Minería de datos: preprocesamiento y clasificación

3 de diciembre de 2024

UNIVERSIDAD DE GRANADA

Práctica para la evaluación de la asignatura

Máster en Ciencia de Datos e Ingeniería de Computadores, Curso 2024-2025

Índice

---

1. Objetivos y evaluación

En esta práctica se deberán utilizar los métodos de preprocesamiento y aprendizaje vistos en la asignatura **Minería de datos: preprocesamiento y clasificación** para resolver un problema concreto.

El estudiante debe adquirir destrezas para mejorar los datos con el objetivo de obtener los mejores resultados posibles explorando diferentes algoritmos de aprendizaje (y sus posibles parametrizaciones) para un problema concreto. En el guion se plantean una serie de conjuntos de datos, y se asignará a cada grupo un conjunto distinto sobre el que aplicar el preprocesamiento y los distintos algoritmos. No hay, por tanto, ninguna competición entre equipos, ya que cada uno abordará un problema distinto.

La evaluación de la práctica se hará en función de varios criterios, como el desarrollo del trabajo realizado, la coordinación del equipo, la exposición del trabajo realizado, la diversidad de técnicas utilizadas, etc. La documentación que debe entregar cada equipo, consistente en una presentación que debe cumplir con un mínimo de especificaciones indicadas en la Sección 4, debe detallar de forma clara el proceso de trabajo seguido desde el inicio hasta la entrega de los resultados finales.

2. Composición de los equipos

Los equipos se componen de 5 miembros. La composición de los equipos se realizará a lo largo del curso, junto con el nombre del mismo. Los datos mencionados se deberán indicar en PRADO junto al coordinador del equipo, que centralizará la comunicación con los profesores.

Es previsible que, para el inicio de la práctica, los equipos ya estén formados. Aquellos alumnos que no tengan equipo, serán asignados por los profesores a un nuevo equipo hasta completar 5 integrantes.

3. Normativa y problemas

Para construir modelos de cada uno de los conjuntos de datos a utilizar, nos centraremos en el uso de cinco métodos de clasificación básicos: (1) kNN y derivados, (2) árbol de clasificación C4.5/CART/cualquier árbol, (3) SVM, (4) regresión logística con cualquier complejidad y/o regularización y (5) Naïve Bayes. Además, también se utilizarán otros cinco métodos basados en ensembles: (1) Voting, (2) Bagging, (3) AdaBoost, (4) Gradient Boosting y (5) Stacking. Es responsabilidad del equipo el reparto de cada algoritmo de

clasificación entre sus integrantes. Cada integrante se encargará de escoger un método de clasificación básico y un método de ensemble. Éste último podrá usar cualquier algoritmo de aprendizaje por debajo (por ejemplo, un Bagging de SVMs o un Voting de varios tipos de modelos), independientemente del algoritmo básico que le haya correspondido al integrante. Como se ha indicado, además del algoritmo básico, cada miembro del equipo deberá elegir un algoritmo de *ensemble* entre los siguientes:

- Bagging (Random Forest), ExtraTrees, Random Subspaces,
- AdaBoost,
- Gradient Boosting (XGBoost, LightGBM, CatBoost, HistGradientBoosting),
- Voting,
- Stacking.

En total, cada miembro del equipo deberá abordar un algoritmo de clasificación *básico* y un algoritmo de *ensemble* de los arriba mencionados.

De forma integral, el equipo deberá aplicar las técnicas de preprocesamiento (codificación, detección de outliers, discretización, tratamiento de valores perdidos, eliminación de instancias con ruido, selección de características, reducción de dimensionalidad...) que se consideren oportunas para cada una de las técnicas de aprendizaje indicadas. Es posible que parte del preprocesamiento se aplique de forma común para todas las técnicas y, a posteriori, cada miembro haga preprocesamientos especializados para sus técnicas de aprendizaje.

El trabajo de los componentes del equipo consistirá en que cada uno de ellos se ocupe de sus técnicas de aprendizaje elegidas, así como del procesamiento necesario; poniendo luego en común entre todos los resultados obtenidos y ofreciendo posibles soluciones o ideas al resto de componentes del equipo.

Se proporcionará un conjunto etiquetado, que se asignará por sorteo a cada equipo, que representará un problema de clasificación o de regresión. Dependiendo del problema, es posible que el conjunto ya esté particionado en *training* y *test*. **En el caso en el que no exista partición de test, el equipo deberá crear una partición de test antes de aplicar cualquier técnica**.

El equipo trabajará **exclusivamente sobre la partición inicial de training** (proporcionada o creada para generar la partición de test). El ajuste de hiperparámetros de los métodos de clasificación, la elección de los preprocesamientos y el orden de ejecución de estos últimos deberá decidirse exclusivamente con el conjunto de *training* utilizando los procedimientos de validados apropiados. Para esta tarea, se recomienda el trabajo en local (en vuestras máquinas) frente al uso de Google Colab o similares.

Una vez decididos los mejores ajustes de los modelos de aprendizaje y preprocesamientos, se evaluarán las instancias de la partición de test (con el método `predict()` correspondiente) para obtener los valores de las métricas de rendimiento que se explicarán más adelante.

IMPORTANTE: En ningún caso se podrán utilizar los resultados de la partición de test para optimizar los modelos de aprendizaje o preprocesamiento, tal y como se muestra en la Figura 1.

El trabajo debe basarse en la repetición del siguiente procedimiento las veces que consideréis oportunas:

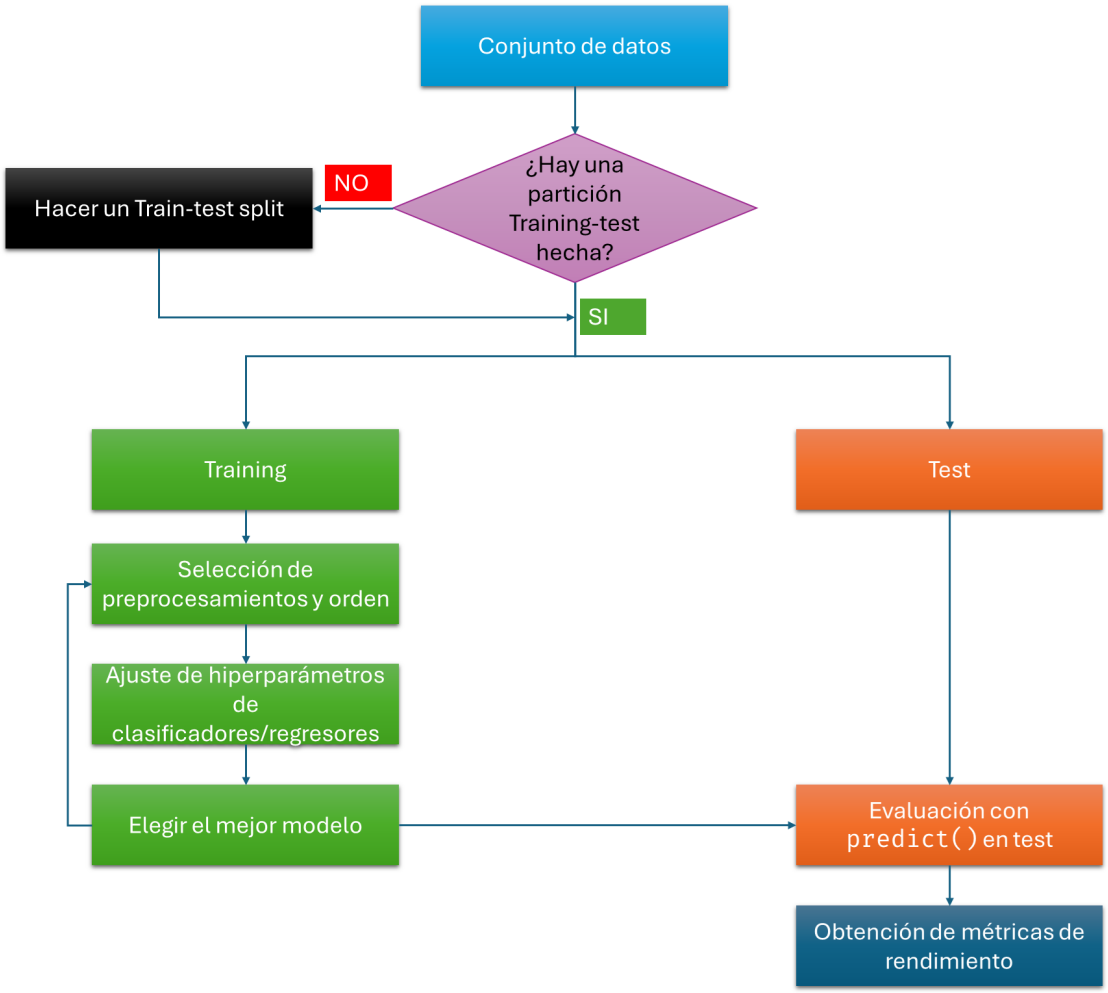

Figura 1: Flujo de trabajo con las particiones para la práctica.

1. Preprocesar y aprender algún modelo sobre el conjunto de entrenamiento, usando validación cruzada, para ajustar los hiperparámetros de forma manual o automática.

2. Usando este modelo, predecir la/s etiqueta/s o el valor numérico correspondiente (según sea un problema de clasificación o de regresión) de cada una de las instancias del conjunto de test.

La evaluación se hará sobre al menos dos métricas distintas. Para los problemas de **clasificación** siempre se utilizará la tasa de los ejemplos correctamente clasificados (*accuracy*) junto a otra métrica a elección por el equipo que encaje bien con las propiedades del problema (por ejemplo, F-measure, AUC, Kappa, macro-average, ...). Para los problemas de **regresión** se utilizará la raíz del error cuadrático medio o *Root Mean Square Error* (*RMSE*) y otra a elección del equipo (por ejemplo, *MAE*, R^2, *Hinge loss*, ...).

Para la realización de esta actividad estará permitido el uso de dos lenguajes de programación y sus bibliotecas o paquetes: **R** y **Python**. Ambos están permitidos para cualquier tipo de tarea, usando cualquier paquete R (visto en clase o no) o bibliotecas (por ejemplo *Scikit-Learn*, *NumPy*), *Pandas* o cualquier módulo *third-party* que se encuentre en un repositorio público. Se recomienda la utilización de **Python** para preprocesamientos simples y formateado de datos, y **R** para preprocesamientos de datos más

complejos. Los clasificadores que se pueden usar pueden ser de **R** ó **Python**, cualquier paquete de CRAN o módulo en **Python**; siempre y cuando se ciñan al tipo de algoritmo de aprendizaje concreto (árbol, bayes, SVM, etc.)

Los conjuntos de datos propuestos pueden contener valores categóricos (nominales) y numéricos en sus atributos, así como en el atributo de salida. Dos de los algoritmos de clasificación básicos propuestos trabajan con valores categóricos de forma natural, mientras que los algoritmos kNN, regresión logística y SVM requieren del uso de métricas de distancia concretas para la gestión de valores categóricos o técnicas de codificación. Así mismo, los algoritmos basados en ensembles dependen de los algoritmos base que utilizan, aunque algunos ensembles concretos de árboles, como XGBoost o LightGBM, utilizan codificaciones simples de datos categóricos y otros profundizan en el tratamiento mixto de datos nominales y numéricos (como CatBoost).

A continuación, se listan los problemas y conjuntos de datos asociados a abordar. **Los conjuntos podrán ser descargados desde las respectivas webs originales de los autores que se proporcionan para cada datasets, respetando así los derechos de cada conjunto.**

3.1. Problemas y conjuntos de datos

1. Título: DengAI: Predicción de casos de dengue

   - Descripción: El objetivo es predecir el número de casos de dengue en dos ciudades de Puerto Rico y Perú en base a información medioambiental adquirida por agencias gubernamentales de Estados Unidos. Se dispone de 1457 ejemplos de entrenamiento con datos faltantes y 24 variables de naturaleza mixta. Dichas variables incluyen información sobre la ciudad y la fecha en que se tomaron las medidas, y datos relativos a la temperatura, las precipitaciones, la humedad, la vegetación, etc.
   - Referencias: https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/80/, https://link.springer.com/book/10.1007/978-3-319-10247-4

2. Título: Predicción del nivel de daños en edificios provocados por un terremoto

   - Descripción: El objetivo de este proyecto es estimar el nivel de destrucción de edificios provocado por el terremoto Gorkha (que tuvo lugar en Nepal en 2015). Se trata de un problema de predicción ordinal, con tres grados de daño (bajo, medio, completo). El dataset incluye 38 características mixtas (con información relativa al número de pisos del edificio, los años del mismo, la altura, el área, el tipo de cimientos y tejado, etc.) y 260.601 ejemplos de entrenamiento.
   - Referencias: https://www.drivendata.org/competitions/57/nepal-earthquake/, https://orca.cardiff.ac.uk/id/eprint/112802/, https://github.com/ayrna/orca

3. Título: Predicción de bombas de agua defectuosas

   - Descripción: A partir de datos del Ministerio del Agua de Tanzania, el objetivo es predecir qué bombas funcionan, cuáles necesitan algunas reparaciones y cuáles no funcionan en absoluto. Se pretende predecir una de estas tres clases basándose en 39 variables mixtas (incluyendo localización geográfica, población en torno al pozo, organización que lo instaló, precio del agua, etc.) y

59.400 ejemplos de entrenamiento. Una mejor comprensión de qué puntos de agua fallarán puede mejorar las operaciones de mantenimiento y garantizar que haya agua limpia y potable disponible para las comunidades de Tanzania.

- Referencias: https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/, https://link.springer.com/book/10.1007/978-3-319-10247-4

4. Título: Flu Shot Learning: predicción relativa a las vacunas contra la gripe H1N1 y la gripe estacional

   - Descripción: En este proyecto nos centramos en predecir si las personas recibieron la vacuna contra la gripe H1N1 y la gripe estacional utilizando la información que compartieron sobre sus antecedentes, opiniones y comportamientos de salud. En concreto, debemos predecir dos variables binarias (si el encuestado recibió la vacuna contra la gripe H1N1 y si el encuestado recibió la vacuna contra la gripe estacional) a partir de 35 variables relativas al nivel de preocupación de la persona encuestada en relación a la gripe, si ha comprado una mascarilla, si es trabajador/a del sector sanitario, grupo de edad, etc. Se dispone, en total, de 26.706 ejemplos de entrenamiento.
   - Referencias: https://www.drivendata.org/competitions/66/flu-shot-learning/, https://link.springer.com/book/10.1007/978-3-319-10247-4

5. Título: Monitorización de actividad doméstica de ancianos que viven solos

   - Descripción: Este conjunto de datos se creó para construir un modelo que permitiera monitorizar, a partir de la información recopilada por una serie de sensores, la actividad de una persona de avanzada edad que viva sola en su casa. Se utilizan sensores químicos por ser menos invasivos y por respetar más la privacidad. La información proporcionada por estos sensores, junto con los datos de control de los sensores de movimiento, se intentan utilizar como indicadores para detección de actividades anómalas. En concreto, según indica la documentación, se dispone de 10 variables de entrada y 444.631 ejemplos.
   - Referencias: https://archive.ics.uci.edu/dataset/799/single+elder+home+monitoring+gas+and+position y Marín, Daniel, et al. “Home monitoring for older singles: A gas sensor array system."Sensors and Actuators B: Chemical 393 (2023): 134036.

6. Título: Clasificación de setas

   - Descripción: Este conjunto de datos incluye 61.069 ejemplos de setas con sombreros basados en 173 especies (353 setas por especie). Cada seta se identifica como definitivamente comestible, definitivamente venenosa o de comestibilidad desconocida y no recomendada (esta última clase se combinó con la clase venenosa). Se emplean 20 características de naturaleza heterogénea para su clasificación y presentando datos faltantes.
   - Referencias: https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset y Wagner, Dennis, Dominik Heider, and Georges Hattab. “Mushroom data creation, curation, and simulation to support classification tasks."Scientific reports 11.1 (2021): 8134.

7. Título: Mantenimiento automático de una línea de metro: detección temprana de fallos en la maquinaria

- Descripción: De un metro operativo, se recogieron lecturas de presión, temperatura, corriente del motor y válvulas de entrada de aire de la Unidad de Producción de Aire (APU) de un compresor. Se dispone de 1.516.948 instancias y 15 características. El objetivo sería realizar un análisis del sistema, así como detectar las anomalías o fallos del mismo. Se trata de un problema complejo, en donde el componente temporal puede ser considerado o ignorado por parte de los estudiantes. De hecho, este conjunto de datos representa e incluye algunos de los desafíos reales que surgen a la hora de realizar el mantenimiento predictivo de tareas industriales.
- Referencias: https://archive.ics.uci.edu/dataset/791/metropt+3+dataset y Davari, Narjes, et al. "Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry."2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2021.

8. Título: Predicción de la popularidad de noticias online

   - Descripción: Este problema incluye un conjunto heterogéneo de características sobre los artículos publicados en www.mashable.com en un período de dos años. El objetivo es predecir el número de *shares* en las redes sociales (popularidad) a partir de 58 características. El conjunto de entrenamiento consta de 39.644 ejemplos. Uno de los principales desafíos de este proyecto consiste en abordar adecuadamente la naturaleza desbalanceada de los datos. En concreto, en este problema de regresión, las salidas deseadas siguen una distribución *power law* con una larga cola a la derecha.
   - Referencias: https://archive.ics.uci.edu/dataset/332/online+news+popularity y Fernandes, K., Vinagre, P., & Cortez, P. (2015). "A proactive intelligent decision support system for predicting the popularity of online news". In Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015 (pp. 535-546); https://github.com/paobranco/ImbalancedLearningRegression, https://link.springer.com/book/10.1007/978-3-319-98074-4, https://link.springer.com/article/10.1007/s10462-024-10724-3

9. Título: Detección de Fraude en Transacciones Financieras

   - Descripción: Este proyecto implica detectar transacciones financieras fraudulentas utilizando características como el monto de la transacción, el país del titular de la tarjeta, el tipo de tarjeta, etc. El conjunto de datos contiene 284.807 ejemplos de transacciones con 31 características. No hay datos faltantes pero sí un enorme desbalanceo en términos de la proporción de transacciones fraudulentas y legítimas (solo un 0,172 % del total son fraudulentas). Un problema similar, pero más complejo, es el representado por el *Bank Account Fraud (BAF) dataset* con 6 millones de instancias y 31 características. Si el alumnado prefiere abordar este último puede hacerlo y la evaluación del proyecto tendrá en cuenta la complejidad inicial del mismo.
   - Referencias: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud y Dal Pozzolo, Andrea "Adaptive Machine learning for credit card fraud detection" ULB MLG PhD thesis (supervised by G. Bontempi) (2015). Para el segundo problema: https://github.com/feedzai/bank-account-fraud

y Jesus, Sérgio, et al. "Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation." Advances in Neural Information Processing Systems 35 (2022): 33563-33575, https://link.springer.com/book/10.1007/978-3-319-98074-4, https://link.springer.com/article/10.1007/s10462-024-10724-3, https://github.com/paobranco/ImbalancedLearningRegression

10. Título: Predicción de retrasos en vuelos

    - Descripción: Problema de regresión en donde el objetivo es predecir el retraso de los vuelos a partir de información básica sobre los mismos. Según la propia documentación del conjunto de datos, no hay datos faltantes pero sí valores atípicos, y el volumen de los datos es muy notable: 5.929.413 instancias con 8 características (distancia entre aeropuertos, mes, día del mes, día de la semana, antigüedad del avión, y duración del vuelo, entre otras).
    - Referencias: https://www.kaggle.com/datasets/lcsldatasets/flights; Hensman, James, Nicolas Durrande, and Arno Solin. "Variational Fourier features for Gaussian processes." Journal of Machine Learning Research 18.151 (2018): 1-52; y Meanti, Giacomo, et al. "Kernel methods through the roof: handling billions of points efficiently." Advances in Neural Information Processing Systems 33 (2020): 14410-14422, https://link.springer.com/book/10.1007/978-3-319-47578-3

11. Título: Predicción multi-etiqueta del género de una película a partir de la trama de la misma

    - Descripción: Contiene 120.919 resúmenes de textos de tramas de películas de Internet Movie Database (www.imdb.com), etiquetados con uno o más de entre los 28 géneros posibles. En total, se dispone de 1001 características binarias.
    - Referencias: "Imdb" dataset en http://www.uco.es/kdis/mllresources/. Read, Jesse. "Scalable multi-label classification". PhD Diss. University of Waikato, 2010; y Read, Jesse, et al. "Classifier chains for multi-label classification." European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD) 2009. Si la lectura de los ficheros ".arff" resulta complicada, se recomienda revisar la documentación técnica disponible al respecto, como por ejemplo https://stackoverflow.com/questions/59271661/cannot-load-arff-dataset-with-scipy-arff-loadarff, https://link.springer.com/book/10.1007/978-3-319-41111-8, http://scikit.ml/

12. Título: ECDBL'14 CM prediction

    - Descripción: La predicción de mapas de contacto (MC) es una tarea de clasificación bioinformática (y, en concreto, de predicción de estructuras de proteínas) que constituye un caso de prueba ideal para un reto de big data por varias razones. Los conjuntos de datos de MC alcanzan fácilmente decenas de millones de instancias, cientos (si no miles) de atributos y presentan un desequilibrio de clases extremadamente alto. El conjunto puede descargarse aquí usando la cuenta GO-UGR: https://drive.google.com/file/d/1QqcWCV9JT6MNrLV_FyCZ7Y5Rk0JLsRDB/view?usp=sharing
    - Referencias: https://academic.oup.com/bioinformatics/article/28/19/2441/289358.

4. Entrega

La actividad finaliza el día 26 de Enero del 2025, a las 23:59 horas. En ese momento, se debe entregar toda la documentación requerida donde debéis explicar el trabajo realizado por cada miembro del equipo y las colaboraciones entre vosotros. Además, deberá de añadirse un video en la que los distintos miembros del equipo exponen dicha presentación (cada uno/a su parte correspondiente del trabajo, obviamente).

Un único miembro del equipo (no importa cuál) deberá entregar tres archivos en la plataforma **PRADO**, en la parte correspondiente a la asignatura.

1. El primer fichero será un comprimido que contendrá el proyecto software con todos los scripts/notebooks de **R** y **Python** generados durante el trabajo en la práctica, suficientemente explicado como para poder ejecutarlo y obtener los resultados incluidos en la memoria (a partir de los datos de partida).

2. El segundo documento, que corresponde a las diapositivas con la presentación del trabajo (en formato **pdf** exclusivamente), describe el trabajo llevado a cabo, indicando de forma clara el trabajo realizado por cada componente del equipo y el trabajo común realizado (preprocesamiento necesario para todas las técnicas, aportación de ideas, etc).

3. El tercer archivo consistirá en un vídeo de presentación de **25 minutos en total** donde cada integrante expondrá su trabajo particular con los algoritmos que haya trabajado y el equipo también introducirá el problema y el preprocesamiento realizado de forma común **usando las transparencias del punto anterior**.

La documentación del punto 2 anteriormente mencionado constará de una presentación de unas 7 diapositivas como máximo por miembro del equipo de clasificación. El equipo deberá añadir 3 diapositivas de explicación y exploración del problema iniciales, más una diapositiva que sirva de portada con el nombre de los/las componentes del equipo, nombre del equipo y asignación de algoritmos básicos y ensembles a cada miembro. Será obligatorio incluir, dentro de esas diapositivas por miembro:

- Una diapositiva final con la evolución de las métricas usando la validación elegida y el conjunto de entrenamiento y el valor final en el conjunto de test que se alcanza con el mejor modelo del proceso anterior; en formato gráfico para los dos algoritmos (básico y ensemble) en la misma gráfica.

- Una diapositiva con la lista completa de algoritmos de preprocesamiento utilizados y sus diferentes configuraciones a lo largo de la práctica (qué se ha probado en el transcurso de la misma con intervalos de hiperparámetros). Además, también se considerarán las diferentes configuraciones (hiperparámetros) del algoritmo de aprendizaje.

El resto de diapositivas se puede utilizar para describir el proceso y detallar mejor los resultados, motivaciones, justificaciones, decisiones tomadas, etc. No se debe incluir código en la presentación ni descripciones teóricas de los algoritmos explicados en clase. Sí que será necesario describir aquellas técnicas utilizadas que no se hayan visto en clase (breve descripción), incluir referencias, etc, ...