

Análisis comparativo entre diferentes entornos de aprendizaje automático para el análisis de sentimientos

Karen L. Vazquez¹, Mireya Tovar¹, José A. Reyes-Ortiz², Darnes Vilariño¹

¹Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación, Puebla, México

²Universidad Autónoma Metropolitana,
Departamento de Sistemas, Azcapotzalco, México

karnlet@gmail.com, {mtovar, darnes}@cs.buap.mx, jaro@correo.azc.uam.mx
<http://www.lke.buap.mx/>

Resumen. El objetivo en este artículo es el estudio del Análisis de Sentimientos con aprendizaje automático y precisamente realizar una comparativa de clasificadores automáticos implementados en la plataforma de software para el aprendizaje automático, WEKA y el lenguaje de programación Python. Los experimentos se elaboraron con tres conjuntos de datos provenientes de SemEval 2016 para la resolución de la tarea 5, subtarea 2 [9]. El enfoque utilizado para el análisis de sentimiento está dividido en 5 fases y los resultados muestran que Python obtiene un mejor desempeño con los clasificadores utilizados.

Palabras clave: Análisis de sentimientos, procesamiento de lenguaje natural, aprendizaje automático.

Comparative Analysis of different Machine Learning Environments for Sentiment Analysis

Abstract. The main aim of this paper is the study of Sentiment Analysis with machine learning techniques and to perform a comparison of automatic classifiers implemented in WEKA and Python programming language. The experiments were carried out with three data sets from SemEval 2016 for solving the task 5, subtask 2 [9]. The approach proposed for sentiment analysis is divided into five phases and the results show that Python achieves a better performance than WEKA with the classifiers used.

Keywords. Sentiment analysis, natural language processing, machine learning.

1. Introducción

El Análisis de Sentimientos es una rama del Procesamiento de Lenguaje Natural en el cual se identifica una emoción a través de una oración, frase o expresión escrita en Internet, permitiendo el monitoreo de opiniones sobre diferentes temas discutidos en la Web. En el caso del estudio abordado en este artículo, se analizaron frases u oraciones escritas en español e inglés en las que se opina sobre el servicio de Restaurantes y opiniones escritas en el idioma inglés sobre Laptops. Se realizaron experimentos utilizando 3 clasificadores automáticos: Máquina de Soporte Vectorial (SVM), Naïve Bayes y Naïve Bayes Multinomial, cada uno se prueba con los tres conjunto de datos en el software de aprendizaje automático Weka y en Python, con el fin de realizar una comparación de resultados entre estas dos herramientas.

El artículo se encuentra distribuido de la siguiente manera: en la Sección 2. se presentan algunos trabajos relacionados con el análisis de sentimientos; en la Sección 3. se presenta la parte teórica que sustenta esta investigación; en la Sección 4. se presenta las fases que integran al enfoque propuesto como solución al análisis de sentimientos; en la Sección 5. se muestran los resultados experimentales obtenidos; en la Sección 6. se presenta una comparación entre Python y Weka de acuerdo a los resultados obtenidos en los experimentos y finalmente en la Sección 7. se presentan las conclusiones de esta investigación.

2. Trabajos relacionados

Existen diferentes métodos para el estudio de Análisis de Sentimientos, sin embargo, lo más estudiado es el Aprendizaje Automático y en la literatura existen trabajos que usan diferentes formas de utilizar dicho aprendizaje. A continuación se muestran algunas participaciones en la investigación de la Minería de Opiniones.

Para la solución de una de las tareas de SemEval 2016, en [14] proponen un sistema de aprendizaje para la clasificación de tweets en dos puntos escala, con una arquitectura que consiste en una red neuronal multicapa, dicha red es alimentada de tweets pre procesados como entrada y está predice las etiquetas binarias de los tweets. Los tweets son pre procesados: todos los URLs son codificados dentro de un token URL, las cuentas de usuarios mencionadas en un tweets son codificadas como un token USER, antes de alimentar a la red el tweet se convierte en minúsculas, no se remueven algunas palabras vacías ya que definen relaciones útiles entre palabras y frases, seguido del pre procesado continua la capa de incrustación de los mapas de una secuencia de palabras presentes en el tweet de entrada a la longitud fija correspondiente del valor real de los vectores. En esta investigación utilizan las redes neuronales recurrentes en la arquitectura del sistema, un apilamiento de las capas recurrentes en la parte superior de uno al otro permite la composición de las representaciones semánticas de palabras y frases con el tiempo. La capa *suma* ayuda a combinar la representación del sentimiento específico de las frases a fin de producir una

representación agregada y la salida de esa capa esta conectada con la capa densa que consiste en unidades lineales rectificadas. La capa densa está conectada con la salida de unidades sigmoides que predice la probabilidad de asignar una etiqueta positiva o negativa para el tweet de entrada. Para los experimentos de la red utilizan 30 como longitud máxima de secuencia de entrada, vocabulario de tamaño 400,000 dimensionalidad (d) de incrustación de palabra es de 100, 3 capas recurrentes y se utilizó una deserción del 50% después de cada capa durante el entrenamiento. El sistema produce una recuperación macro promedio de 0.784, mientras que el mejor sistema obtuvo 0.797 de puntaje.

En el trabajo presentado por [10], se propone una metodología para el análisis de sentimientos multimodales, consiste en sentimientos de vídeos de la Web mediante la demostración de un modelo que utiliza modalidades de audio, visuales y textuales como fuentes de información. La extracción de características visuales se hace por medio de un analizador de expresión facial y clasifica las expresiones faciales con el fin de definir las categorías de sentimientos, se utilizaron polaridades positivas, negativas y neutras como clases de sentimientos. En el experimento se utilizan las características extraídas por FSDK 1.7 junto con las extraídas utilizando GAVAM, usaron ELM para construir el modelo de análisis de sentimientos de expresiones faciales. Se produjo una precisión de 68.60%. Para las características de audio se extrajeron mediante un marco del tipo de 30 Hz y una ventana deslizante de 100 ms y para calcular las características se utilizó el software OpenEar. En el caso de la identificación de los sentimientos en el texto se siguió el paradigma de computación Sentic, el cual considera el texto como expresión de ambos: *semantics* y *sentics*. Los experimentos se realizaron con el conjunto de datos de Youtube y con diferentes clasificadores supervisados: Naïve Bayes, SVM, ELM, redes neuronales y los mejores resultados se obtuvieron con el clasificador ELM.

En [6] se llevó a cabo una metodología para determinar la posible popularidad, opinión y sentimiento de un producto en diferentes localidades a través de usuarios de género masculino y femenino, el análisis es en tweets sobre el iPhone 6. La extracción de estos tweets se realizó por medio de la API de Twitter, estos tweets contienen datos no relevantes es por eso que los autores realizaron una limpieza básica usando Java. Utilizaron una herramienta de Procesamiento de Lenguaje Natural de Stanford y SentiWordNet. Para determinar el género de los usuarios de los tweets utilizaron *NamSor* que es una herramienta de minería de datos y *Rapid Miner* como extensión para la clasificación de género. Los autores realizaron comparaciones con diferentes escenarios del mundo real, y los resultados muestran sentimientos negativos hacia la pantalla y el touch del iPhone 6, y sentimientos positivos hacia la cámara que es bien recibida por los usuarios generales y revisores.

Siguiendo la técnica de aprendizaje automático con datos de twitter, en [13] automáticamente proporcionan la opinión sobre un producto. El pre procesamiento de datos es la parte más importante en el proceso y comprende el reemplazo de emoticones, gestión de URL, hash-tags, espacios en blanco, identificación de puntuación y conversión de mayúsculas. En la extracción de datos se utilizó

la técnica basada a nivel de oración y consta de Tokenización, POS tagger, Stopwords, SentiWordNet, clasificación y evaluación usando el clasificador SVM, el sistema calcula el porcentaje de tweets positivos y negativos de un producto en particular.

En el caso del trabajo desarrollado en [7] se hace uso de tweets, como las reseñas escritas por usuarios hacia una película en particular. En el pre procesado de los tweets hacen uso de Procesamiento de Lenguaje Natural (por sus siglas en inglés PLN), este pre procesado se refiere a la limpieza y normalización del texto, para hacer el análisis de sentimientos (stop words, puntuación, palabras duplicadas, caracteres repetidos, emoticones y acrónimos de Internet y, URLs). Eligieron el clasificador Naïve Bayes como técnica aplicada para ciertas clases de problemas. Asignan pesos a varios factores como el número de vistas en cada película, el número de tweets, etc., los pesos de todos los factores son combinados para predecir el éxito total de taquilla sobre la película.

Para el desarrollo de este trabajo, se consideran tres clasificadores de aprendizaje automático supervisado, algunos de ellos son utilizados en la literatura mencionada anteriormente y muestran buenos resultados en el Análisis de Sentimientos. Máquina de Soporte Vectorial (SVM), Naïve Bayes y Naïve Bayes Multinomial son los estudiados en esta investigación, para cada uno se utiliza como característica léxica a los unigramas y un esquema de pesado como TF-IDF, se emplean tres conjuntos de datos con dos dominios y dos idiomas. Adicionalmente se realiza una comparación entre dos tipos de lenguajes de aprendizaje automático como son Python y Weka.

3. Aprendizaje supervisado

Como clasificador de polaridad se utiliza aprendizaje automático supervisado, el cual comprende algoritmos automáticos de clasificación los aplicados son: Máquina de Soporte Vectorial (SVM), Naïve Bayes y Naïve Bayes Multinomial.

3.1. Máquina de soporte vectorial

Máquina de Soporte Vectorial (SVM) es una técnica para la clasificación de datos [2]. El objetivo de las máquinas de soporte vectorial es producir un modelo que se base en los datos de entrenamiento que predice los valores objetivo de los datos de prueba, dado únicamente los atributos de los datos de prueba. Las máquinas de soporte vectorial en Python cuentan con métodos de aprendizaje supervisado utilizados para la clasificación, regresión y detección de valores atípicos. La herramienta **scikit-learn** [3] de Python es una máquina de aprendizaje en Python, que proporciona simples y eficientes herramientas para la minería de datos y análisis de sentimientos, contiene varios *vectorizers*¹ de traducción para los documentos de entrada en vectores de características.

¹ Función utilizada en scikit-learn <http://scikit-learn.org/>

3.2. Naïve Bayes

El clasificador Bayesiano, asigna la clase más probable dado un ejemplo descrito por su vector de características [12]. Naïve Bayes es un método sencillo e intuitivo con un funcionamiento similar a otros enfoques. Naïve Bayes combina la eficiencia (tiempo óptimo de rendimiento) con una razonable precisión [5]. Este clasificador tiene como inconveniente asumir independencia condicional entre los rasgos lingüísticos. Si las características principales son los tokens extraídos de los textos es evidente que no pueden considerarse como independientes, ya que las palabras co-ocurren en un texto siendo de alguna manera unidas por diferentes tipos de dependencias sintácticas y semánticas. Pero incluso si Naïve Bayes produce un modelo muy simplificado, sus decisiones de clasificación son precisas. Este modelo probabilístico está basado en la regla de Bayes, junto con un fuerte supuesto de independencia, dicho supuesto es que dada una clase (positiva o negativa) las palabras son condicionalmente independientes entre sí, el supuesto no afecta tanto a la exactitud en la clasificación del texto, pero hace más rápido los algoritmos de clasificación aplicables para el problema [8]. Python permite el uso del clasificador Naïve Bayes con el kit de herramientas de lenguaje natural (NLTK)².

3.3. Naïve Bayes multinomial

Naïve Bayes Multinomial es un modelo de distribución de palabras en un documento como un polinomio. Un documento se trata como una secuencia de palabras y se supone que cada posición de la palabra se genera independientemente de cualquier otra. Es un clasificador rápido, fácil de implementar y relativamente eficaz [11]. El modelo Naïve Bayes Multinomial permite considerar la frecuencia de aparición de cada término en los documentos, esto es importante, ya que podemos suponer que una alta frecuencia de aparición aumenta la probabilidad de pertenecer a una clase particular [1].

4. Enfoque propuesto

Para los objetivos planteados se realizan las siguientes etapas o fases: Pre procesamiento, Extracción de características, Fase de entrenamiento, Fase de prueba y Evaluación. La característica principal es el uso de Frecuencia Inversa del Documento para un Término (TF-IDF), la cual representa el número de documentos en el cual un término dado es calculado y para clasificar las polaridades en las opiniones se utiliza aprendizaje automático supervisado con los clasificadores: Máquina de Soporte Vectorial, Naïve Bayes y Naïve Bayes Multinomial. A continuación se describe a detalle las tres fases propuestas.

- Pre procesamiento:

² <http://www.nltk.org/book/ch06.html>

- **Colección de opiniones.** Extraer únicamente opiniones de los documentos en formato XML.
- **Purificación de opiniones.** Proceso en el que se obtienen las opiniones libres de palabras vacías, signos de puntuación, acentos y caracteres aislados.
- **Tokenización.** Tokenizar cada opinión por palabra.
- **Stemming.** Proceso en el cual se reducen las palabras de cada opinión, frecuentemente se incluye la eliminación de los afijos derivados.
- **Filtrado de opiniones.** En esta parte se clasifican las opiniones de entrenamiento por las posibles polaridades: positivo, negativo, neutral y conflicto.
- Extracción de características: Se considera como característica léxica Frecuencia Inversa del Documento para un Término (TF-IDF por sus siglas en inglés) que es una técnica que indica la relevancia de una palabra con respecto al documento seleccionado y al corpus en general, lo cual permite calificar a los documentos del corpus con base a las palabras claves, es decir, si las palabras tiene mayor peso, entonces significa que el documento está más relacionado con las palabras que uno con las mismas palabras pero con menor peso.
- Fase de entrenamiento: Con las características obtenidas en la fase anterior se realiza el proceso de entrenamiento. De acuerdo al algoritmo de clasificación considerado se construye el modelo que posteriormente se utiliza en la fase de prueba. Las clases consideradas son cuatro: positiva, negativa, neutro y conflicto. Los algoritmos de clasificación supervisado usados en esta fase, tanto en WEKA como en Python, son: Máquina de Soporte Vectorial, Naïve Bayes y Naïve Bayes Multinomial.
- Fase de prueba: Los datos de prueba son clasificados de acuerdo al modelo propuesto por el clasificador. La polaridad propuesta a cada opinión de los datos de prueba son comparadas con los datos del gold estándar.
- Evaluación: Para medir los resultados de los experimentos se utiliza como medidas de evaluación *Precisión* y *Recall* para medir los resultados del algoritmo utilizando el clasificador SVM, los resultados del clasificador Naïve Bayes Multinomial, y la medida de *Accuracy* se utiliza para los resultados del algoritmo con el clasificador Naïve Bayes.

Precisión y *Recall* son medidas que se basan en la comparación de un resultado esperado y el resultado efectivo del sistema evaluado [4]. Estas medidas han sido adaptadas para la evaluación de la clasificación en el análisis de sentimientos. La precisión se mide con la Ecuación (1) y el *recall* con la Ecuación (2):

$$Precision = \frac{t_p}{t_p + f_p}, \quad (1)$$

$$Recall = \frac{t_p}{t_p + f_n}, \quad (2)$$

Donde:

t_p : verdaderos positivos,

t_n : verdaderos negativos,
 f_p : falsos positivos,
 f_n : falsos negativos.

La medida armónica entre el *recall* y la *precisión* es la función F_1 (ver Ecuación 3):

$$F_1 = \frac{2pr}{(p + r)}, \quad (3)$$

Donde:

p : *Precision*,
 r : *Recall*.

La medida **Accuracy** (exactitud) se refiere a la evaluación del sesgo de las predicciones, es decir, responde a la pregunta, ¿Cuál es el promedio de las predicciones correctas? La ecuación de la exactitud se muestra en la fórmula (4):

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}. \quad (4)$$

A continuación se presentan los resultados obtenidos con el enfoque propuesto.

5. Resultados obtenidos

Esta sección describe los datos que se usaron para probar el enfoque propuesto, también se muestran los resultados obtenidos aplicando lo mencionado anteriormente para cada conjunto de datos.

5.1. Conjunto de datos

Los datos utilizados para esta investigación se tomaron de los conjuntos de datos proporcionados por Semeval 2016 para la solución de la tarea 5, subtarea 2 [9]. Se trabajó con 3 conjuntos diferentes: Conjunto de opiniones para Restaurantes en el idioma español, conjunto de opiniones para Restaurantes en el idioma inglés y conjunto de opiniones para Laptops en el idioma inglés. En la Tabla 1 se muestran ejemplos de opiniones recuperadas de los datos de prueba para cada dominio.

En la Tabla 2 se muestra la cantidad de opiniones en los datos de entrenamiento y en los datos de prueba por cada dominio. En la Tabla 3 se presenta el total de opiniones por tipo de polaridad: positivo (*Pos*), negativo (*Neg*), neutral (*Neu*) y conflicto (*Con*), en cada dominio de los datos de entrenamiento.

5.2. Resultados experimentales

El objetivo de este artículo es comparar el uso de clasificadores en distintos entornos, Weka y Python, probando los mismos conjuntos de datos en cada uno.

Tabla 1. Ejemplos de opiniones para cada dominio.

<i>Dominio</i>	<i>Opiniones</i>
Restaurantes (Español)	La comida estuvo muy sabrosa. Quien sea amante de la carne tiene una carta bastante amplia para elegir., aunque ayer no tenían chuleton. Lo único que nos sorprendió es que nos sirvieran los entrantes y los platos principales a la vez.
Restaurantes (Inglés)	Yum!Serves really good sushi. Not the biggest portions but adequate. Green Tea creme brulee is a must! Don't leave the restaurant without it.
Laptops (Inglés)	Well, my first apple computer and I am impressed. Works well, fast and no reboots. Waiting to install MS Office and see how it goes from there. Have always been a PC guy, but decided to try Apple. Glad I did so far.

Tabla 2. Total de opiniones por conjunto de datos y dominio.

<i>Dominio</i>	<i>Entrenamiento</i>	<i>Prueba</i>	<i>Gold</i>
Restaurantes (Español)	2,121	881	881
Restaurantes (Inglés)	1,435	404	404
Laptops (Inglés)	2,082	545	545

Tabla 3. Conjunto de datos de entrenamiento por polaridad.

<i>Dominio</i>	<i>Pos</i>	<i>Neg</i>	<i>Neu</i>	<i>Con</i>	<i>Total</i>
Restaurantes (Español)	1,519	443	101	58	2,121
Restaurantes (Inglés)	1,012	327	55	41	1,435
Laptops (Inglés)	1,210	707	123	41	2,081

Para medir los resultados en estos experimentos se utiliza como medidas de evaluación *Precisión* y *Recall* para medir los resultados de SVM y Naïve Bayes Multinomial, y la medida de *Acurracy* usado únicamente para los resultados obtenidos con el clasificador Naïve Bayes.

Resultados en Python. Para el caso del clasificador automático Máquina de Soporte Vectorial (SVM) en Python, el mejor resultado se logra con el conjunto de datos del dominio de Restaurantes en Español obteniendo el 69% de precisión, de la misma manera en el caso del clasificador Naïve Bayes Multinomial obteniendo el 67% de precisión. En el caso de Naïve Bayes se logra el 72% de exactitud.

En las Tablas 4-6 se muestran con detalle los resultados obtenidos en cada clasificador probado y en cada conjunto de datos.

Resultados en Weka. Probando los datos en el entorno de Weka, los resultados obtienen un menor porcentaje de *Acurracy* en el dominio de Laptops en el idioma inglés. Sin embargo, para el conjunto de Restaurantes en el idioma español se alcanza el 0.69% de precisión con el clasificador Naïve Bayes Multinomial y

Tabla 4. Resultados obtenidos en Python y SVM.

Dominio	<i>Precisión</i>	<i>Recall</i>	<i>F₁</i>
Restaurantes (Español)	0.69	0.78	0.73
Restaurantes (Inglés)	0.69	0.76	0.72
Laptops (Inglés)	0.66	0.72	0.68

Tabla 5. Resultados obtenidos en Python y Naïve Bayes Multinomial.

Dominio	<i>Precisión</i>	<i>Recall</i>	<i>F₁</i>
Restaurantes (Español)	0.67	0.73	0.69
Restaurantes (Inglés)	0.64	0.71	0.67
Laptops (Inglés)	0.66	0.71	0.68

Tabla 6. Resultados obtenidos en Python y Naïve Bayes.

Dominio	<i>Acuraccy</i>
Restaurantes (Español)	0.72
Restaurantes (Inglés)	0.70
Laptops (Inglés)	0.55

SVM. A continuación se muestran las Tablas 7, 8 y 9 con los resultados obtenidos para cada conjunto de datos.

Tabla 7. Resultados obtenidos en Weka y SVM.

Dominio	<i>Precisión</i>	<i>Recall</i>	<i>F₁</i>
Restaurantes (Español)	0.69	0.77	0.72
Restaurantes (Inglés)	0.67	0.75	0.70
Laptops (Inglés)	0.60	0.66	0.63

Tabla 8. Resultados obtenidos en Weka y Naïve Bayes Multinomial.

Dominio	<i>Precision</i>	<i>Recall</i>	<i>F₁</i>
Restaurantes (Español)	0.69	0.77	0.73
Restaurantes (Inglés)	0.70	0.76	0.73
Laptops (Inglés)	0.67	0.66	0.66

Tabla 9. Resultados obtenidos en Weka y Naïve Bayes.

Dominio	Acuraccy
Restaurantes (Español)	0.71
Restaurantes (Inglés)	0.59
Laptops (Inglés)	0.40

6. Comparación de resultados entre Python y Weka

Teniendo los resultados con cada entorno y clasificador, con base en los resultados de la medida armonica F_1 y *accuracy* se observa que SVM y Naïve Bayes tienen un mejor comportamiento en Python, y con Naïve Bayes Multinomial el algoritmo se comporta mejor en Weka (ver Tabla 10). Consideramos que se obtiene un mejor comportamiento de aprendizaje automático utilizando Python, puesto que la extracción de características se realizan automáticamente usando las herramientas para el análisis de datos y procesamiento del lenguaje natural *scikit-learn* y *NLTK*.

Tabla 10. Comparación entre Python y Weka.

Dominio	SVM		NB Multinomial		Naïve Bayes	
	Weka	Python	Weka	Python	Weka	Python
	F_1	F_1	F_1	F_1	<i>Accuracy</i>	<i>Accuracy</i>
Restaurantes (Español)	0.72	0.73	0.73	0.69	0.71	0.72
Restaurantes (Inglés)	0.70	0.72	0.73	0.67	0.59	0.70
Laptops (Inglés)	0.63	0.68	0.66	0.68	0.40	0.55

En las Figuras 1, 2 y 3 se muestran las gráficas con los resultados de los experimentos entre ambos entornos y tipo de clasificador.

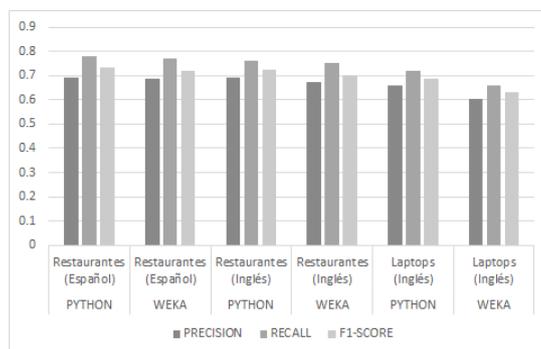


Fig. 1. Resultados con Máquina de Soporte Vectorial.

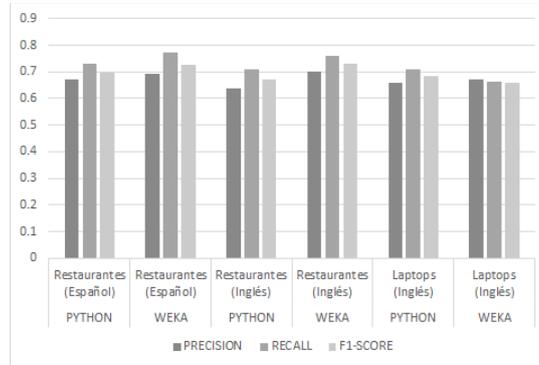


Fig. 2. Resultados con Naïve Bayes Multinomial.

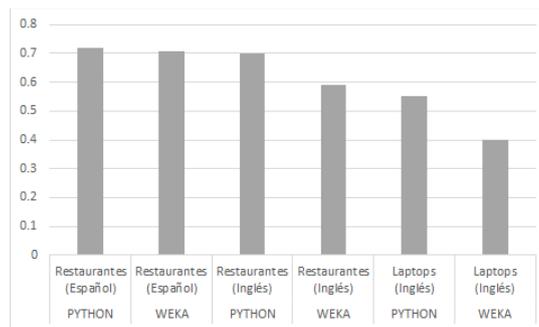


Fig. 3. Resultados con Naïve Bayes.

7. Conclusiones

En esta investigación se presentan los resultados obtenidos en el análisis de sentimientos utilizando tres clasificadores de aprendizaje supervisado: Máquina de Soporte Vectorial, Naïve Bayes y Naïve Bayes Multinomial. Estos algoritmos son utilizados para clasificar opiniones de los dominios de Restaurantes y Laptops, por cada opinión se detecta una de las cuatro posibles polaridades: positivo, negativo, neutral y conflicto. También se utiliza Python con *nlTK* y *scikit learn*, y Weka. Con base en los resultados obtenidos se muestra que los mejores resultados son con los clasificadores diseñados en Python. Logrando un 69% de precisión para el dominio de Restaurantes en Español con el clasificador SVM en Python, el 70% de precisión en el dominio de Restaurantes en Inglés con el clasificador Naïve Bayes Mutinomial en Weka y el 67% de precisión para el dominio de

Laptops en Weka con Naïve Bayes Multinomial. En el caso de la exactitud en los tres dominios son mejores los resultados en Python. Consideramos que se tiene este comportamiento debido a que la extracción de características, en el caso de Python, son con las herramientas interconstruidas en el lenguaje. Así mismo se concluye que con las pruebas realizadas en este trabajo de investigación el mejor clasificador es SVM con Python, Naïve Bayes Multinomial con Weka y Naïve Bayes con Python. Como trabajos a futuro consideramos el uso de otros tipos de características así como el empleo de herramientas de medición de polaridad de las palabras como SentiWordNet.

Agradecimientos. Esta investigación es parcialmente apoyada por el proyecto PRODEP-SEP ID 00570 (EXB-792) DSA/103.5/15/10854, por el proyecto ID 00570 VIEP-BUAP. Apoyado por el Fondo Sectorial de Investigación para la Educación, proyecto Conacyt 257357.

Referencias

1. Bermejo, P., Gámez, J.A., Puerta, J.M.: Improving the performance of naive bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. *Expert Systems with Applications* 38(3), 2072 – 2080 (2011), <http://www.sciencedirect.com/science/article/pii/S0957417410007748>
2. Betancourt, G.A.: Las máquinas de soporte vectorial (svms). *Scientia Et Technica* 27 XI, 67–72 (2005)
3. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. pp. 108–122 (2013)
4. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: *IJCAI*. pp. 348–353 (2007)
5. Gamallo, P., Garcia, M.: Citius: A naive-bayes strategy for sentiment analysis on english tweets. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. pp. 171–175. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (August 2014)
6. Hridoy, S.A.A., Ekram, M.T., Islam, M.S., Ahmed, F., Rahman, R.M.: Localized twitter opinion mining using sentiment analysis. *Decision Analytics* 2(1), 1 (2015)
7. Mulay, S.A., Joshi, S.J., Shaha, M.R., Vibhute, H.V., Panaskar, M.P.: Sentiment analysis and opinion mining with social networking for predicting box office collection of movie. *International Journal of Emerging Research in Management & Technology* 5(1), 74–79 (2016)
8. Narayanan, V., Arora, I., Bhatia, A.: Fast and accurate sentiment classification using an enhanced naive bayes model. *Lecture Notes in Computer Science* 8206, 194–201 (2013)
9. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryigit, G.: Semeval-2016 task 5: Aspect based sentiment analysis. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*

- (SemEval-2016). pp. 19–30. Association for Computational Linguistics, San Diego, California (June 2016), <http://www.aclweb.org/anthology/S16-1002>
10. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* 174, 50–59 (2016)
 11. Rennie, J.D., Shih, L., Teevan, J., Karger, D.R., et al.: Tackling the poor assumptions of naive bayes text classifiers. In: *Proceedings of the Twentieth International Conference on Machine Learning*. vol. 3, pp. 616–623. Washington DC) (2003)
 12. Rish, I.: An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. vol. 3, pp. 41–46. IBM New York (2001)
 13. Sangeetha Suresh Harikantra, R.F.: Opinion mining on twitter data. *International Journal of Innovative Research in Science, Engineering and Technology* 5(9), 205–209 (2016)
 14. Yadav, V.: thecerealkiller at semeval-2016 task 4: Deep learning based system for classifying sentiment of tweets on two point scale. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pp. 100–102. Association for Computational Linguistics, San Diego, California (June 2016)