html˝ AFFIN obtenido del repositorio Git ”Tweet-Dissectionçon la licencia de ”https://www.opendatacommons.org/licenses/odbl/1.0/”. En la implementación se opto por vectorizar en primera instancia toda la data propuesta, por consiguiente, se desarrollo un modulo para amplificar la vectorización uniéndola con los lexicones mencionados con anterioridad, este fragmento de aumento fue obtenido proveniente de esta investigación. Descubrimos que al utilizar lexicones, el uso de un RandomForest con 200 modelos generados, es mas preciso que el uso de un AdaBoost Classifier, el gasto computacional es mayor, no obstante estos resultados en gran medida pueden ser por las diferentes métricas de entrenamiento . Se presentan los resultados obtenidos luego de implementar arboles de decisión con Adaboost.

Numero de datos para la prueba: 2003
Precision promedio: 0.6412887756738784
Recall promedio: 0.6453207728679428

Matriz de confusion:

	positive	negative	neutral
positive	415	87	152
negative	56	472	150
neutral	112	161	398

Metricas de desempeño:

	precision	recall
positive	0.634557	0.711835
negative	0.696165	0.655556
neutral	0.593145	0.568571

Figura 7: Metricas para Adaboost

Se evidencia que al utilizar RandomForest con 300 modelos, el clasificador tiende a ser mas preciso, cabe resaltar que el gasto computacional es proporcional, ademas de que toda la data utilizada esta normalizada.

```
Numero de datos para la prueba: 2003
Precision promedio: 0.6986283361285525
Recall promedio: 0.6997215930063646

Matriz de confusion:
+----------+------------+------------+-----------+
|          |   positive |   negative |   neutral |
+==========+============+============+===========+
| positive |        450 |         68 |       136 |
+----------+------------+------------+-----------+
| negative |         47 |        519 |       112 |
+----------+------------+------------+-----------+
| neutral  |        114 |        126 |       431 |
+----------+------------+------------+-----------+

Metricas de desempeño:
+----------+-------------+----------+
|          |   precision |   recall |
+==========+=============+==========+
| positive |    0.688073 | 0.736498 |
+----------+-------------+----------+
| negative |    0.765487 | 0.72791  |
+----------+-------------+----------+
| neutral  |    0.642325 | 0.634757 |
+----------+-------------+----------+
```

Figura 8: Métricas para RandomForest 300 modelos

2. CONCLUSIONES

- Se evidencio una mejora absoluta y diferenciable en la matriz de confusión al momento de realizar la implementación primero, con Arboles de decisión y segundo añadiendole el algoritmo AdaBoost, gracias a la actualización de pesos que mejora la identificación.
- Los lexicones permiten un mayor peso al momento de analizar oraciones, se tienen en cuenta palabras especificas.

3. Bibliografia

[1] Minqing Hu and Bing Liu. ”Mining and Summarizing Customer Reviews.”Proceedings of the ACM SIGKDD International Conference on Knowledge. Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA.

[2] NLTK. Çode for nltk.corpus.reader.sentiwordnet- ecuperado de: ”https://www.nltk.org/modules/nltk/corpus/reader/sentiwordnet.html”

[3] Jake VanderPlas. Ïn-Depth: Decision Trees and Random Forests.Recuperado de: https://jakevdp.github.io/ PythonDataScienceHandbook/ 05.08-random-forests.html