Práctica IV

Detección de clickbait

Primera etapa

Introducción

Clickbait es un fenómeno muy extendido en el mundo online.
noticias

Es una forma de crear titulares y avances destinados a captar la atención de los lectores para aumentar el tráfico.

La función de informar queda relegada a un papel secundario

Introducción

La definición de clickbait para esta práctica es la siguiente: seguir

“Clickbait es un método para generar avances, especialmente en línea, que omite deliberadamente parte de la información con el objetivo de generar curiosidad creando un vacío informativo, atrayendo así la atención de los lectores y haciendo que hagan clic”

Objetivo

Crear un modelo de aprendizaje automático para detectar si un texto es un clickbait o no

Especificación

En equipos de 3-4 miembros hagan lo siguiente:

1) Cargue el corpus TA1C_dataset_detection_train.csv

- La columna de texto teaser se utilizará como función

- La columna Valor de la etiqueta se utilizará como objetivo (clase)

2) Aplicar normalización de texto al valor de la característica

3) Divida el corpus en conjuntos de entrenamiento y desarrollo utilizando el 75 % para entrenamiento y el 25 % para desarrollo.

4) Crea diferentes representaciones de texto. Puedes probar diferentes rangos de n-gramas.

5) Utilice diferentes métodos de aprendizaje automático para entrenar un modelo y predecir instancias del conjunto de desarrollo

6) Realizar los ajustes necesarios al modelo para mejorar el rendimiento sobre el conjunto de desarrollo.

7) Utilice el modelo mejor ajustado para predecir las instancias del conjunto de pruebas disponibles en el
Archivo TA1C_dataset_detection_dev.csv

Normalización de texto

Se recomienda aplicar una normalización diferente.
Técnicas para el valor característico

El proceso de normalización podría incluir algunos de los siguientes pasos: tokenización, limpieza de texto, palabras vacías y lematización.

Es importante probar diferentes combinaciones de los puntos anteriores.
pasos

División del corpus

- Utilice la función train_test_split de scikit-learn
- La instancia del corpus debe barajarse (barajarse = Verdadero)
- Establezca la semilla aleatoria fija en 0 (random_state=0)
- Activar la función estratificar (estratificar=y)

Validación cruzada

Debe utilizar un método de validación cruzada el tren

Establecer una validación cruzada estratificada de cinco pasos

(StratifiedkFold(n_splits=5) o cross_val_score(estimador, X, y, cv=5, puntuación='f1_macro')

Características y representaciones textuales

Podrías probar unigramas, bigramas, trigramas y combinaciones de ellos para extraer características.

Podrías utilizar representaciones de texto binarias, de frecuencia y TF-IDF

También se recomienda explorar el uso de TruncatedSVD para reducir la dimensionalidad y enriquecer las características

Métodos de aprendizaje automático

Pruebe diferentes algoritmos tradicionales de aprendizaje automático (ML)

Podrías utilizar los mismos métodos implementados en el caso anterior.
práctica

Además, puedes probar los métodos de conjunto como:

- Bosque aleatorio (RandomForestClassifier)
- Potenciación de gradiente (GradientBoostingClassifier)

Sería útil ajustar los hiperparámetros del algoritmo para mejorar los resultados.

Métricas de evaluación

Se debe aplicar un informe de clasificación (classification_report) a las predicciones del conjunto de desarrollo para cada variación de:

- Proceso de normalización
- Representación de texto
- Método de aprendizaje automático
- Ajustes de hiperparámetros

- La métrica principal es f1_macro, esta se utilizará para seleccionar la mejor modelo

Predicciones sobre el conjunto de pruebas

El mejor modelo debe ser reentrenado utilizando el corpus completo del archivo TA1C_dataset_detection_train.csv con las configuraciones seleccionadas (normalización, representación de texto, modelo ML, hiperparámetros, balance de clases, etc.)

El modelo reentrenado debe usarse para predecir el valor de etiqueta de las instancias del archivo TA1C_dataset_detection_dev.csv

Predicciones sobre el conjunto de pruebas

El archivo CSV de salida debe tener lo siguiente características:

- Nombre: detección.csv
- Columnas: ID del tweet y valor de la etiqueta
- Carácter separador: Coma ","

desequilibrio de clases

La distribución de clases en el corpus muestra un desequilibrio significativo (71% No clickbait y 29% clickbait)

El desequilibrio de clases a menudo conduce a sesgos en los modelos de aprendizaje.

Sería útil utilizar algunos métodos para equilibrar las clases.
distribución

El submuestreo y el sobremuestreo son dos de los métodos más equilibrados común (
https://imbalanced-learn.org/stable/)

Evidencia

Código fuente

Un informe en formato PDF que describa lo siguiente:

- Tarea a resolver
- Métodos de aprendizaje automático seleccionados
- Hiperparámetros ajustados
- Informe de clasificación de cada experimento

Evidencia

Una tabla que describe los experimentos realizados mostrando los mejores configuración de cada método ML en el conjunto de desarrollo

Método ML	Hiperparámetros de ML	Normalización de texto	Representación de texto	Métodos de equilibrio	Puntuación f promedio macro
Regresión logística	máx_iter = 200	Tokenización, lematización	Unigrama, binarizado	Ninguno	0,85
Bayes ingenuo	por defecto	Tokenización, palabras vacías	Bigrama, frecuencia	Submuestreo	0.88
...	...		...		...
Perceptrón multicapa	hidden_layer_sizes = (200, 100)	Tokenización, limpieza de texto, lematización	Unigrama + bigrama, sobremuestreo Tf-idf		0.9

Evidencia

El archivo detection.csv