Clasificador de Películas

Lester Andrés García Aquino
Carnet: 1003115
Universidad Rafael Landívar
Guatemala

Oskar Majus De Paz
Carnet: 1034711
Universidad Rafael Landívar
Guatemala

Max Fernando Díaz
Carnet: 1145916
Universidad Rafael Landívar
Guatemala

Ángel Javier Jiménez Morales
Carnet: 1032517
Universidad Rafael Landívar
Guatemala

Walter Gerardo Acabal Matias
Carnet: 1152418
Universidad Rafael Landívar
Guatemala

ABSTRACT

En este proyecto, presentamos el desarrollo e implementación de un sistema de clasificación de películas basado en Naive Bayes, utilizando reseñas del sitio web Rotten Tomatoes. Nuestro sistema aprovecha las bases probabilísticas del clasificador Naive Bayes para categorizar las reseñas de películas como positivas o negativas. La implementación se realizó utilizando Python para el modelo de aprendizaje automático y Spring Boot para los servicios backend y fronted basados en Java, con Flask facilitando la integración de ambos. El preprocesamiento de datos incluyó la limpieza de texto y la tokenización. El modelo de Naive Bayes entrenado demostró una alta precisión y eficiencia en la clasificación de reseñas de películas. Nuestros resultados resaltan la efectividad de Naive Bayes para tareas de procesamiento de lenguaje natural y muestran la interoperabilidad fluida entre las tecnologías de Python y Java. Este trabajo subraya el potencial para el despliegue escalable y robusto de modelos de aprendizaje automático en aplicaciones web.

CONCEPTOS CCS

• Clasificación de la información • Algoritmos y métodos: Aprendizaje Automático, Teorema Bayes • Lenguajes de Programación: Python, Java, • Bibliotecas de Software: Flask, Pandas. • Tecnologías: SpringBoot

PALABRAS CLAVE

Clasificador Naive Bayes, Teorema de Bayes, Python, Java, SpringBoot

1 INTRODUCCIÓN

La clasificación de películas es una tarea importante en la industria cinematográfica y en los sistemas de recomendación de películas. La clasificación de películas puede utilizarse para mejorar la búsqueda de películas, la recomendación de películas y la organización de colecciones de película.
Existen diversos métodos para la clasificación de películas, incluyendo métodos basados en reglas, métodos basados en aprendizaje automático y métodos basados en redes neuronales. Los métodos basados en el teorema de Bayes son un tipo de método de aprendizaje automático que se ha utilizado con éxito para la clasificación de películas.
Este estudio describe la implementación de un clasificador de películas utilizando el Teorema de Bayes, con una arquitectura que combina Spring Boot para el desarrollo del sistema web y la librería de Python Flask para ejecutar el Teorema de Bayes dentro del lenguaje de Java.

2 FUNDAMENTOS

2.1 Teorema de Bayes

El Teorema de Bayes, también conocido como la regla de bayes, es un principio fundamental en la teoría de la probabilidad y la estadística. Nombrado en honor a Tomas Bayes, este teorema describe la probabilidad de un evento basado en el conocimiento previo de las condiciones que podrían estar relacionadas con el evento.

El Teorema de Bayes proporciona una forma matemática de actualizar las probabilidades de hipótesis, dada la evidencia. La fórmula del teorema de bayes es la siguiente:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Donde:

- **P(A|B)** es la probabilidad posterior de la hipótesis **A** dado el dato **B.**

- **P(B|A)** es la probabilidad de observar el dato **B** dado que la hipótesis **A** es verdadera.
- **P(A)** es la probabilidad priora de la hipótesis **A.**
- **P(B)** es la probabilidad total de observar el dato **(B)**

El teorema de bayes tiene una amplia gama de aplicaciones y se utiliza en diversos campos, incluyendo la estadística, la informática, la medicina y la ingeniería.

En el contexto del aprendizaje automático, el teorema de Bayes se utiliza a menudo en la inferencia bayesiana y los modelos probabilísticos. Permite actualizar nuestras creencias sobre una hipótesis basándonos en nuevas evidencias.

Aplicaciones del Teorema de Bayes en el aprendizaje automático:
- Clasificador Naive Bayes
- Optimización Bayesiana
- Redes de Creencia Bayesiana

2.1 Clasificador Naive Bayes

En el ámbito del aprendizaje automático y la minería de datos, el clasificador es un algoritmo probabilístico utilizado para tareas de clasificación. Basado en el teorema de Bayes, esté método es popular debido a la simplicidad, eficiencia y efectividad, especialmente con conjuntos de datos de gran tamaño.

El clasificador funciona calculando la probabilidad de que la instancia pertenezca a cada clase y asigna la etiqueta de la clase con mayor probabilidad.

Pasos del algoritmo:
1. Entrenar el modelo:
    - Se recibe un conjunto de datos de entrenamiento con ejemplos ya etiquetados.
    - Para cada clase, se calcula la probabilidad a priori y la probabilidad condicional de cada característica.
2. Clasificar una nueva instancia
    - Se recibe una nueva instancia de datos sin etiqueta.
    - Se calcula la probabilidad posterior de cada clase (P(A|B)) utilizando el teorema de Bayes:
        - $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$
        - Donde P(B) es la probabilidad marginal de la instancia

Aplicaciones:
- Clasificación de correos electrónicos
- Filtrado de documentos.
- Detección de anomalías
- Recomendaciones

Ventajas:
- Facilidad de Implementación.
- Buena Precisión
- Manejo de datos dispersos
- Escalabilidad

Desventajas:
- Suposición de independencia
- Sensibilidad de datos ruidosos
- Problemas con clases desequilibradas.

3 METODOLOGIA

Para realizar este clasificador de películas se tomó la decesión de utilizar el clasificador de Naive Bayes sobre el perceptrón simple, se tomo está decisión ya que los datos eran textuales y categóricos.

- Por lo tanto, Naive Bayes es particularmente efectivo con datos textuales debido a su capacidad para manejar grandes cantidades de palabras en un texto.
- Permite clasificar de manera más efectiva cuando se trata de texto con múltiples características independientes. En el caso de las películas, las palabras en una reseña pueden ser tratadas como características independientes que contribuyen a la probabilidad de una clasificación positiva o negativa.
- Bayes es rápido de entrenar y eficiente, lo que es útil cuando se trabaja con grandes conjuntos de datos.
- Bayes puede funcionar mejor que el perceptrón simple con conjuntos de datos más pequeños, lo cual es útil cuando no se tiene una gran cantidad de críticas para entrenar el modelo.

3.1 Construcción e Implementación del Clasificador

Para construir el clasificador se recolectaron críticas de películas de la página de Rotten Tomatoes, estás críticas fueron descargadas en archivo de formato .CSV. El clasificador fue desarrollado en Python 3 utilizando la librería de Pandas, ya que Python es un lenguaje muy popular para el aprendizaje automático. El clasificador se conectó por la librería Flask a la aplicación creada en el lenguaje de programación Java.

El conjunto de datos de Rotten Tomatoes se carga y limpia, y el modelo naive Bayes se entrena con estos datos limpios. Todo esto se realiza al iniciar el programa.

Pasos por seguir:
- El proceso comienza con la carga del archivo .CSV utilizando la librería de pandas.
- Luego se limpian los datos del archivo:

- Se reemplazan todas las comas (,) en los campos de texto por punto y coma (;)
- Se eliminan las comillas dobles (") y espacios en blanco ( ) adicionales.
- Se normaliza las fechas al formato 'YYY-MM-DD'.

- Terminada la limpieza se procede a sobrescribir el archivo original con los datos limpios y se guarda una copia en la carpeta 'src/main/resources'.
- Convertir una cadena de texto en una lista de palabras en minúsculas para facilitar su análisis.
- Se debe filtrar las reseñas en positivas y negativas, contando las palabras en ambas categorías y calcular las probabilidades a priori de encontrar una reseña positiva o negativa.
    - Crear un modelo que almacena los conteos y probabilidades en un diccionario.
- Para clasificar un texto, se convierte el texto de entrada en una lista de palabras, se calculan las probabilidades logarítmicas para cada palabra según el modelo entrenado y se determina si el texto es "Fresh" o "Rotten" en función de cuál probabilidad es mayor.

Como se menciona anteriormente para la construcción del modelo se utilizó Python 3 utilizando las librerías Pandas y Numpy, se importa la librería Flask para crear una API Restful que permite la comunicación entre el modelo de Python y el backend de Spring Boot en Java

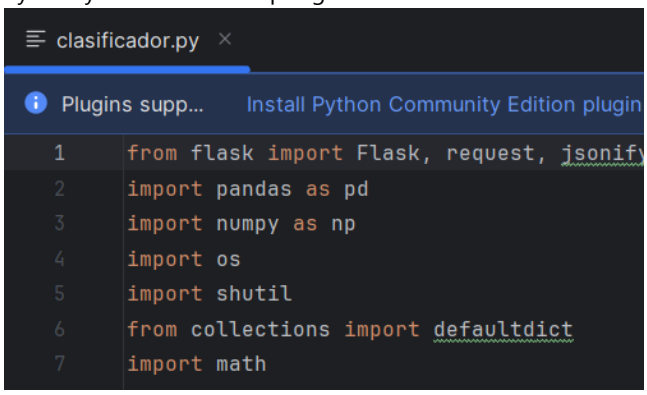


Figura 1: Importación de librerias en Python para la creación del clasificador

Spring Boot maneja el fronted y las solicitudes los usuarios, invocando la API de Flask para obtener las predicciones

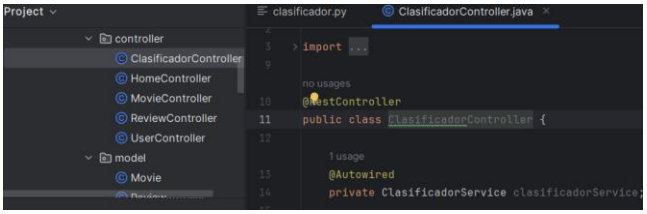
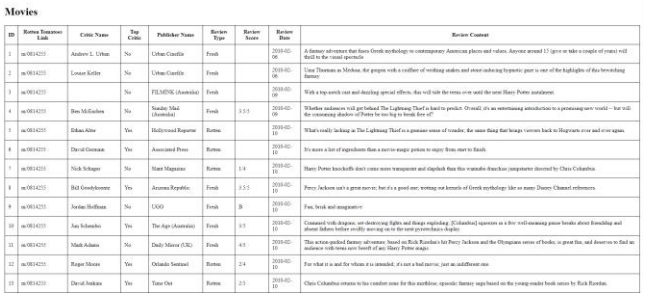


Figura 2: Arquitectura MVC desarrollada en Java para obtener la API de Python

Una vez invocada la API, se muestra el listado de las películas que has sido clasificadas


Movies

ID	Rotten Tomatoes Link	Critic Name	Top Critic	Publisher Name	Review Type	Review Score	Review Date	Review Content
1	m/0814255	Andrew L. Urban	No	Urban Cinefile	Fresh		2010-02-06	A fantasy adventure that fuses Greek mythology to contemporary American places and values. Anyone around 13 (give or take a couple of years) will thrill to the visual spectacle
2	m/0814255	Louise Keller	No	Urban Cinefile	Fresh		2010-02-06	Uma Thurman as Medusa, the gorgon with a coiffure of writhing snakes and stone-inducing hypnotic gaze is one of the highlights of this bewitching fantasy
3	m/0814255		No	FILMINK (Australia)	Fresh		2010-02-09	With a top-notch cast and dazzling special effects, this will tide the teens over until the next Harry Potter instalment
4	m/0814255	Ben McEachen	No	Sunday Mail (Australia)	Fresh	3.5/5	2010-02-09	Whether audiences will get behind The Lightning Thief is hard to predict. Overall, it's an entertaining introduction to a promising new world – but will the consuming shadow of Potter be too big to break free of?
5	m/0814255	Ethan Alter	Yes	Hollywood Reporter	Rotten		2010-02-10	What's really lacking in The Lightning Thief is a genuine sense of wonder, the same thing that brings viewers back to Hogwarts year and year again.
6	m/0814255	David Germain	Yes	Associated Press	Rotten		2010-02-10	It's more a list of ingredients than a movie-magic potion to enjoy from start to finish.
7	m/0814255	Nick Schager	No	Slant Magazine	Rotten	1/4	2010-02-10	Harry Potter knockoffs don't come more transparent and slapdash than this wannabe-franchise juggernaut directed by Chris Columbus.
8	m/0814255	Bill Goodykoontz	Yes	Arizona Republic	Fresh	3.5/5	2010-02-10	Percy Jackson isn't a great movie, but it's a good one, trotting out kernels of Greek mythology like so many Disney Channel references.
9	m/0814255	Jordan Hoffman	No	UGO	Fresh	B	2010-02-10	Fun, brisk and imaginative
10	m/0814255	Jen Schaefer	Yes	The Age (Australia)	Fresh	3/5	2010-02-10	Crammed with dragons, set-destroying fights and things exploding. [Columbus] squeezes us a few well-meaning pause breaks about friendship and absent fathers before swiftly moving on to the next preordained chapter.
11	m/0814255	Mark Adams	No	Daily Mirror (UK)	Fresh	4/5	2010-02-10	This action-packed fantasy adventure, based on Rick Riordan's hit Percy Jackson and the Olympians series of books, is great fun, and deserves to find an audience with teens now bereft of any Harry Potter magic.
12	m/0814255	Roger Moore	Yes	Orlando Sentinel	Rotten	2/4	2010-02-10	For what it is and for whom it is intended, it's not a bad movie, just an indifferent one
13	m/0814255	David Jenkins	Yes	Time Out	Rotten	2/5	2010-02-10	Chris Columbus returns to his comfort zone for this mirthless, episodic fantasy saga based on the young-reader book series by Rick Riordan.

Figura 3: Listado de películas que han sido cargadas al modelo de bayes para su entrenamiento

3.3 Entrenamiento del modelo

Al momento de entrenar modelos de aprendizaje automático, es común dividir los datos en conjuntos de entrenamiento y prueba. Esta división es crucial para evaluar de manera efectiva el desempeño del modelo en datos no vistos, simulando como se comportaría en situaciones reales fuera del ambiente de entrenamiento.

Para el entrenamiento del modelo de este clasificador se inicio a realizar pruebas con un archivo pequeño que contenía 25 registros que equivale al 20% total de los registros, esto se realizó para optimizar recursos al inicio e ir observando, si realmente estaba aprendiendo patrones y no estaba solo memorizando los datos. Gradualmente se fue aumentando el porcentaje de registros para el entrenamiento hasta llegar a un total del 70% por las siguientes razones:

Los datos se pueden dividir de la siguiente manera:

1. División Típica:
    - 70% para el entrenamiento:
        - La mayoría de los datos se utiliza para entrenar el modelo. Esto permite que el modelo aprenda lo

más posible sobre las características y patrones presentes en los datos.

- 30% para prueba:
  - Este conjunto de datos se utiliza para probar la capacidad del modelo para generalizar a nuevos datos. No se utiliza durante el entrenamiento y sirve para evaluar la precisión.
- Variaciones:
  - División 80/20 o 60/40 dependiendo de la cantidad total de datos disponibles y de lo complejo que sea el problema
- Validación Cruzada:
  - Para una evaluación más robusta, se puede utilizar la validación cruzada, especialmente en conjuntos de datos más pequeños o variados. Esto implica dividir los datos en k subconjuntos y utilizar cada uno de ellos como un conjunto de prueba en diferentes iteraciones, mientras que el resto actúa como conjunto de entrenamiento.

CONCLUSIONES

- Para la terea de clasificación de películas, el modelo bayesiano, demuestra ser más conveniente que el perceptrón simple, porque proporciona una metodología robusta para la clasificación de texto.
- La combinación de Python y Java mediante Flask y Spring Boot ofrece una solución eficiente y escalable para aplicaciones de clasificación en tiempo real.
- Al reservar un conjunto de datos de prueba que el modelo nunca ha visto durante su entrenamiento, se puede verificar si el modelo simplemente está memorizando los datos o si realmente está aprendiendo patrones generalizables.

REFERENCIAS

[1] Autores: GeekforGeeks. 2024. Naïve Bayes Classfiers. Disponible en: Naive Bayes Classifiers - GeeksforGeeks

[2] GeekforGeeks.(2024). Bayes theorem in machine learning. Disponible en: Bayes Theorem in Machine learning - GeeksforGeeks

[3] IBM. (s.f). Clasificadores Naïve Bayes. IBM. Disponible en: https://www.ibm.com/topics/naive-bayes

[4] Barrios, J. (2020). Inteligencia artificial y aprendizaje automático para todos. Disponible en: https://www.juanbarrios.com/inteligencia-artificial-y-machine-learning-para-todos/

Price:$15.00

Repositorio de GitHub

https://github.com/LesterAGarciaA97/RottenTomatoes_MovieClassifier