Popularidad de Noticias Online

Descripción del problema

Tenemos un problema de clasificación binaria con un dataset en el que nos proporcionan diferentes características sobre una determinada noticia para intentar predecir si la noticia será popular en internet o no. El dataset se compone de 39797 samples recopilados de la página de noticias Mashable, y tiene los atributos y los targets en el mismo archivo. El dataset nos proporciona en la última columna de los datos el número de shares (interacciones/veces compartido) de una noticia en esa página, y ese número es el que vamos a usar para decidir si un artículo es popular o no, pero no lo usaremos como un atributo sino que lo convertiremos en valores booleanos para usarlos como target de la clasificación binaria.

- **X**: Tenemos 60 atributos en el dataset, pero dos de ellos (url y días entre la publicación y la adquisición del dataset) no se usan para predecir y son meramente informativos para poner los datos en contexto y diferencias una noticia de otra, lo que nos deja con 58 atributos de predicción que son características de cada noticia. Hay cuatro tipos de atributos en función de su valor: números enteros, valores entre 0 y 1, booleanos y nominales, aunque los valores nominales son precisamente los que no se usan para predecir. Podemos agrupar los atributos en cuanto a si se basan en las palabras de la noticia, los links de la noticia, el contenido multimedia de la noticia, el día de salida de la noticia, sus etiquetas o a un Procesamiento del Lenguaje Natural de la noticia.

Característica	Basado en	Tipo
Nº de palabras en el título	Palabras	Entero
Nº de palabras en el texto	Palabras	Entero
Longitud media de las palabras	Palabras	Entero
Ratio de palabras no vacías	Palabras	Ratio
Ratio de palabras únicas	Palabras	Ratio
Ratio de palabras únicas no vacías	Palabras	Ratio
Nº de links	Links	Entero
Nº de links a noticias de Mashable	Links	Entero
Nº de shares de links de Mashable (min, max y media)	Links	Entero
Nº de imágenes	Multimedia	Entero
Nº de vídeos	Multimedia	Entero
Día de la semana	Temporal	Nominalˆ
¿Publicado en fin de semana?	Temporal	Booleano
Nº de etiquetas	Etiquetas	Entero
Peor etiqueta (min, max y media de shares)	Etiquetas	Entero
Etiqueta media (min, max y media de shares)	Etiquetas	Entero
Mejor etiqueta (min, max y media de shares)	Etiquetas	Entero
Categoría de la noticia	Etiquetas	Nominalˆ

Característica	Basado en	Tipo
Cercanía a top 5 grupos ALDˆˆ	PLN	Ratio
Subjetividad del título	PLN	Ratio
Subjetividad del texto	PLN	Ratio
Diferencia absoluta entre 0.5 y ratio subjetividad del texto	PLN	Ratio
Polaridad del título	PLN	Ratio
Ratio de palabras positivas y negativas	PLN	Ratio
Ratio de palabras positivas frente a las neutrales	PLN	Ratio
Ratio de palabras negativas frente a las neutrales	PLN	Ratio
Polaridad de las palabras positivas (min, max, media)	PLN	Ratio
Polaridad de las palabras negativas (min, max, media)	PLN	Ratio
Polaridad del texto	PLN	Ratio
Diferencia absoluta entre 0.5 y ratio polaridad del texto	PLN	Ratio

ˆ: Los valores nominales vienen ya convertidos a valores booleanos con one-hot encoding.

ˆˆ: Asignación Latente de Dirichlet

- **Y**: Las etiquetas a predecir son etiquetas binarias donde 0 es que la noticia no es popular y 1 que la noticia sí es popular, marcando este límite en 1400 interacciones. Hemos decidido usar este umbral ya que es el umbral marcado en el paper asociado al dataset. Si el número de shares es menor que 1400 clasificamos a la noticia con un 0 (no es popular), si iguala o supera ese umbral la clasificamos con un 1 (sí es popular) $Y = 0, 1$
- **f**: nuestra función objetivo es aquella que dada un vector $x \in X$ nos da un valor un valor (booleano) $y, y \in Y, Y = \{0, 1\}$

El problema por tanto se podría haber resuelto también usando regresión para predecir el número de shares de una noticia, y convirtiendo ese share en 0 o 1 según el umbral que hemos visto. Hemos preferido hacerlo a través de clasificación binaria ya que supone un planteamiento más sencillo y podemos gestionar mejor los datos predichos que estén muy cerca del umbral.

Correlación de los datos

Para ver si las variables están relacionadas entre sí vamos a calcular la matriz de coeficientes de correlación de Pearson usando la función `corr()` de los dataframes de Pandas. Luego representamos esa matriz con la función heatmap() del módulo seaborn.

Visualización de los datos

Para visualizar los datos usamos la función TSNE del módulo manifold de sklearn. Esta función nos permite reducir la dimensionalidad de nuestro conjunto de datos

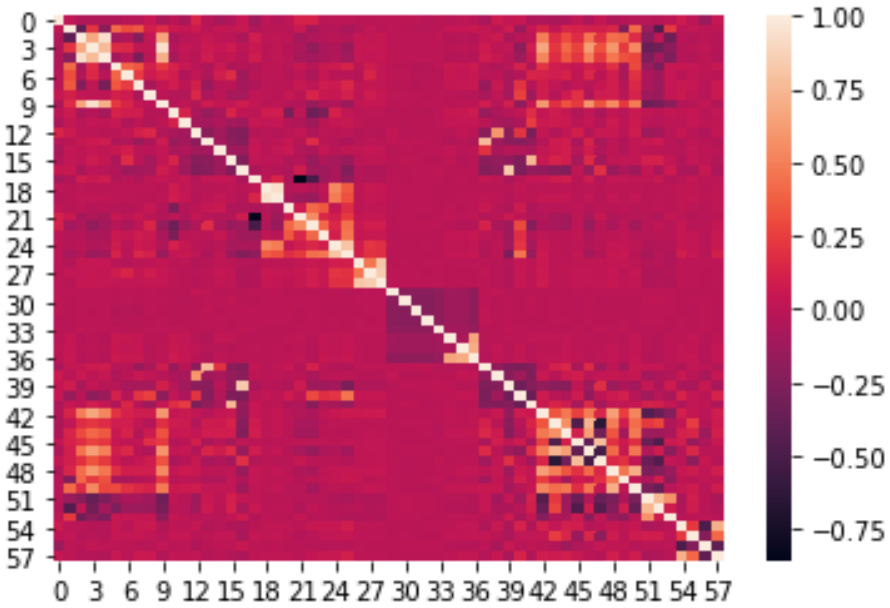

Figure 1: Correlación entre las variables del problema (coef Pearson)

de altas dimensiones a una cantidad de dimensiones que sea representable gráficamente. Con el parámetro n_components indicamos el número de dimensiones al que queremos reducir nuestro conjunto.

Separación de los datos

Preprocesado de los datos

Estandarización

Calculamos la varianza y la media de nuestro conjunto de entrenamiento como la media de la media y la varianza de cada atributo. Obtenemos una varianza de 1210949639.72 y una media de 18237.71. Estos son valores muy altos por lo que la estandarización aquí se nos hace prácticamente obligatoria. Usamos la función StandardScaler() del módulo preprocessing de sklearn, que nos hace una estandarización de los datos, es decir, resta la media y divide por la desviación típica. Después de aplicar estandarizado la varianza es 1 y la media es 0 (aprox).

Outliers