Trabajo final para la asignatura APBD

Máster en Data Science y Big Data. CFP-Universidad de Sevilla

Juan Galán Páez
Dpto. Ciencias de la Computación e Inteligencia Artificial

Abril, 2024

1 Instrucciones

1.1 Problema

El problema elegido para el trabajo es uno de los conjuntos de datos de aprendizaje de kaggle, en el que el objetivo es predecir el precio de venta de viviendas a partir de sus características. Para más información sobre el problema y descargar los datos entre en la página de la competición en kaggle.

1.2 Objetivos

1.2.1 Spark

El objetivo de este trabajo es practicar con las diferentes herramientas que Spark nos proporciona para el procesamiento distribuido de grandes cantidades de datos. Aunque, para evitar problemas de recursos, se ha elegido un conjunto de datos pequeño, que pueda ser procesado mediante Spark en cualquier máquina, debemos suponer que estamos ante un problema Big Data, lo que tiene una serie de implicaciones:

- Los conjuntos de datos son tan grandes que no pueden ser procesados en una sola máquina (por eso usamos tecnologías distribuidas).

- Cualquier tratamiento del conjunto de datos debe realizarse mediante las herramientas que Spark proporciona. Es decir, no podemos cargar en python local (Pandas, Numpy, etc.) el dataset, procesarlo y luego volver a subirlo a Spark. No confundir Pandas con *Pandas on Spark*, que sí podemos usar.

- Como hemos visto en clase, es viable traer de spark pequeños fragmentos de datos (normalmente estadísticos y agregaciones) para procesarlos de forma más cómoda mediante Pandas o visualizarlos mediante matplotlib.

1.2.2 Ténicas de Machine Learning

El principal objetivo del trabajo es la resolución de problemas mediante Spark, por lo que la obtención de predicciones optimales es un objetivo secundario (aunque se valorará positivamente).

En la página de la competición en kaggle, encontraremos numerosas soluciones de otros usuarios que nos pueden servir de guía. Algunas consideraciones:

- Se permite que nuestras soluciones estén inspiradas en soluciones de otros usuarios realizadas en otros lenguajes.

- Es decir, no se permite copiar código de otras soluciones realizadas en Spark.

- Todas las decisiones que se tomen en el proceso de análisis deben estar justificadas en el código. Por ejemplo, no se admite tomar el conjunto de hiperparámetros o la selección de variables óptima de la solución de un usuario y aplicarlo directamente a la nuestra. Si se seleccionan variables o parámetros, dicha selección irá acompañada del procedimiento de selección de variables o ajuste de hiperparámetros correspondiente.

1.3 Solución al problema

El problema a resolver tiene bastante similitud con lo visto en clase, en las prácticas 3 y 4, para el dataset del Titanic. Aunque en este caso el problema a resolver es de regresión.

La solución al problema empezará con la exploración y el preprocesado, donde prepararemos el dataset para luego poder aplicar modelos y ajuste de hiperparámetros. Dadas las características del conjunto de datos, en el

preprocesado, como mínimo, tendremos que tratar valores perdidos y variables categóricas. Otras tareas interesantes pueden ser selección de variables (correlaciones, importancia de variables, etc.), reducción de dimensionalidad, normalización, conversión de categóricas a oneHot, etc.

Con respecto a la parte de análisis se deben probar varios modelos y ajustar sus hiperparámetros mediante validación cruzada. La evaluación del modelo final debe hacerse con un conjunto de datos diferente al de entrenamiento (como vimos en clase). Se recomienda automatizar procesos mediante el uso de pipelines, funciones y bucles para evitar repetir código, especialmente teniendo en cuenta el elevado número de variables del conjunto de datos.

1.3.1 Métrica

La métrica a usar para la evaluación de soluciones es RMSE (Root Mean Square Error).

NOTA: La métrica de la competición es **RMSLE** (*Root Mean Square Logarithmic Error*), sin embargo, esta métrica no se encuentra dentro de las que ofrece el objeto *RegressionEvaluator* por lo que tendríamos que definir nuestro propio evaluador, cosa que no hemos visto. Como aproximación también se podría, opcionalmente, trabajar con el RMSE y la transformación logarítmica de la respuesta. Adicionalmente, se proporciona un fichero *rmsle.py* que contiene una implementación para Spark de la métrica RMSLE.

1.4 Herramientas

Las APIs de Spark más importantes para este proyecto serán la API de DataFrames (o la API de Pandas en Spark si se prefiere) y la API de ML para DataFrames. En el caso de los DataFrames hay libertad para usar los métodos de la API o lenguaje SQL según nos sintamos más cómodos.

En clase se ha trabajado con la familia de algoritmos y evaluadores para clasificación binaria. En este caso estamos ante un problema de regresión por lo que debemos usar la familia de algoritmos y evaluadores asociada a este tipo de problemas. La documentación para algoritmos de regresión se puede ver en este enlace y la API en este otro.

Recordemos que también tenemos a nuestra disposición la API RDDs para procesamiento a bajo nivel (e.g. transformación de características).

1.5 Entrega

- La entrega del trabajo se hará a través de la plataforma de enseñanza virtual.

- El entregable será un fichero comprimido que contenga el notebook con la solución y la estructura de carpetas/ficheros adicionales que se haya usado. El notebook con la solución debe estar comentado (brevemente).
- La solución debe contener como mínimo una **fase de preprocesado y tratamiento de valores perdidos** y **otra de modelado** con al menos **dos modelos diferentes** (con su correspondiente **ajuste de hiperparámetros**), de forma similar a lo visto en clase. Se valorará positivamente el uso de funciones y objetos que no se han visto en clase (aunque no es necesario).
- Por último, es interesante aunque opcional, hacer una subida a kaggle. En ese caso, se indicará en la entrega el usuario de kaggle con el que se ha realizado dicha subida. Recordemos que esto requiere realizar el preprocesado y preparación del dataset completo (test + train).
- Las dudas se resolverán por correo o en tutoría (concretar día y hora por correo).
- Los trabajos son individuales. Los trabajos copiados no serán evaluados.