Grado en Ingeniería Informática

Minería Web

Curso 2024/2025

Práctica 2: Minería de Contenido de la Web

Universidad de Jaén

1. Introducción

La extracción de información de documentos web (y de documentos en general) representa hoy en día un sector de gran relevancia económica, con un impacto significativo en múltiples ámbitos de nuestra sociedad. Esta práctica, conocida como minería web, ofrece soluciones innovadoras a una amplia gama de desafíos, mejorando nuestra interacción con la información digital de manera profunda y variada.

Entre sus aplicaciones más destacadas se encuentra la capacidad de personalizar recomendaciones de productos, películas, artículos y otros contenidos, basándose en las preferencias y gustos individuales. Esto no solo enriquece la experiencia del usuario si no que también optimiza la eficacia de las plataformas que implementan tales recomendaciones. Además, la minería web facilita la segmentación de audiencias para campañas publicitarias, permitiendo a las empresas dirigir sus esfuerzos de manera más precisa y eficiente. Esta capacidad de segmentación se traduce en campañas más exitosas, al alcanzar a aquellos individuos más susceptibles a la propuesta comercial presentada. La evaluación de productos y servicios mediante el análisis de comentarios y opiniones en la web es otra ventaja significativa. Y un largo etcétera de aplicaciones.

2. Objetivo de la práctica

La finalidad de la práctica es por un lado la familiarización del estudiante con técnicas básicas de minería de datos en general, y en particular con aquellas centradas en la extracción de conocimiento en documentos web. Se pide por tanto que el estudiante sea capaz de aplicar las diferentes técnicas de preprocesamiento de documentos vistas en clase, así como la aplicación de diferentes técnicas de clustering y clasificación también analizadas en clase.

3. Descripción y Estructura de la Práctica

La práctica puede realizarse de manera individual o en grupos de 2 personas máximo.

En esta práctica se trabajará principalmente con **Python** mediante el empleo de la librería *scikit-learn*. No obstante, el estudiante es libre de utilizar otra herramienta de minería de datos si así lo desea. Para el desarrollo de esta práctica, nos apoyaremos en el tutorial *Working With Text Data* (**Nota:** el tutorial es para una versión antigua

de scikit-learn, la 1.4, no obstante, la mayor parte de su funcionalidad debería de funcionar para versiones futuras) de scikit-learn que nos proporciona un tutorial paso a paso de procesamiento de texto. Se recomienda que el estudiante realice el tutorial paso a paso, pero prestando especial atención al uso de la herramienta `Pipeline`. Asimismo, se recomienda que se instale la librería `Pandas` y `numpy` para la gestión sencilla y eficiente de datos. Para instalar todas estas librerias, simplemente ejecuta esta orden en un nuevo entorno virtual de Python:

```
pip install scikit-learn pandas numpy
```

La estructura de la práctica se organiza en tres partes diseñadas para profundizar en la comprensión y aplicación de técnicas avanzadas de procesamiento de textos y análisis de datos:

- **Agrupamiento de Documentos**: Esta sección requiere que los estudiantes apliquen diversas técnicas de agrupamiento discutidas durante el curso para llevar a cabo un estudio experimental. Deberán implementar diferentes algoritmos de clustering sobre un conjunto de documentos, analizar los resultados obtenidos de cada método y, finalmente, elaborar conclusiones fundamentadas sobre la efectividad y las peculiaridades de cada algoritmo utilizado.
- **Clasificación de Documentos**: En este módulo, utilizando el mismo conjunto de documentos, se espera que los estudiantes empleen distintos algoritmos de clasificación estudiados en el curso. Similar al ejercicio de agrupamiento, es esencial que se realice un análisis detallado de los resultados de clasificación, extrayendo conclusiones significativas sobre la precisión, eficacia, y las limitaciones de los métodos aplicados.
- **Competición de Kaggle**: Para estimular la cooperación y el intercambio de conocimientos entre los participantes, se propone un desafío en la plataforma Kaggle. El objetivo es mejorar los modelos de clasificación de documentos desarrollados previamente. Los estudiantes tendrán que experimentar con diversas estrategias de minería de datos y ajuste de parámetros para lograr el mejor desempeño posible, promoviendo así un ambiente de aprendizaje competitivo y colaborativo.

El **preprocesamiento** de los datos se destaca como un componente fundamental en cada fase del proyecto. Se enfatiza la importancia de transformar el texto en *tokens* a través de técnicas como la **eliminación de stopwords, stemming, y otras**

metodologías revisadas en el curso. Además, se anima a los estudiantes a experimentar con distintas representaciones de los datos, tales como la representación **binaria**, por **frecuencia**, o **TF-IDF**, para evaluar cómo cada enfoque afecta la calidad del análisis y el conocimiento extraído de los documentos.

En todos los estudios realizados, se llevará a cabo un estudio con **Validación Cruzada Estratificada de 5 folds (5-SCV)**, revise la documentación de la clase `StratifiedKFold` en scikit-learn para ver como realizarlo. **Se recomienda utilizar un valor de semilla fijo para que el estudio sea reproducible**.

3.1. Datos utilizados

Se utilizará un conjunto de datos de noticias del periódico Huffington Post desde el año 2012 a 2022. El conjunto de datos completo se compone de aproximadamente 200.000 documentos relacionados con 42 categorías diferentes. Debido al gran tamaño del conjunto de datos, para el apartado de agrupamiento y clasificación obligatorio, se trabajará con un conjunto reducido de unos 10.000 documentos (el 5 % del total) de 4 categorías distintas. Sin embargo, para la competición de Kaggle, se trabajará obligatoriamente con el conjunto de datos completo con el objetivo de que suponga un reto.

El conjunto de datos para esta competición se compone de registros que incluyen los siguientes atributos:

- category: categoría en la que se publicó el artículo (**es nuestra clase**).
- headline: el titular del artículo de noticias.
- text: es el texto principal de la noticia.
- authors: lista de autores que contribuyeron al artículo.
- link: enlace al artículo original.
- short_description: resumen del artículo de noticias.
- date: fecha de publicación del artículo.

El alumno es libre de utilizar aquellas variables que considere relevantes de cara a la realización de la práctica, justificando las decisiones llevadas a cabo.

3.2. Agrupamiento

Se deben realizar diferentes experimentos de agrupamiento sobre la versión reducida del conjunto de datos. Estos experimentos consistirán en aplicar el algoritmo *k-means* sobre diferentes representaciones del texto: binaria, frecuencia y TF-IDF. Se debe ignorar el atributo de clase. Utiliza el valor K=4. Para cada representación utilizada, contestar a las siguientes cuestiones:

1. Utiliza diferentes semillas de números aleatorios para que los centroides iniciales cambien de lugar. Observa como cambian los resultados. ¿Por qué el algoritmo *k-means* es tan sensible a los cambios de configuración iniciales?
2. Busca el agrupamiento más adecuado y almacena las asignaciones de los grupos en un fichero. Esta asignación nos servirá para realizar la validación externa posterior.
3. Visualiza los clusters obtenidos. Apóyate en el algoritmo t-SNE para esta tarea.
4. Realiza un análilsis del clustering obtenido, haz tanto una evaluación interna (con métricas de cohesión y separación), como una evaluación externa (usando los grupos calculados anteriormente y las clases). Analiza los resultados y extrae conclusiones sobre los mismos.
5. Ejecuta, solo para la representación TF-IDF, el algoritmo de mezcla de gaussianas, el cual implementa el algoritmo EM (clase `GaussianMixture` en scikit-learn) usando 4 componentes de mezcla (`n_components=4`). Compara el resultado obtenido con k-NN.

3.3. Clasificación

Al igual que en el apartado anterior, se trabajará sobre la versión reducida del conjunto de datos. En este caso, el trabajo a realizar consistirá en aplicar el algoritmo *k-NN* (`KNeighborsClassifier`) y *Naïve Bayes*. Del mismo modo, se aplicarán estos algoritmos con las diferentes representaciones (binaria, frecuencia y TF-IDF) siempre que sea posible.

Contestar a las siguientes cuestiones:

- Respecto a k-NN, para cada representación utilizada:

    - Prueba con diferentes valores de `k`, esquemas de pesos y valor `p` que indica la potencia de la distancia de Minkowski. Se recomienda analizar al menos 5 combinaciones diferentes. Comenta los resultados obtenidos y analízalos. Identifica los parámetros que han obtenido la máxima precisión.

- Respecto a Naïve Bayes, para cada representación utilizada:

    - Ejecuta la versión Gaussiana (`GaussianNB`) y la versión Multinomial (`MultinomialNB`) y compara su rendimiento. Comenta los resultados obtenidos.

- Finalmente, compara los resultados obtenidos por k-NN y Naïve Bayes y determina cual es el mejor algoritmo y la mejor representación atendiendo a la precisión obtenida. Considera también otros aspectos como el tiempo de ejecución o el consumo de memoria (entre otros) para enriquecer tu análisis.

3.3.1. Competición de Kaggle

Kaggle es una plataforma de competiciones de ciencia de datos muy conocida que se caracteriza por sus competicion con premios millonarios creados por diferentes empresas. En esta práctica se trabajará sobre una competición privada en donde se usa el conjunto de datos completo en un problema de clasficación, lo que supone todo un reto. La información adicional y los detalles de la competición se establecen dentro de la competición creada en Kaggle para tal efecto. El alumnado es completamente libre de utilizar todas las estrategias y métodos que conozca para maximizar el rendimiento de la clasificación, teniendo que explicar cada paso llevado a cabo. Se anima a que todo el mundo participe y se recuerda que es un problema muy difícil que puede llevar a mucha frustración. Se recuerda que el objetivo final de este apartado es el de aprender nuevas técnicas y favorecer la cooperación y colaboración, por lo que no importa que el resultado no sea muy bueno.

Al final de la competición el ganador recibirá como premio **0.75 puntos** extra sobre la nota final, el segundo clasificado **0.5 puntos** y el tercero **0.25 puntos**, siempre y cuando obtengan mejoras respecto a lo realizado en los apartados anteriores. **Los tres ganadores deberán exponer en una presentación de unos 10 minutos aproximadamente su desarrollo.**

El enlace de la competición es este: Competición de Kaggle.

4. Documentación y entrega

Se enviará un fichero ZIP que contenga lo siguiente:

- Memoria en formato PDF con los pasos llevados a cabo en cada tarea, las tablas de resultados y el análisis realizado.
- Si se usa Python, código fuente en Python de los estudios experimentales realizados.

5. Envío

La fecha tope de entrega será el día **22 de abril a las 23:59** en la tarea de PLATEA habilitada a tal efecto.