GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING PARA LA ELIMINACIÓN DE INFORMACIÓN SENSIBLE

Autor: Beatriz Martínez García

Director: Alejandro Llorente Pinto

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Aplicación de Técnicas de Machine Learning para la eliminación de información sensible en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2022/23 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

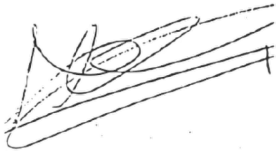
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Beatriz Martínez García Fecha: 22/ 06/ 2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Alejandro Llorente Pinto Fecha: 22 / 06 / 2023

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING PARA LA ELIMINACIÓN DE INFORMACIÓN SENSIBLE

Autor: Beatriz Martínez García

Director: Alejandro Llorente Pinto

Madrid

APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING PARA LA ELIMINACIÓN DE INFORMACIÓN SENSIBLE

Autor: Martínez García, Beatriz.
Director: Llorente Pinto, Alejandro.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este trabajo de investigación pretende aplicar técnicas basadas en modelos de regresión lineal y regresión logística para evaluar modelos entrenados partir de datos reales (por medio de Python). Tras el entrenamiento, los objetivos son determinar el impacto que tienen las técnicas de eliminación de información en el proceso de predicción, así como estudiar técnicas que permitan eliminar información sensible implícita en las variables que lo constituyen. En el trabajo se utiliza un conjunto de datos público que hace referencia a información obtenida por el gobierno de Estados Unidos sobre la vivienda en el área de Boston. Estos datos contienen información implícita y explícita sobre la proporción de personas de raza negra por ciudad y del porcentaje de personas de estatus inferior, variables que se han considerado sensibles. Tras aplicar técnicas de medida de equidad en los modelos, se ha determinado que el modelo discrimina por las variables mencionadas. Por ello, se han entrenado dos modelos adicionales utilizando técnicas de reducción de *unfairness*. El primer modelo ha sido entrenado tras eliminar las variables sensibles del conjunto de datos y se ha concluido que no es suficiente para lograr un modelo que no discrimina debido a la existencia de información sensible implícita en otras variables. Para el segundo modelo se ha utilizado un conjunto de datos transformado por el Método de Gram-Schmidt. Se ha conseguido un modelo más equitativo que los dos anteriores, a costa de una pérdida en la capacidad predictiva.

Palabras clave: equidad, Aprendizaje Automático, regresión lineal, regresión logística, información sensible.

1. Introducción

La supervisión humana está siendo sustituida por algoritmos de análisis de datos que permiten optimizar el proceso y obtener predicciones más precisas que ayudan en la toma de decisiones. Esta tecnología, basada en la rama de la Inteligencia Artificial conocida como Aprendizaje Automático o Machine Learning, es muy novedosa y crece a una velocidad exponencial. La automatización de la toma de decisiones es una realidad y la falta de transparencia, justicia y responsabilidad durante el proceso puede resultar en un trato injusto hacia personas que pertenecen a grupos determinados. Es por este motivo por el que la relación entre ética y tecnología ha de adaptarse, y parte de esta adaptación consiste en nuevas normativas que se encargan de preservar los derechos fundamentales de las personas.

El problema principal reside en la inexistencia de un método global y claro para eliminar los aspectos discriminatorios de los modelos que ayudan en la toma de decisiones debido a la novedad de la tecnología. En este trabajo se propone la implementación de una técnica basada en el método de Gram-Schmidt que permite eliminar tanto la información sensible explícita como la implícita en el resto de las variables.

2. Definición del Proyecto

Es este trabajo se hace uso de un conjunto de datos público que recoge información sobre la vivienda del área de Boston recopilada por el gobierno de Estados Unidos. Es a partir de este dataset de donde se entrena el modelo. Se han identificado dos variables que pueden ser discriminatorias: la variable "B", donde aparece información sobre la proporción de personas de raza negra por ciudad, y la variable "LSTAT", que representa el porcentaje de personas de estatus inferior. Se sigue el siguiente procedimiento cada vez que se entrena un nuevo modelo:

- Preprocesamiento de los datos y entrenamiento del modelo. En este caso se entrena un modelo de regresión lineal, ya que la variable objetivo es de tipo continuo (valor medio de las viviendas ocupadas por sus propietarios en miles de dólares).
- Obtención de medidas estadísticas de interés: coeficiente de correlación lineal de Pearson, coeficiente de determinación R2.
- Medida de *fairness* tras el entrenamiento de los modelos. Para ello se ha dividido el conjunto de datos en tres tramos para cada una de las variables y se han comparado, obteniendo tres medidas de *fairness* para cada variable.

Durante el proyecto se entrenan tres modelos: el modelo original, para el que se hace uso del conjunto de datos sin modificar; un modelo a partir de un dataset en el que se han eliminado las columnas correspondientes a la variable "B" y la variable "LSTAT"; y un tercer modelo entrenado a partir de un conjunto de datos transformado por el método de Gram-Schmidt.

3. Descripción del modelo y herramientas utilizadas

En la sección anterior se hace un recorrido del procedimiento seguido para conseguir un modelo que no discrimine. En este apartado se muestra la forma que tiene el modelo entrenado, así como las herramientas utilizadas para conseguir eliminar el sesgo.

- Modelo de regresión lineal [6]:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \cdots + \beta_n * x_n$$

  Donde y es la variable dependiente, x_i las variables que explican el modelo y β_i los parámetros o coeficientes determinados durante el entrenamiento

- Medida de *fairness* para variables continuas [1]:

$$fairness = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N0} * \sum_{i=0}^{N0} \hat{y}i\,|$$

  Esta expresión representa la diferencia que existe entre las medias de las predicciones de los tramos en los que se divide el conjunto de datos para las dos variables.

- Método de Gram-Schmidt para la eliminación de información sensible implícita en el resto de las variables [5]:

  Dado un conjunto de vectores $\{v_1, v_2, \ldots, v_n\}$ linealmente independientes y pertenecientes a un espacio vectorial euclídeo V:

1) Se define una base ortogonal $\{u_1, u_2, \ldots, u_n\}$ como la base ortogonal perteneciente al espacio vectorial del conjunto de vectores definido anteriormente y se hace $u_1 = v_1$.
2) A continuación, para todo i > 1:

$$u_i = v_i - u_1 \frac{\langle v_i, u_1 \rangle}{\langle u_1, u_1 \rangle} - u_2 \frac{\langle v_i, u_2 \rangle}{\langle u_2, u_2 \rangle} - \cdots - u_{n-1} \frac{\langle v_i, u_{n-1} \rangle}{\langle u_{n-1}, u_{n-1} \rangle}$$

4. Resultados

Los resultados más importantes del trabajo tienen que ver con las medidas de fairness obtenidas tanto con el modelo original como con los modelos entrenados tras aplicar las técnicas de reducción de unfairness. Se muestran los resultados en la siguiente tabla, donde se representan las tres medidas de fairness obtenidas para las dos variables sensibles en cada caso:

Modelo 1 (original)			**Modelo 2 (tras eliminar variables sensibles)**			**Modelo 3 (tras aplicar método de Gram-Schmidt)**		
B, 1	B, 2	B, 3	B, 1	B, 2	B, 3	B, 1	B, 2	B, 3
9,9689	7,2170	9,2988	6,0742	6,2918	6,5281	5,0272	0,7301	3,3651
LSTAT,1	LSTAT,2	LSTAT,3	LSTAT,1	LSTAT,2	LSTAT,3	LSTAT,1	LSTAT,2	LSTAT,3
11,2278	4,6875	12,9240	9,8218	5,1816	9,1951	1,5909	1,8662	0,5154

Además de los resultados anteriores, también se considera importante la evaluación de la -capacidad predictiva de los modelos por medio del coeficiente de determinación R2.

- R2 modelo original: 0,741
- R2 modelo tras eliminar las variables sensibles: 0,670
- R2 modelo tras aplicar método de Gram-Schmidt: 0,223

5. Conclusiones

La diferencia en las medias de predicciones ha disminuido de manera considerable tras la aplicación de los métodos de reducción de unfairness, lo que significa que las técnicas aplicadas han funcionado y el modelo es menos discriminatorio, siendo el método de Gram-Schmidt el más efectivo.

Hay una pérdida considerable en la capacidad predictiva de los modelos asociada a la disminución del sesgo. Esto significa que las variables sensibles son predictivas del modelo.

6. Referencias

[1] Calders, T., Karim, A., Kamiran, F., Ali, W. y Zang, X. (2013). *Controlling Attribute Effect in Linear Regression*. 2013 IEEE 13th International Conference in Data Mining https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6729491

[2] Hardt, M., Price, E., Srebro, N. (2016). *Equality of Opportunity in Supervised Learning*. Advances in Neural Information Processing Systems 29 (NIPS 2016)

[3] Maisueche, A. (2019). *Utilización del Machine Learning en la Industria 4.0*. [TFM, Universidad de Valladolid] https://uvadoc.uva.es/bitstream/handle/10324/37908/TFM-I-1372.pdf?sequence=1&isAllowed=y

[4] Martínez de Ibarreta, C., Álvarez, C., Borrás, F., Budría, S., Curto, T. y Escobar, L. S. (2021). *Modelos Cuantitativos para la Economía y la Empresa en 101 ejemplos.* EV Services.

[5] Morocho, P. (2010). *Proceso de Gram Schmidt* [Diapositiva de PowerPoint]. SlideShare https://www.slideshare.net/paolamorochoa/proceso-de-gram-schmidt

[6] Starmer, J. [StatQuest with Josh Starmer]. (2017). *Linear Regression and Linear Models* [Lista de reproducción]. YouTube. https://www.youtube.com/watch?v=PaFPbb66DxQ&list=PLblh5JKOoLUIzaEkCLIUxQFjPIlapw8nU

APPLICATION OF MACHINE LEARNING TECHNIQUES FOR THE REMOVAL OF SENSITIVE INFORMATION

Author: Martínez García, Beatriz.
Supervisor: Llorente Pinto, Alejandro.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This research project aims to apply techniques based on linear regression and logistic regression models to evaluate models trained on real data using python. After training, the objectives are to determine the predictive capability of the model and study techniques that allow for the removal of implicit sensitive information in the variables that compose it. The study utilizes a public dataset obtained by the United States government regarding housing in the Boston area. These data contain implicit and explicit information about the proportion of black individuals per city and the percentage of individuals with lower socioeconomic status. These variables have been deemed sensitive. It has been determined through fairness measurement techniques in the models that the model discriminates based on the sensitive variables. Therefore, two additional models have been trained using unfairness reduction techniques. The first model was trained after removing the sensitive variables from the dataset, and it has been concluded that it is not sufficient to achieve a non-discriminatory model due to the existence of implicit sensitive information in other variables. For the second model, a dataset transformed by the Gram-Schmidt method has been used. A more equitable model has been achieved compared to the previous two, at the cost of a loss in predictive capability.

Keywords: fairness, Machine Learning, linear regression, logistic regression, sensitive information.

1. Introduction

Human supervision is being replaced by data analysis algorithms that optimize the process and provide more accurate predictions to aid decision-making. This technology, based on the branch of Artificial Intelligence known as Machine Learning, is highly innovative and growing at an exponential rate. The automation of decision-making is a reality, and the lack of transparency, fairness, and accountability during the process can result in unfair treatment towards specific groups of people. This is why the relationship between ethics and technology needs to adapt, and part of this adaptation involves new regulations that aim to preserve individuals' fundamental rights.

The main problem lies in the absence of a global and clear method to eliminate discriminatory aspects from decision-making models due to the novelty of the technology. This research proposes the implementation of a technique based on the Gram-Schmidt method, which allows for the removal of both explicit and implicit sensitive information from the remaining variables.

2. Project Definition

In this project, a publicly available dataset containing information about housing in the Boston area, collected by the United States government, is used. This dataset is used to train the model. Two variables that may be discriminatory have been identified: the variable "B," which contains information about the proportion of black individuals per city, and the variable "LSTAT," representing the percentage of individuals with lower socioeconomic status. The following procedure is followed each time a new model is trained:

- Data preprocessing and model training: In this case, a linear regression model is trained since the target variable is continuous (the average value of owner-occupied homes in thousands of dollars).
- Calculation of statistical measures of interest: Pearson's linear correlation coefficient, R-squared coefficient of determination.
- Fairness measurement after model training: The dataset is divided into three groups for each variable, and a comparison is made, resulting in three fairness measures for each variable.

Throughout the project, three models are trained: the original model, using the unmodified dataset; a model using a dataset where the columns corresponding to the "B" and "LSTAT" variables have been removed; and a third model trained using a dataset transformed by the Gram-Schmidt method.

3. Model description

In the previous section, an overview of the procedure followed to achieve a non-discriminatory model is provided. In this section, we will discuss the structure of the trained model and the tools used to eliminate bias.

- Linear regression model [6]:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \cdots + \beta_n * x_n$$

  Where y is the dependent variable, x_i are the independent variables, and β_i are the coefficients determined during training.

- Fainess measures for continuous variables [1]:

$$fairness = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N0} * \sum_{i=0}^{N0} \hat{y}i\ |$$

  This expression represents the difference between the means of the predictions for the divided segments of the dataset for the two variables.

- Gram-Schmidt method for the removal of implicit sensitive information in the remaining variables [5].

  Given a set of vectors {v1, v2, ..., vn} that are linearly independent and belong to a Euclidean vector space V.

  1) An orthogonal basis {u1, u2, ..., un} is defined as the orthogonal basis belonging to the vector space of the previously defined set of vectors, and u1 is set equal to v1.

2) Then, for all i > 1:

$$u_i = v_i - u_1 \frac{\langle v_i, u_1 \rangle}{\langle u_1, u_1 \rangle} - u_2 \frac{\langle v_i, u_2 \rangle}{\langle u_2, u_2 \rangle} - \cdots - u_{n-1} \frac{\langle v_i, u_{n-1} \rangle}{\langle u_{n-1}, u_{n-1} \rangle}$$

4. Results

The most important results of the study are related to the fairness measures obtained for both the original model and the models trained after applying unfairness reduction techniques. The results are presented in the following table, which shows the three fairness measures obtained for the two sensitive variables in each case:

Model 1 (original)			**Model 2 (after removing the sensitive variables)**			**Model 3 (after Gram-Schmidt method)**		
B, 1	B, 2	B, 3	B, 1	B, 2	B, 3	B, 1	B, 2	B, 3
9,9689	7,2170	9,2988	6,0742	6,2918	6,5281	5,0272	0,7301	3,3651
LSTAT,1	LSTAT,2	LSTAT,3	LSTAT,1	LSTAT,2	LSTAT,3	LSTAT,1	LSTAT,2	LSTAT,3
11,2278	4,6875	12,9240	9,8218	5,1816	9,1951	1,5909	1,8662	0,5154

In addition to the previous results, the evaluation of the predictive capability of the models through the coefficient of determination R2 is also considered important.

- R2 for the original model: 0.741.
- R2 for the model after removing the sensitive variables: 0.670.
- R2 for the model after applying the Gram-Schmidt method: 0.223.

5. Conclusion

The difference in prediction means has significantly decreased after applying the unfairness reduction methods, indicating that the applied techniques have been effective in making the model less discriminatory, with the Gram-Schmidt method being the most effective.

There is a considerable loss in the predictive capability of the models associated with the reduction of bias. This suggests that the sensitive variables are predictive of the model.

6. References

[1] Calders, T., Karim, A., Kamiran, F., Ali, W. y Zang, X. (2013*). Controlling Attribute Effect in Linear Regression.* 2013 IEEE 13th International Conference in Data Mining https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6729491

[2] Hardt, M., Price, E., Srebro, N. (2016). *Equality of Opportunity in Supervised Learning*. Advances in Neural Information Processing Systems 29 (NIPS 2016)

[3] Maisueche, A. (2019). *Utilización del Machine Learning en la Industria 4.0*. [TFM, Universidad de Valladolid]

https://uvadoc.uva.es/bitstream/handle/10324/37908/TFM-I-1372.pdf?sequence=1&isAllowed=y

[4] Martínez de Ibarreta, C., Álvarez, C., Borrás, F., Budría, S., Curto, T. y Escobar, L. S. (2021). *Modelos Cuantitativos para la Economía y la Empresa en 101 ejemplos*. EV Services.

[5] Morocho, P. (2010). *Proceso de Gram Schmidt* [Diapositiva de PowerPoint]. SlideShare https://www.slideshare.net/paolamorochoa/proceso-de-gram-schmidt

[6] Starmer, J. [StatQuest with Josh Starmer]. (2017). *Linear Regression and Linear Models* [Lista de reproducción]. YouTube. https://www.youtube.com/watch?v=PaFPbb66DxQ&list=PLblh5JKOoLUIzaEkCLIUxQFjPIlapw8nU

Índice de la memoria

Índice de figuras

Ilustraciones

Ecuaciones

Índice de tablas

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

La Inteligencia Artificial, y en concreto el Machine Learning, es un sector nuevo y con un gran potencial de crecimiento que cobra cada vez más importancia a nivel social. La toma de decisiones a nivel empresarial está cada vez más automatizada y resulta inevitable que las acciones que llevan a cabo las empresas, fruto de estas decisiones, sean determinadas por algoritmos que manejan grandes cantidades de datos. Las empresas, en constante contacto con el cliente, son conscientes de que muchos patrones de comportamiento se encuentran implícitos en los datos.

Es importante que las decisiones que impulsan a las empresas a actuar sigan un código ético y cumplan con las regulaciones antidiscriminatorias establecidas por los gobiernos. Por eso, y porque la implementación de algoritmos y técnicas de manejo de datos masivos se llevó a cabo hace tan solo unos años, es necesario estudiar la relación entre la ética y la tecnología de una manera en la que no se había hecho antes. El objetivo de este trabajo es implementar técnicas de Machine Learning para identificar y reducir el impacto de variables discriminatorias en modelos predictivos utilizados en la toma de decisiones, para que estas sean más equitativas y justas.

1.2 PROBLEMÁTICA Y POSIBLES SOLUCIONES

La supervisión humana está siendo sustituida por algoritmos basados en análisis de datos con el objetivo de optimizar y obtener predicciones más precisas que ayuden a la toma de decisiones. Sin embargo, la falta de transparencia, responsabilidad y justicia durante el proceso puede desembocar en el trato injusto hacia ciertos grupos de personas que comparten un atributo determinado, o en un impacto desproporcionado sobre el mismo en el caso de que la acción que se desencadena de esa acción se lleve a cabo.

El problema principal reside en la inexistencia de un método claro y global para eliminar la discriminación de los modelos de predicción. Esto sucede debido a la novedad del Machine Learning y de la ética aplicada a la Inteligencia Artificial. Podría pensarse que para evitar el trato injusto de un grupo determinado bastaría con obviar las variables que contienen esa información de manera directa. Sin embargo, los sistemas automatizados pueden ser entrenados con datos históricos y otras variables pueden contener de manera implícita la información sensible que se quiere evitar. Por ejemplo, si se quiere evitar que el sexo de una persona tenga algún tipo de influencia sobre la decisión que se quiere tomar, pretender que eliminar esa variable del modelo soluciona el problema no es del todo correcto, ya que otras variables como su profesión pueden estar correlacionadas con la variable que recoge la información sobre el sexo (por ejemplo, en el sector sanitario es mayoritaria la presencia femenina). Si un modelo de Machine Learning es entrenado con datos que reflejan desigualdades existentes en la sociedad, las predicciones del modelo pueden ser sesgadas en contra de ciertos grupos de personas.

El objetivo de automatizar los procesos con técnicas de análisis de datos es facilitar predicciones más precisas y óptimas que proporcionen una mayor satisfacción al usuario y a las partes interesadas. Si a esto se une la necesidad de cumplir con una determinada regulación antidiscriminatoria e introduciendo el concepto de *fairness* como la búsqueda de la equidad durante el proceso, el objetivo del análisis es minimizar la pérdida de precisión en la capacidad de predicción de los modelos, buscando que sean lo más equitativos posible. Los casos de uso más habituales de la aplicación de técnicas de cálculo del *fairness* incluyen sistemas de crédito, contratación, y sistemas de seguridad pública. En estos casos, es especialmente importante asegurar que las decisiones tomadas por los modelos de Machine Learning no estén sesgadas por variables sensibles como la raza, género, orientación sexual, u origen étnico.

Existen varias técnicas que pueden utilizarse para asegurar que las predicciones de un modelo de Machine Learning no estén sesgadas por variables sensibles. Una técnica comúnmente utilizada es la de "re-muestreo", que implica utilizar un subconjunto de los datos de entrenamiento para asegurar que los modelos estén expuestos a una variedad de

personas de diferentes hgrupos. Otra técnica es "ponderar las clases", que implica asignar pesos más altos a las instancias de los datos de entrenamiento que pertenecen a los grupos subrepresentados en el conjunto de datos.

Sin embargo, existe otra aproximación que permite, por un lado, eliminar las variables sensibles y, por otro, descontar la información de estas variables sensibles sobre el resto de las variables. Para realizar estos procesos, una de las técnicas propuestas es la aplicación del método de Gram-Schmidt, un método matemático utilizado para generar un conjunto ortogonal de vectores a partir de un conjunto de vectores dado. Este método se utiliza comúnmente en álgebra lineal y en la teoría de la información.

Para aplicar el método de Gram-Schmidt en este contexto, primero se seleccionan las variables sensibles de los datos de entrenamiento. Estas variables se utilizan para crear un conjunto de vectores que representan la información de las variables sensibles. A continuación, se utiliza el método de Gram-Schmidt para generar un conjunto ortogonal de vectores que representan la información de las variables sensibles.

Una vez que se ha generado el conjunto ortogonal de vectores, se pueden eliminar de los datos de entrenamiento originales. Esto se puede hacer eliminando las columnas correspondientes a las variables sensibles o mediante técnicas de proyección. De esta manera se pueden entrenar modelos de aprendizaje automático sin la información de las variables sensibles. Estos modelos deben producir predicciones menos sesgadas que los modelos entrenados con toda la información de los datos de entrenamiento.

Sin embargo, es importante tener en cuenta que la eliminación de la información de las variables que contienen información sensible también puede reducir la precisión del modelo. Por lo tanto, es importante evaluar el rendimiento del modelo después de aplicar el método de Gram-Schmidt y decidir si el sacrificio en la precisión es justificado por la reducción del sesgo.

Capítulo 2. ESTADO DE LA CUESTIÓN

2.1 INTELIGENCIA ARTIFICIAL Y MACHINE LEARNING

Se entiende como transformación digital en un ámbito empresarial al proceso de adoptar soluciones tecnológicas e innovadoras en los modelos de negocio [12]. Este es un hecho que ha supuesto retos y cambios en todos los sectores y empresas, sin importar su tamaño. La digitalización de los diferentes procesos dentro de una organización ha supuesto cambios nunca vistos a una velocidad que requiere de una capacidad de aprendizaje sin precedentes. Hasta este momento, a lo largo de la historia han tenido lugar tres revoluciones industriales que transformaron la manera de vivir de las personas de la época. El impacto en la sociedad y en la economía que han tenido las nuevas tecnologías hacen que se pueda hablar de una Cuarta Revolución Industrial, también conocida como Industria 4.0. Es en este contexto donde aparece la Inteligencia Artificial, la habilidad que tiene una computadora para presentar las mismas capacidades de un humano a nivel de procesamiento de información, aprendizaje y toma de decisiones [12].

El Machine Learning, o Aprendizaje Automático, es un área dentro de la Inteligencia Artificial que está cobrando cada vez más importancia en todos los sectores y que se fundamenta en las matemáticas, la estadística y la computación [12][9]. Esta forma de IA permite a un sistema aprender de los datos en vez de aprender mediante la programación explícita. De esta manera, y mediante procesos iterativos, se programan algoritmos capaces de aprender de los datos para realizar predicciones cada vez más precisas que ayudan a la toma de decisiones. Este trabajo está centrado en modelos de ML de tipo predictivo. El entrenamiento de este tipo de modelos tiene como objetivo obtener un modelo capaz de ofrecer una predicción lo más precisa posible a partir de una serie de inputs o valores de las variables de entrada.

Se ha mencionado que el Machine Learning se basa en las matemáticas, la estadística y la computación. Pues bien, es lógico pensar que anterior a los avances en computación debieron

existir métodos predictivos menos precisos basados en la estadística y las matemáticas, ciencias mucho más antiguas. Estas técnicas son el Teorema de Bayes, el método de Mínimos Cuadrados y las Cadenas de Márkov [12]. A partir de la década de 1940 comenzaron los avances relacionados con la computación.

Bernard Marr, en un artículo de Forbes titulado *"A Short History of Machine-Learning Every Manager Should Read"* [14], hace un viaje en el tiempo recordando los mayores avances en lo que a Inteligencia Artificial se refiere, desde 1950, cuando se creó el "Test de Turing" que pretendía descubrir si un ordenador era capaz de engañar a un ser humano, hasta 2016, cuando el algoritmo *AlphaGo* desarrollado por Google fue capaz de ganar al *Go*, el juego de mesa de origen chino considerado el juego más complejo del mundo, a un jugador profesional. Entre estas dos fechas se dieron numerosos avances, como la creación de las redes neuronales [14]. Es importante destacar la rapidez con la que se producen cambios en este sector. En lo que se refiere al Aprendizaje Automático, todos los años existe una evolución significativa. Por ejemplo, en los últimos años, los Modelos de Lenguaje han revolucionado el procesamiento de lenguaje natural. Modelos como BERT (Bidirectional Encoder Representations from Transformers) o GPT (Generative Pre-trained Transformer) han supuesto un gran avance para tareas de traducción automática, clasificación de texto, respuesta y formulación de preguntas y análisis de sentimientos, entre otros [17].

Dentro del Machine Learning existen cuatro tipos distintos de aprendizaje: supervisado, no supervisado, reforzado y profundo (*Deep Learning*). Se utiliza aprendizaje supervisado cuando se desea obtener una función que depende de datos conocidos. Se emplean algoritmos que aprenden de datos de entrada conocidos y aprenden de manera iterativa para realizar una tarea específica [14].

La regresión es un tipo de aprendizaje supervisado y es en lo que se centra este trabajo, por lo que se hablará de este tipo de aprendizaje. Hay dos tipos de aprendizaje supervisado: clasificación y regresión. Por un lado, en la clasificación se entrenan algoritmos que devuelven un valor discreto y, por otro lado, en la regresión se produce una predicción que es función de una serie de datos de entrada, cuyo valor no tiene por qué ser discreto [1].

Por último, es importante mencionar el concepto de singularidad tecnológica. Este concepto se entiende a partir de la idea de que la tecnología crece de manera exponencial y la singularidad se alcanzará el momento en el que la inteligencia artificial haya evolucionado de tal manera que no será controlable por el ser humano, su creador [20]. Esto supondría un cambio absoluto en el curso de la historia y es por este motivo por el que, aunque no sea muy probable en estos momentos, hay personas que se dedican a calcular la probabilidad de que esto ocurra. Tal vez, esta situación puede sonar un tanto extrema en la situación actual. Sin embargo, da que pensar y supone uno de los numerosos motivos por los cuales se está redefiniendo la relación entre la ética y la tecnología, así como la normativa que se encarga de regular y limitar todo lo que tiene que ver con los nuevos avances.

2.2 REGULACIÓN ACTUAL

Esta sección está basada en la Propuesta de Reglamento del Parlamento Europeo y del Consejo por el que se establecen normas armonizadas en materia de Inteligencia Artificial (Ley de Inteligencia Artificial) y se modifican determinados actos legislativos de la unión (Bruselas, 2021) [18]. El objetivo de este apartado es dar a conocer bajo qué reglamento se encuentra España en el momento de la realización del trabajo, así como dar a conocer de manera general los aspectos que interesa conocer que apoyan la investigación en el ámbito de la ética aplicada a la Inteligencia Artificial.

En el apartado 3.5 de la *Exposición de Motivos* se determina que la propuesta pretende preservar los derechos recogidos en la Carta de los Derechos Fundamentales de la Unión Europea, teniendo en cuenta que el uso de la IA puede repercutir de manera negativa en los mismos. Para ello hace referencia a los diversos artículos de la Carta que pretende proteger. Para este trabajo resulta interesante detenerse en el art.1 (derecho a la dignidad humana), el art. 21 (la no discriminación) y el art.23 (la igualdad entre hombres y mujeres).

En base al art. 1 el Reglamento establece normas de uso de la IA, prohibiciones de determinadas prácticas y requisitos, obligaciones y normas para los operadores y sistemas de IA de alto riesgo, así como normas sobre el control y vigilancia en el mercado. El objetivo

de este reglamento es que los sistemas de IA sean utilizados de manera segura y ética, de manera que se respeten los derechos y libertades fundamentales de las personas.

2.3 EQUIDAD EN MACHINE LEARNING

El estado de la cuestión en el ámbito de equidad en Machine Learning se refiere a justicia tanto en el desarrollo como en el uso de los procesos automáticos de toma de decisiones. Esto incluye las áreas que se encargan de detectar los sesgos en los datos, los métodos para eliminar o reducir los sesgos en los modelos o el análisis e interpretación de los modelos que lleva a la toma de decisiones y posterior acción por parte de las empresas.

En el apartado anterior, se ha comentado que el concepto de equidad se refiere a la no discriminación tanto en el proceso de toma de decisiones, como en el impacto que tiene la acción final derivada de las mismas. El objetivo final es evitar estos dos tipos de discriminación.

La importancia de las técnicas de Machine Learning para optimizar procesos está aumentando a una gran velocidad. Se trata de un área con un gran margen de mejora y hoy en día no existe un método global para asegurar la equidad en los modelos derivados de grandes conjuntos de datos. Muchos de los estudios que se han llevado a cabo sobre este tema se centran en eliminar uno o los dos tipos de discriminación y presentan una o más limitaciones como la incapacidad de evitar los dos tipos de discriminación al mismo tiempo, la incapacidad de abarcar más de un atributo sensible o la limitación a un pequeño rango de clasificadores.

Estos estudios típicamente siguen una de dos estrategias: la primera consiste en el preprocesamiento de los datos de entrenamiento y la segunda se basa en modificar clasificadores existentes. Ejemplos de estudios que siguen la primera estrategia son Dwork et al., 2012 [5], que defiende que individuos similares sean tratados de manera parecida introduciendo una medida que determine la similitud entre los individuos respecto a la tarea en cuestión, o Feldman et al., 2015 [6], un estudio que se centra en minimizar el impacto desproporcionado de una decisión sobre un grupo de sujetos que compartan una característica determinada. En cuanto a estudios que siguen la segunda estrategia, se puede

encontrar Kamishima et al.,2015 [11], un estudio en el que se utiliza la regresión logística como método efectivo para reducir el impacto de las decisiones.

Por otra parte, otro estudio más completo [2] introduce un método capaz de reducir ambos tipos de discriminación. Utiliza la covarianza entre las variables sensibles y la distancia de las variables sensibles a la frontera de decisión como medida de *fairness* y se presentan dos formulaciones complementarias. La primera consiste en maximizar la precisión con restricciones de equidad y la segunda pretende minimizar la discriminación, esto es, maximizar la equidad, con restricciones sobre la precisión de los modelos. Esta última formulación no es óptima para la toma de decisiones a nivel empresarial ya que, si la correlación entre las variables sensibles y el resto de los atributos es alta de base, maximizar la equidad supondría pérdidas significativas en la precisión.

Por último, Hardt et al., 2016 [8], se basa en un principio de equidad que pretende que las tasas de verdadero positivo y falso positivos sean las mismas para varios grupos de personas. Se formula un predictor que depende de la variable sensible y no discrimina con respecto a la misma y que es la respuesta a una *loss function.*

Capítulo 3. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El trabajo se basa en el estudio de técnicas basadas en modelos de regresión lineal y regresión logística como base para interpretar modelos entrenados con datos reales que permiten realizar predicciones que ayudan a la toma de decisiones. Este apartado tiene como objetivo justificar la elección de los diferentes recursos que permitirán cumplir con los objetivos, así como proporcionar las explicaciones teóricas que hay detrás de las funciones que se programarán durante la realización del trabajo.

3.1 LENGUAJE Y ENTORNO DE PROGRAMACIÓN

El lenguaje de programación que permitirá la realización del trabajo es Python. Se utilizarán las siguientes bibliotecas.

- Pandas: biblioteca de Python utilizada para analizar y manipular datos en DataFrames o Series. En este trabajo va a resultar muy útil para preparar los datos antes de aplicar técnicas de Aprendizaje Automático.
- Numpy: biblioteca de Python que permite trabajar con arrays multidimensionales y funciones matemáticas.
- Matplotlib: biblioteca de Python utilizada para la obtención de gráficos de diferentes tipos.
- Statsmodels: biblioteca de Python utilizada para realizar análisis estadísticos. Va a proporcionar medidas estadísticas de interés para la evaluación de los modelos.
- Scikit-learn: biblioteca de Python de Aprendizaje Automático ampliamente utilizada. Va a permitir entrenar los modelos y obtener medidas útiles para su evaluación.
- Scikit-lego: biblioteca de Python que se basa en “scikit-learn” que proporciona más herramientas para la programación de algoritmos de Aprendizaje Automático.

El entorno de programación utilizado es el de Google Colab. Se trata de un entorno que permite programar y ejecutar Python en el propio navegador sin necesidad de configuración.

Otra ventaja de esta manera de programar es que permite compartir el código de manera sencilla, ya que se almacena en Google Drive. Además, en los cuadernos de Colab es posible combinar texto y código, lo que resulta muy útil a la hora de incluir anotaciones o explicaciones. Como punto negativo de este entorno de programación está la restricción de tiempo, ya que no se puede ejecutar un programa más de 12 horas seguidas y el ordenador debe estar encendido para poder ejecutarse.

3.2 ENTRENAMIENTO DE LOS MODELOS

Para entrenar los modelos por medio de Python, se utilizará la regresión lineal o la regresión logística en función de la naturaleza de la variable objetivo. Si esta variable es continua, se procederá a entrenar un modelo de regresión lineal. Si la variable objetivo es de tipo binario el modelo que se ha de entrenar es un modelo de regresión logística. Una regresión es una función que pretende ofrecer un valor para la variable objetivo, dados unos valores determinados de las variables explicativas. Esta sección busca explicar la forma que tienen ambos tipos de regresión.

3.2.1 REGRESIÓN LINEAL

Hay dos tipos de regresión lineal: regresión lineal simple y regresión lineal múltiple [21]. Este trabajo se centrará en la regresión lineal múltiple, ya que habrá más de una variable explicativa del modelo. En caso de que se decida entrenar un modelo de regresión lineal, se obtendrá una ecuación del tipo [21]:

$$E\,1:\ y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \cdots + \beta_n * x_n$$

En esta ecuación, por una parte, y representa la variable dependiente, variable endógena o variable objetivo. Por otra parte, x_i, para todo i comprendido entre 1 y el número total de variables n, son las variables independientes o explicativas del modelo y β_i, para todo i comprendido entre 1 y n, los parámetros o coeficientes que muestran la influencia de cada variable independiente sobre la variable endógena. Los parámetros miden efectos marginales. Cabe destacar que β_0 es el término independiente de la ecuación y no se

interpreta. El método utilizado para obtener la mejor predicción dado un conjunto de datos es el método de mínimos cuadrados. [15]

3.2.1.1 Método de Mínimos Cuadrados

En esta sección se introduce el concepto de error o residuo. Este término representa la diferencia existente entre la estimación y la realidad para cada observación [15]. El método de mínimos cuadrados busca que el valor de los residuos sea lo más próximo a cero posible. Un valor igual a cero para todos los residuos supondría un ajuste perfecto, ya que el error sería nulo.

Asumiendo que se va a producir algún tipo de error, la situación ideal sería que se produjesen tanto errores positivos como negativos, ya que si todos los residuos fuesen del mismo signo es probable que se estuviese produciendo una distorsión de los datos por estar sobrevalorando o infravalorando alguna de las variables. Esto explica que la media de los residuos sea igual a cero. De esta manera, el método de mínimos cuadrados busca minimizar la suma de los cuadrados de los residuos para evitar la cancelación en la suma de todos ellos [15].

3.2.1.2 Bondad del ajuste

La bondad del ajuste, medida por el *coeficiente de determinación* R^2, es la proporción de la variabilidad de la variable dependiente (y), explicada por la variabilidad de las variables independientes (x) [15]. En otras palabras, el coeficiente de determinación mide cómo de bueno es el modelo.

El valor de R^2 puede tomar cualquier valor comprendido entre cero y la unidad. Si el coeficiente de determinación de un modelo es igual a la unidad, entonces el modelo estaría explicando la realidad a la perfección.

En la sección anterior se introdujo el concepto de residuo como la diferencia existente entre la estimación y la realidad para cada observación. Teniendo en mente esta definición, se podría decir que la varianza de la variable dependiente y, se puede explicar como la suma de

la varianza de $\hat{y}$, es decir, la estimación de y y parte explicada por el modelo, y la varianza de e, el error o residuo no explicado por el modelo [15]. Matemáticamente:

$$E\,2: V(y) = V(\hat{y}) + V(e)$$

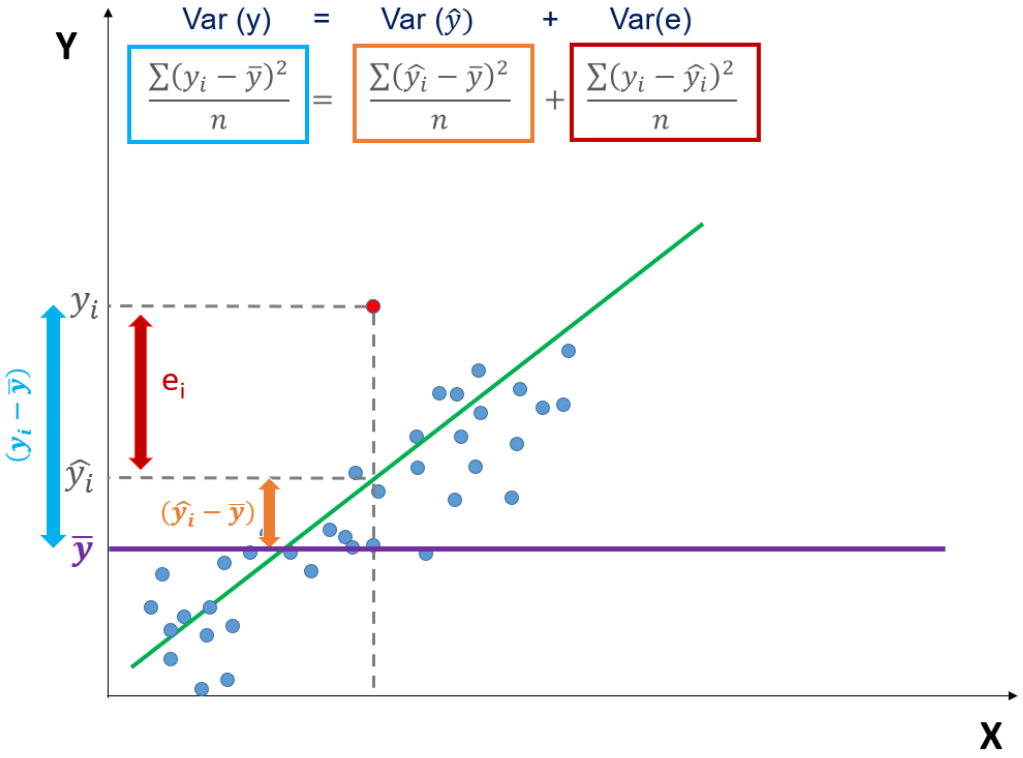


Ilustración 1: Bondad del ajuste en modelo de regresión [15]

Atendiendo a la definición con la que se ha iniciado este apartado, el coeficiente de determinación se calcula de la siguiente manera [15]:

$$E\,3: R^2 = \frac{V(\hat{y})}{V(y)} = 1 - \frac{V(e)}{V(y)}$$

Otro apunte importante sobre esta medida es que, para un mismo modelo, si aumenta el número de variables, aumenta el valor del coeficiente de determinación [15]. Es por este motivo por el que existe otra medida conocida como R^2 *corregido* que penaliza al modelo con más variables. Esta medida es muy útil para comparar modelos anidados, esto es, modelos que pretenden predecir el mismo suceso, pero que contienen un número diferente

de variables. Esta medida es necesaria, ya que hay variables que al añadirlas implican un aumento de la capacidad predictiva del modelo y otras que no tienen nada que ver con lo que se desea predecir.

3.2.2 REGRESIÓN LOGÍSTICA

Se entrenará un modelo de regresión logística en caso de que la variable objetivo sea una variable dicotómica [22]. El valor que devuelve la función para la variable objetivo en este caso es la probabilidad de que esta tome el valor 1 [15]. Es decir, si se estima que la variable dependiente tome un valor de 0.6, entonces la probabilidad de que el individuo para el que se están introduciendo las variables en concreto cumpla el suceso que se ha determinado como 1, es del 60%. A continuación, la Ilustración 2 representa la gráfica de una regresión logística simple:

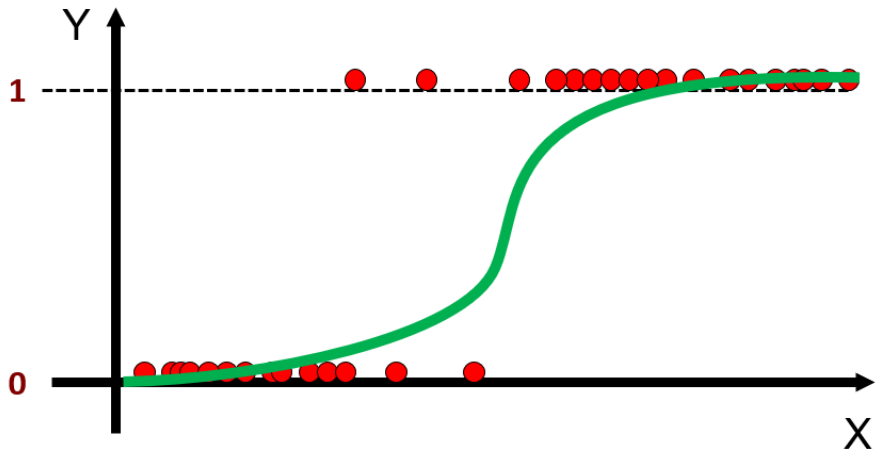


Ilustración 2: Modelo de regresión logística [15]

En esta gráfica, el eje Y representa la probabilidad en función de una determinada X. Como se puede observar, la gráfica no es lineal, por lo que la función establecida en la sección anterior para casos en los que se entrena un modelo de regresión lineal no es válida en este caso. Será necesario utilizar una función no lineal: el *logit* [15].

$$E\,4:\ Logit = Ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \cdots + \beta_n * x_n$$

$$E\,5:\ p = \frac{e^{Logit}}{1+e^{Logit}}$$

Donde *p* representa la probabilidad, *n* es el número de variables explicativas *x* y β_i para todo *i* perteneciente a *n* los parámetros a estimar. Obsérvese que, en este caso, al ser la pendiente variable en todos los puntos de la curva, los parámetros no miden efectos marginales.

Atendiendo a la expresión de la probabilidad y a la definición de decisión binaria, se intuye que es necesario establecer una regla que clasifique los sucesos como 0 o 1. Normalmente, la regla se basa en que los sucesos que tienen una probabilidad mayor a 0,5 se clasifican como 1 y viceversa [15].

A la hora de entrenar un modelo de regresión logística también es necesario estimar los parámetros. En este caso existe una diferencia fundamental con la estimación de los parámetros en un modelo de regresión lineal, para los cuales se acude al método de mínimos cuadrados. En este caso, el entrenamiento del modelo se hará por máxima verosimilitud [22].

3.2.2.1 Método de Máxima Verosimilitud

En el apartado anterior se ha mencionado que la manera de estimar los parámetros en un modelo de regresión logística es mediante el método de máxima verosimilitud. Este método busca encontrar cuáles son los coeficientes más probables dados unos datos determinados.

De la sección anterior también se obtiene que *p* es la probabilidad de que se produzca un suceso. Si se parte de que la muestra es aleatoria y las observaciones son independientes entre sí, entonces la estadística determina que la probabilidad conjunta se calcula como el producto de las probabilidades individuales. Esta es la manera de obtener la función de verosimilitud, que depende del valor de los parámetros. Una vez obtenida esta función, se deriva y se obtiene el valor más probable de los parámetros. [16]

3.2.2.2 Bondad del ajuste

Como en la sección referente a los modelos de regresión lineal, hay maneras de medir cómo de bueno es el ajuste para un modelo de regresión logística.

En esta sección se van a exponer dos métodos utilizados para medir la precisión de los modelos a la hora de predecir.

- Matriz de confusión

  Esta alternativa compara el número de casos predichos correctamente comparando las predicciones obtenidas mediante el modelo con el verdadero valor de las observaciones. Se calcula como un porcentaje siguiendo la siguiente lógica [15]:

$$E\ 6: porcentaje\ de\ casos\ correctos = \frac{nº\ de\ 1\ correctos + nº\ de\ 0\ correctos}{nº total\ de\ casos}$$

- Sensibilidad y especificidad

  La sensibilidad corresponde al número de "1" predichos correctamente y la especificidad corresponde al número de "0" correctamente predichos [15].

3.3MEDIDA DE FAIRNESS

Para determinar si el modelo entrenado es equitativo o no, se obtendrá una medida en la que se profundizará en apartados posteriores. Esta sección tiene como objetivo mostrar los diferentes métodos para obtener la medida en cuestión según la naturaleza de la variable que se quiera analizar.

3.3.1 PARA VARIABLES CONTINUAS

En caso de que la variable sensible sea continua, se hará uso de la siguiente fórmula [3]:

$$E\ 7: fairness = \left| \frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N0} * \sum_{i=0}^{N0} \hat{y}i \right|$$

Donde N1 es el número de personas que comparten un atributo determinado, N0 el número de personas que no comparten ese atributo, e $\hat{y}i$ la predicción dada por el modelo para un índice *i* determinado. La fórmula introducida en este caso representa la comparación de las medias de las predicciones de los dos grupos. Si el modelo predice de manera diferente para grupos con distintos atributos, entonces habrá una diferencia significativa entre las medias de las predicciones.

3.3.2 PARA VARIABLES BINARIAS

En esta sección se van a presentar dos métodos comunes para medir la equidad en un modelo de Aprendizaje Automático en el que la variable objetivo es una variable dicotómica.

El primer método que se va a presentar es la *paridad demográfica*. Este concepto estipula que la distribución de las predicciones debe ser idéntica entre las subpoblaciones [10]. Esto quiere decir que, si se tiene un atributo sensible, la decisión que se quiere tomar debe ser independiente de este, consiguiendo una representación equitativa de los grupos demográficos en cada situación. En términos matemáticos [6]:

$$E\ 8:\ p(\hat{y} = 1|A = 1) = p(\hat{y} = 1|A = 0)$$

Siendo $\hat{y}$ una decisión binaria y A el atributo que se considera sensible.

La medida que se utiliza para medir la paridad demográfica se conoce como *p%-rule* y tiene la siguiente forma [6]:

$$E\ 9:\ p\%\ rule = \frac{p}{100} \leq min\left(\frac{p(\hat{y} = 1|A = 1)}{p(\hat{y} = 1|A = 0)}, \frac{p(\hat{y} = 1|A = 0)}{p(\hat{y} = 1|A = 1)}\right)$$

Este es un método común, útil para casos en los que interesa que haya diversidad demográfica. Por ejemplo, en el caso de la contratación, donde puede interesar tener una plantilla diversa en términos de etnia o sexo. Sin embargo, este método tiene limitaciones, ya que garantiza la existencia de diversidad, pero no de justicia. Para casos en los que se quiera garantizar que un sujeto tenga las mismas oportunidades sin importar si pertenece o no a un grupo determinado, se utiliza el concepto de *igualdad de oportunidades*.

La igualdad de oportunidades busca que personas que pertenezcan a grupos demográficos diferentes tengan las mismas oportunidades. Matemáticamente [8]:

$$E\ 10:\ p(\hat{y} = 1|A = 1, y = 1) = p(\hat{y} = 1|A = 0, y = 1)$$

Siendo $\hat{y}$ una decisión binaria, A el atributo que se considera sensible e y el valor real de la decisión.

De la misma manera que en el caso anterior, existe una medida de igualdad de oportunidades. Esta sigue la siguiente expresión [8]:

$$E\ 11:\ igualdad\ de\ oportunidades = min\left(\frac{p(\hat{y}=1|A=1,y=1)}{p(\hat{y}=1|A=0,y=1)}, \frac{p(\hat{y}=1|A=0,y=1)}{p(\hat{y}=1|A=1,y=1)}\right)$$

3.3.3 MÉTODO DE GRAM-SCHMIDT

El método de Gram-Schmidt es un algoritmo que se utiliza en álgebra lineal para obtener un conjunto de vectores ortogonales en un espacio vectorial euclídeo a partir de un conjunto de vectores linealmente independientes. El proceso se lleva a cabo mediante el siguiente procedimiento [13]:

Dado un conjunto de vectores $\{v_1, v_2, \ldots, v_n\}$ linealmente independientes y pertenecientes a un espacio vectorial euclídeo V.

3) Se define una base ortogonal $\{u_1, u_2, \ldots, u_n\}$ como la base ortogonal perteneciente al espacio vectorial del conjunto de vectores definido anteriormente y se hace $u_1 = v_1$.
4) A continuación, para todo i > 1:

$$E\ 12:\ u_i = v_i - u_1\frac{\langle v_i,u_1\rangle}{\langle u_1,u_1\rangle} - u_2\frac{\langle v_i,u_2\rangle}{\langle u_2,u_2\rangle} - \cdots - u_{n-1}\frac{\langle v_i,u_{n-1}\rangle}{\langle u_{n-1},u_{n-1}\rangle}$$

El conjunto de vectores que se obtiene después del proceso constituye una base ortogonal del espacio vectorial V formada por vectores ortogonales entre sí.

En secciones anteriores se ha mencionado que no existe un método global y claro para eliminar la información sensible que hace que los modelos sean discriminatorios hacia ciertos grupos de personas que comparten unos atributos determinados. El método de Gram-Schmidt constituye una de las técnicas utilizadas para la búsqueda de equidad en los modelos, ya que permite eliminar no solo la información explícita en las variables sensible, sino que también se encarga de eliminar la información sensible implícita en el resto de las variables.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Como ya ha sido mencionado anteriormente, el Machine Learning es un sector del que queda mucho por desarrollar y regular. La automatización de los procesos permite a las empresas tomar decisiones precisas de manera mucho más rápida manejando una gran cantidad de datos. Sin embargo, la novedad de la tecnología hace que sea necesario ampliar la relación entre ética y tecnología que se conoce hasta el momento. Es solo cuestión de tiempo que se introduzcan nuevas regulaciones que obliguen que las empresas tomen decisiones no solo guiadas por el propio beneficio de la empresa, sino que estas deban guiarse por un código ético y antidiscriminatorio.

Este trabajo es una introducción a alguna de las técnicas utilizadas para eliminar información discriminatoria o sensible de un conjunto de datos que se utilizará para entrenar un modelo con un fin predictivo que ayude en el proceso de toma de decisiones.

4.2 OBJETIVOS

El trabajo tiene como objetivos analizar e identificar si un modelo derivado de un conjunto de datos es discriminatorio respecto a un grupo de personas que comparten un atributo determinado, así como aplicar técnicas que permitan la eliminación de la información sensible que hace que el modelo sea discriminatorio, ya que el objetivo final del trabajo es obtener un modelo que permita tomar decisiones de manera justa y equitativa con la menor pérdida de precisión posible.

Para cumplir con estos objetivos será necesario analizar y evaluar la capacidad de predicción de los modelos entrenados durante el proceso de eliminación de información sensible y calcular las medidas de *fairness* que permiten analizar la diferencia en las predicciones entre grupos con atributos distintos.

En concreto en este trabajo se busca identificar si el modelo entrenado teniendo en cuenta la información recogida en las variables sensibles es discriminatorio para luego obtener un nuevo modelo basado en nuevos datos transformados mediante el método de Gram-Smidt que ofrezca predicciones más equitativas. Durante el proceso se evaluará la capacidad de predicción de los modelos entrenados.

4.3 DESCRIPCIÓN DE LOS DATOS

El conjunto de datos utilizado es un conjunto de datos público que deriva de información sobre la vivienda en el área de Boston obtenida por el gobierno de Estados Unidos. Se publicó inicialmente por Harrison, D. and Rubinfeld, D.L. en "*Hedonic prices and the demand for clean air"* y consta de 14 variables, de las cuales una será la variable objetivo sobre la que se realizarán las predicciones. El conjunto de datos contiene 506 observaciones. En la Tabla 1 se encuentran listadas las variables del modelo, así como su clasificación de acuerdo con si son variables continuas o de tipo binario. Realizar esta diferenciación es importante, ya que tanto el entrenamiento del modelo como la técnica aplicada para realizar las medidas de *fairness* dependen de ella. MEDV es la variable objetivo y se trata de una variable continua (medida en miles de dólares), por lo que en este caso el modelo será de regresión lineal. El método utilizado es el de Mínimos Cuadrados Ordinarios. Si por el contrario la variable objetivo hubiese sido de tipo binario, se habría entrenado y evaluado un modelo de regresión logística.

	Variable	**Descripción**	**Tipo**
1	CRIM	Tasa de delincuencia per cápita por ciudad	Continua
2	ZN	Proporción de suelo residencial zonificado para lotes de más de 25.000 ft^2	Continua
3	INDUS	Proporción de acres comerciales no destinados a la venta al por menor por ciudad	Continua
4	CHAS	Variable "dummy" de Charles River	Binaria

5	NOX	Concentración de ácido nítrico (partes por 10 millones)	Continua
6	RM	Número medio de habitaciones por vivienda	Continua
7	AGE	Proporción de unidades ocupadas por sus propietarios construidas antes de 1940	Continua
8	DIS	Distancias ponderadas a cinco centros de empleo de Boston	Continua
9	RAD	Índice de accesibilidad a las autopistas radiales	Continua
10	TAX	Tipo del impuesto sobre bienes inmueble de valor íntegro cada 10.000 dólares	Continua
11	PTRATIO	Ratio alumnos-profesor por ciudad	Continua
12	B	$1000*(Bk-0.63)^2$, donde Bk es la proporción de personas de raza negra por ciudad	Continua
13	LSTAT	% estatus inferior de la población	Continua
14	MEDV	Valor medio de las viviendas ocupadas por sus propietarios en miles de dólares	Continua

Tabla 1: Variables del modelo

4.4METODOLOGÍA

En este apartado se profundiza en la metodología que permite eliminar la información sensible de un conjunto de datos que ya ha sido introducida en apartados anteriores.

4.4.1 PREPROCESAMIENTO Y ANÁLISIS DESCRIPTIVO DEL CONJUNTO DE DATOS

Los primeros pasos tienen que ver con el preprocesamiento y análisis descriptivo del conjunto de datos. Después de entenderlos con un primer análisis, se determina el procedimiento a seguir. Como ha sido explicado en apartado de justificación de la metodología y técnicas aplicadas, en este primer acercamiento al conjunto de datos se

determina qué variables pueden contener información que va a hacer que el modelo discrimine hacia un grupo determinado de personas y de qué tipo son estas variables. De esta manera se establece la técnica que va a permitir el cálculo de la medida de *fairness* y qué tipo de modelo se debe entrenar.

4.4.2 ENTRENAMIENTO Y EVALUACIÓN DE LOS MODELOS

Una vez los datos han sido importados y preparados para poder ser manejados con facilidad a la hora de programar, se procederá a entrenar un primer modelo con los datos originales. Este modelo va a permitir hacer una primera predicción de la variable objetivo que va a ser utilizada para determinar si el modelo discrimina en base a las variables que se desean analizar. El modelo en este caso es de regresión lineal, por lo que será del tipo:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \cdots + \beta_n * x_n$$

Donde *y* es la variable objetivo, *xi* las variables que explican el modelo y βi los coeficientes que determinan la influencia de cada variable explicativa sobre la variable objetivo. Los coeficientes determinan cuánto varía la variable objetivo si la variable explicativa varía en una unidad. El término independiente es β_0 y el residuo u.

Python también permite obtener medidas estadísticas sobre el modelo que permiten evaluar la capacidad predictiva de este. Analizar el significado de estas variables será el siguiente paso cada vez que se entrene un modelo nuevo durante el proceso. La precisión del modelo está directamente relacionada con la calidad de las decisiones. Cuanto más preciso sea el modelo, mayor será la probabilidad de tomar la decisión que proporcione un mayor beneficio. En un modelo con un fin predictivo la medida que se utiliza para medir cómo de bueno es el modelo, es decir, la bondad del ajuste, es el coeficiente de determinación conocido como R^2. El coeficiente de determinación permite medir la proporción de variabilidad de la variable objetivo explicada por la variabilidad de las variables explicativas del modelo. Se trata de una medida comprendida entre cero y uno, y generalmente un valor de R^2 próximo a la unidad significa un modelo ajustado a los datos. Sin embargo, el valor de esta medida está directamente relacionado con el número de variables del modelo. Cuantas

más variables, más alto será el coeficiente de determinación. Esto es así aunque la variable no influya de manera significativa en la predicción. Por esta razón se utilizará otra medida conocida como R^2 corregido en el caso de que se quiera comprobar la capacidad predictiva de dos modelos con un número diferente de variables. Esta nueva medida "penaliza" al modelo cuyo número de variables es mayor. Estas medidas se determinan mediante funciones de Python.

Por otra parte, Python permite obtener otras medidas estadísticas de interés como el p-valor o el coeficiente de correlación lineal de Pearson entre dos variables. El p-valor es una medida importante en la evaluación de modelos explicativos. Permite identificar qué variables influencian de manera significativa la predicción de la variable objetivo. Por otro lado, el coeficiente de correlación lineal es una medida de la dependencia lineal entre dos variables de interés. Su valor se encuentra comprendido entre cero y uno. Cuando el coeficiente de correlación lineal es igual a cero las variables no son dependientes linealmente, y si por el contrario el coeficiente es próximo a la unidad, las variables tienen una alta dependencia lineal. En este trabajo se medirá explícitamente la correlación que existe entre las variable que sospechamos contienen información sensible y las predicciones obtenidas con el modelo entrenado.

4.4.3 MEDIDA DE FAIRNESS

Una vez entrenado el modelo y tras haber analizado su capacidad predictiva, se pasará a medir la divergencia de predicciones por variables sensibles. En este paso se determinará si el modelo está discriminando a algún grupo con un atributo determinado. Para ello, será necesario dividir el conjunto de datos en tramos antes de poder aplicar la siguiente fórmula:

$$fairness = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N0} * \sum_{i=0}^{N0} \hat{y}i\,|$$

Donde N1 es el número de personas que comparten un atributo determinado, N0 el número de personas que no comparten ese atributo, e $\hat{y}i$ la predicción dada por el modelo para un

índice *i* determinado. Esta fórmula es aplicable si las variables sensibles son continuas. En el caso de que fuesen variables de tipo binario el proceso sería diferente.

Para poder dividir los datos en tramos será de gran ayuda obtener un histograma que muestre cómo se distribuyen los valores de las predicciones de la variable objetivo en base a los valores de las variables sensibles. Obtenidos los tramos, será posible calcular las medidas de *fairness* y determinar si el modelo es discriminatorio o no.

No existe una medida concreta a partir de la cual un modelo pase a considerarse discriminatorio. Idealmente, si el modelo no contiene información sensible, el valor que toma la medida de *fairness* es cero, ya que la media de las predicciones para los tramos en los que se ha dividido el conjunto de datos sería la misma. En el momento de la evaluación del modelo que se quiere analizar en este trabajo se determinará si el modelo discrimina teniendo en cuenta que la medida que mide la equidad respecto a una variable de un modelo no sesgado es próxima a cero.

4.4.4 APLICACIÓN DE TÉCNICAS DE REDUCCIÓN DE UNFAIRNESS. MÉTODO DE GRAM-SCHMIDT

En el caso de que la conclusión sea que el modelo no es equitativo, se procederá a entrenar el modelo de nuevo, pero esta vez sin tener en cuenta la información sensible. En muchas ocasiones no basta con eliminar la variable que discrimina de manera directa, ya que la información también puede estar contenida de manera implícita en otras variables. El procedimiento es el mismo: entrenamiento del modelo, evaluación de su capacidad predictiva y posterior medida de *fairness*, esta vez con nuevos valores para las predicciones, pero respetando los tramos. La situación esperada después de realizar este proceso es que, debido a la información sensible recogida de manera implícita en otras variables, el modelo siga discriminando. Es en este momento cuando se recurre al método de Gram-Schmidt.

Aplicando el método de Gram-Schmidt al conjunto de datos sobre los que se realiza este trabajo se obtiene un conjunto de datos transformado que en principio no discrimina por las variables que se han considerado variables sensibles. A continuación, se volverá a entrenar

el modelo con los datos transformados y se volverán a sacar las métricas que ayudarán a determinar si el modelo ha perdido precisión a la hora de predecir. Es importante recalcar que se debe llegar a un equilibrio entre precisión y equidad, ya que estos algoritmos juegan un papel importante en la toma de decisiones que aportan beneficios a nivel empresarial.

Una vez entrenado el nuevo modelo y tras realizar un análisis de su capacidad de predicción, se procede a evaluar las nuevas medidas de *fairness* de la misma manera que se ha hecho en los modelos anteriores.

Capítulo 5. INTERPRETACIÓN DE RESULTADOS

Siguiendo la metodología explicada en el apartado 4.3, en esta sección se mostrarán y explicarán los resultados obtenidos tras la realización del trabajo, en el que se pretende encontrar un modelo no discriminatorio y lo más preciso posible capaz de predecir el valor medio de las viviendas ocupadas por sus propietarios ("MEDV").

5.1 ENTRENAMIENTO DEL PRIMER MODELO Y EVALUACIÓN DE SU CAPACIDAD PREDICTIVA

Tras analizar las diferentes variables que conformarán el modelo y determinar cuáles son las que podrían contener información sensible, se procede a importar y preparar los datos. Como ya ha sido explicado anteriormente, en este trabajo las variables que podrían ser discriminatorias son las que hacen referencia a la raza y al estatus de las personas que forman parte de la muestra. Estas variables son "B" y "LSTAT".

En el siguiente paso, se entrena el primer modelo y se evalúa su capacidad predictiva. Una vez entrenado el modelo se obtienen los siguientes datos acerca de los coeficientes que acompañan a las variables explicativas, el término independiente y otras medidas estadísticas de interés como el coeficiente de determinación R^2:

OLS Regression Results

Dep. Variable:	y	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Wed, 12 Apr 2023	Prob (F-statistic):	6.72e-135
Time:	16:01:46	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	36.4595	5.103	7.144	0.000	26.432	46.487
CRIM	-0.1080	0.033	-3.287	0.001	-0.173	-0.043
ZN	0.0464	0.014	3.382	0.001	0.019	0.073
INDUS	0.0206	0.061	0.334	0.738	-0.100	0.141
CHAS	2.6867	0.862	3.118	0.002	0.994	4.380
NOX	-17.7666	3.820	-4.651	0.000	-25.272	-10.262
RM	3.8099	0.418	9.116	0.000	2.989	4.631
AGE	0.0007	0.013	0.052	0.958	-0.025	0.027
DIS	-1.4756	0.199	-7.398	0.000	-1.867	-1.084
RAD	0.3060	0.066	4.613	0.000	0.176	0.436
TAX	-0.0123	0.004	-3.280	0.001	-0.020	-0.005
PTRATIO	-0.9527	0.131	-7.283	0.000	-1.210	-0.696
B	0.0093	0.003	3.467	0.001	0.004	0.015
LSTAT	-0.5248	0.051	-10.347	0.000	-0.624	-0.425

Omnibus:	178.041	Durbin-Watson:	1.078
Prob(Omnibus):	0.000	Jarque-Bera (JB):	783.126
Skew:	1.521	Prob(JB):	8.84e-171
Kurtosis:	8.281	Cond. No.	1.51e+04

Ilustración 3: Modelo 1

En este caso, el valor del coeficiente de determinación es de 0,741. Esto significa que el modelo entrenado explica más del 70% de la variable objetivo, es decir, del valor medio de las viviendas ocupadas por sus propietarios. El valor de R^2 corregido es de 0,734. Este valor es el que se utilizará para comparar la precisión del resto de modelos. Al finalizar el trabajo se debe evaluar la pérdida de precisión que presentará el modelo que no discrimina, determinando cuánto difiere el coeficiente de determinación del último modelo de esta primera medida.

Aunque este trabajo se centra en la capacidad predictiva del modelo, podría resultar útil hacer un análisis explicativo de las variables de interés. En primer lugar, si se tiene en cuenta la variable "B", se puede observar que el coeficiente que la acompaña tiene un valor de 0.0093. Además, al ser de signo positivo, la correlación entre la predicción de la variable objetivo y la variable "B" es positiva. Esto significa que, si el valor de la variable "B" aumenta en una unidad, el valor medio de la vivienda ocupada por sus propietarios aumentará de media un 0.0093. En cambio, el coeficiente que acompaña a la variable "LSTAT" tiene un valor de -0.5258, lo que significa que existe una correlación negativa entre la predicción y la variable.

En el párrafo anterior se ha hablado de correlación entre dos variables. Sin embargo, para saber si este tipo de correlación es lineal es necesario obtener el coeficiente de correlación lineal de Pearson. Haciendo el cálculo en Python se obtiene que el coeficiente de correlación lineal entre "B" y "MEDV" es de 0,3875 y entre "LSTAT" y "MEDV" es de -0,8571.

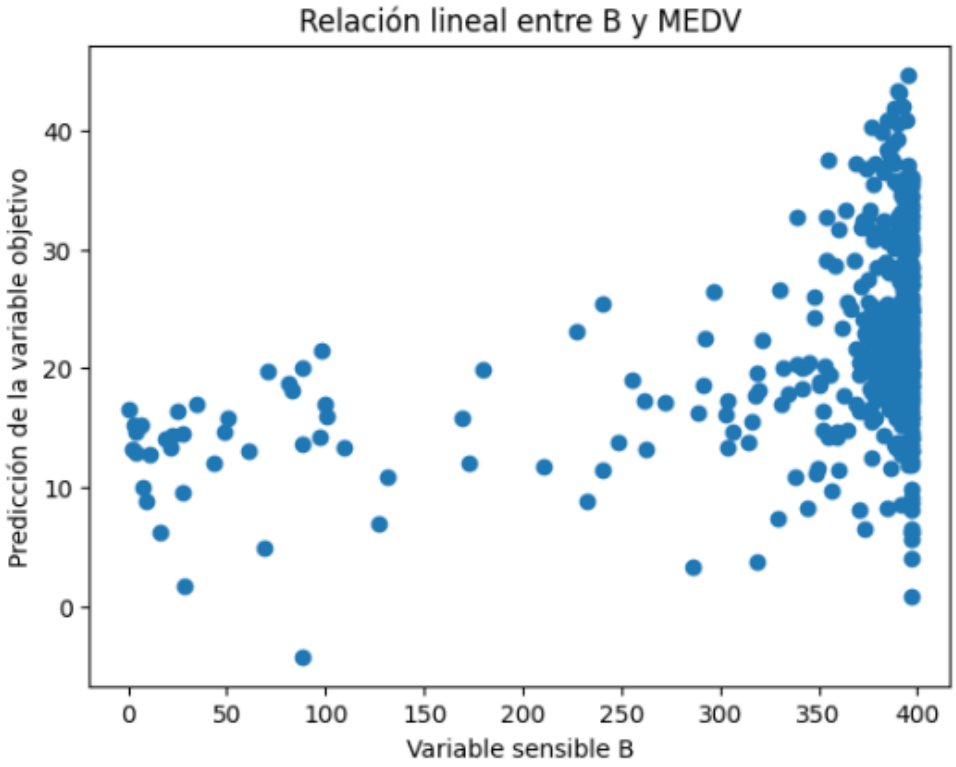


Ilustración 4: Scatter plot entre las variables B y MEDV

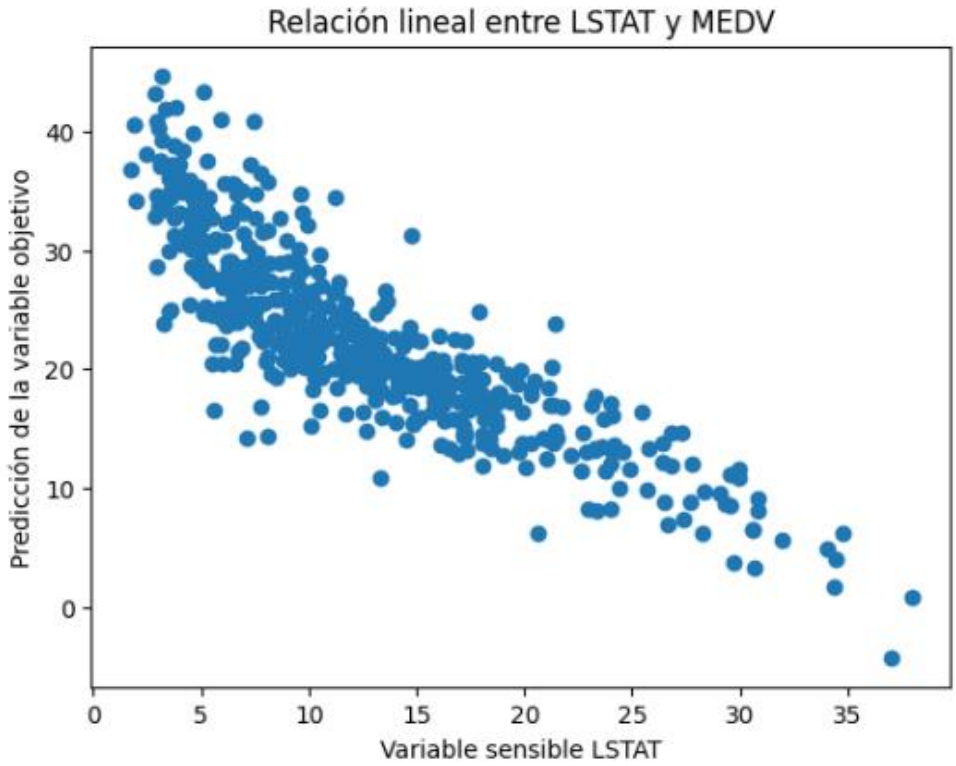


Ilustración 5: Scatter plot entre las variables LSTAT y MEDV

Las ilustraciones 4 y 5 representan la relación lineal que existe entre las variables sensibles y la predicción de la variable objetivo. Como ya se había intuido, existe una relación lineal positiva entre la primera variable sensible y la variable endógena y una relación lineal negativa y pronunciada entre la segunda variable sensible y la variable objetivo.

5.2 MEDIDA DE FAIRNESS DEL PRIMER MODELO

Una vez obtenidas las medidas estadísticas que permiten entender el modelo y evaluar su capacidad de predicción se procede a realizar el cálculo de *fairness*. Para realizar la separación en los tramos que permitirán comparar la media de las predicciones entre los diferentes grupos se obtienen los siguientes histogramas que proporcionan una imagen visual del valor que toman los datos contenidos en las dos variables sensibles:

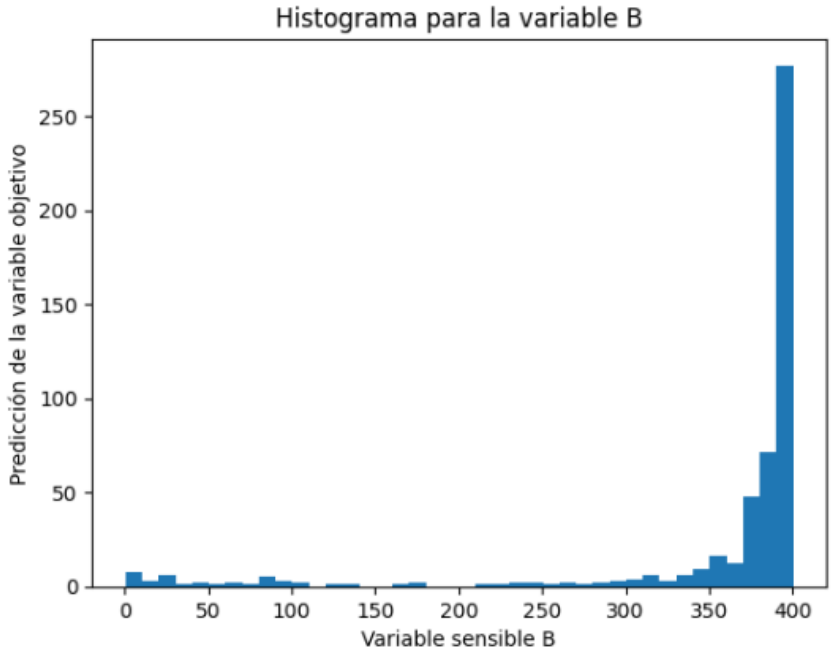


Ilustración 6: Histograma para la variable B

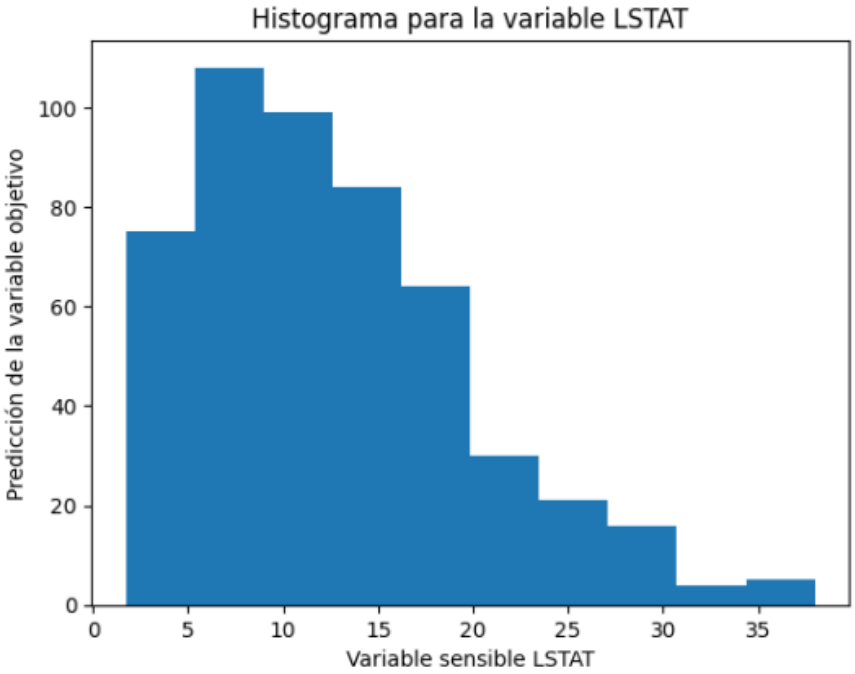


Ilustración 7: Histograma para la variable LSTAT

En el histograma que representa los valores que toma la variable que recoge la información sobre la raza de las personas de la muestra (Ilustración 6), se puede observar que la mayoría de las observaciones toman un valor próximo a 400. Es por este motivo por el que se ha decidido agrupar las observaciones en rangos de 10. Tras analizar esta distribución los rangos que se han utilizado para el cálculo de la medida de *fairness* son: [0, 100], [101, 300] y [301, 400]. El tramo 1 contiene 31 observaciones, el tramo 2 contiene 23 observaciones y en el tramo 3 se acumula la mayoría de las observaciones, siendo estas un total de 452.

En el apartado en el que se ha explicado la metodología seguida para realizar el trabajo se ha introducido la fórmula que ha de utilizarse a la hora de medir el *fairness* en una regresión lineal con variables continuas. En esta ocasión se ha tomado la decisión de dividir los datos en tres tramos. Por este motivo se compararán las medias de las predicciones de cada tramo con la suma de las medias de las predicciones de los otros dos tramos. De esta manera se obtienen tres medidas diferentes. El procedimiento seguido es el siguiente:

$$fairness\ (B,1) = \left|\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * \left(\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i\right)\right| = 9{,}9689$$

$$fairness\ (B,2) = \left|\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * \left(\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i\right)\right| = 7{,}2169$$

$$fairness\ (B,3) = \left|\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * \left(\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i\right)\right| = 9{,}2988$$

Donde N1 es el número de observaciones pertenecientes al tramo 1, N2 el número de observaciones del tramo 2, N3 el número de observaciones que contiene el tramo 3 e $\hat{y}i$ la predicción de la variable objetivo dada por el modelo para una observación *i* determinada. También se han incluido las medidas obtenidas. Idealmente, si el modelo no discrimina, las medidas obtenidas deberían aproximarse a cero y es fácil observar que no es así. Es por este motivo por el que se puede concluir que el modelo es discriminatorio por la variable "B".

En cuanto a la variable que recoge la información sobre el estatus, en la Ilustración 7 se puede observar que los datos se distribuyen de una manera más homogénea por todo el rango de valores de la variable "LSTAT". En este caso las observaciones se agrupan en rangos de 5 valores. De la misma manera que se ha hecho para la variable anterior, se ha dividido la información en tres tramos: [0, 9], [10, 19] y [20, 37,97]. En el caso de la segunda variable sensible, el tramo 1 contiene 219 observaciones, el tramo 2 está compuesto por 213 observaciones y el tramo 3 contiene 74 observaciones.

De la misma manera que se ha hecho para la primera variable analizada, se obtienen las siguientes medidas para la variable "LSTAT":

$$fairness\ (LSTAT, 1) = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 11{,}2278$$

$$fairness\ (LSTAT, 2) = |\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * (\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 4{,}6875$$

$$fairness\ (LSTAT, 3) = |\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i)\ | = 12{,}9240$$

Donde N1 es el número de observaciones pertenecientes al tramo 1, N2 el número de observaciones del tramo 2, N3 el número de observaciones que contiene el tramo 3 e $\hat{y}i$ la predicción de la variable objetivo dada por el modelo para una observación *i* determinada. En este caso ocurre lo mismo que con la variable analizada anteriormente, las medidas son significativamente superiores a cero.

5.3 ENTRENAMIENTO Y MEDIDA DE FAIRNESS DEL MODELO SIN LA INFORMACIÓN SENSIBLE

Una vez se ha determinado que el primer modelo entrenado discrimina por las variables que podrían causar un problema ético, una manera de intentar eliminar el sesgo podría ser

eliminando esas variables del conjunto de datos. Un problema que presenta esta táctica es que, en numerosas ocasiones, la información sensible se encuentra implícita en otras variables. La manera de comprobar si este es el caso es entrenando un nuevo modelo sin tener en cuenta las variables "B" y "LSTAT" y volver a calcular las medidas que permiten identificar si el modelo no es equitativo.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.670
Model:	OLS	Adj. R-squared:	0.663
Method:	Least Squares	F-statistic:	91.31
Date:	Mon, 01 May 2023	Prob (F-statistic):	1.83e-111
Time:	22:11:21	Log-Likelihood:	-1559.5
No. Observations:	506	AIC:	3143.
Df Residuals:	494	BIC:	3194.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	27.1524	5.291	5.132	0.000	16.758	37.547
CRIM	-0.1840	0.036	-5.089	0.000	-0.255	-0.113
ZN	0.0391	0.015	2.535	0.012	0.009	0.069
INDUS	-0.0423	0.069	-0.614	0.539	-0.178	0.093
CHAS	3.4875	0.966	3.611	0.000	1.590	5.385
NOX	-22.1821	4.272	-5.193	0.000	-30.575	-13.790
RM	6.0757	0.397	15.298	0.000	5.295	6.856
AGE	-0.0452	0.014	-3.234	0.001	-0.073	-0.018
DIS	-1.5839	0.224	-7.066	0.000	-2.024	-1.143
RAD	0.2547	0.074	3.425	0.001	0.109	0.401
TAX	-0.0122	0.004	-2.887	0.004	-0.021	-0.004
PTRATIO	-0.9962	0.147	-6.777	0.000	-1.285	-0.707

Omnibus:	260.309	Durbin-Watson:	0.938
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2458.852
Skew:	2.039	Prob(JB):	0.00
Kurtosis:	13.000	Cond. No.	1.16e+04

Ilustración 7: Modelo 2

Una vez entrenado el modelo se pueden volver a obtener las medidas estadísticas por las que se puede determinar si ha habido una pérdida en la capacidad de predicción del modelo. En la Ilustración 8, de nuevo obtenida por medio de Python, aparecen reflejados los nuevos valores para el coeficiente de determinación. En secciones previas se ha mencionado que la medida que interesa analizar para evaluar modelos anidados es la del R^2 corregido. En este modelo esta medida toma un valor de 0,663, inferior a la que se conseguía con el modelo

anterior. Es habitual que modelos con un mayor número de variables tengan una mayor capacidad de predicción, siempre y cuando las variables sean significativas del modelo. En este caso, al haber eliminado la información correspondiente a la de las dos variables sensibles para entrenar el modelo, no resulta raro que su capacidad de predicción sea menor.

A continuación, se procede a calcular las nuevas medidas de *fairness* con las nuevas predicciones y manteniendo los mismos tramos que en el apartado anterior.

Para la primera variable se obtienen las siguientes medidas:

$$fairness\ 2\ (B,1) = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 6{,}0742$$

$$fairness\ 2\ (B,2) = |\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * (\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 6{,}2917$$

$$fairness\ 2\ (B,3) = |\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i)\ | = 6{,}5281$$

En este caso los valores de N1, N2 y N3 no cambian con respecto a las medidas calculadas anteriormente. Los valores de $\hat{y}i$ corresponden a los nuevos valores de las predicciones obtenidas por medio del nuevo modelo. Para esta primera variable las medidas obtenidas con el nuevo modelo son inferiores a las obtenidas en la sección anterior. El modelo entrenado sin tener en cuenta las dos variables sensibles es más equitativo que el modelo original. Sin embargo, se observa que las medidas siguen siendo superiores a cero.

Siguiendo el mismo procedimiento para la segunda variable sensible se obtienen las siguientes medidas:

$$fairness\ 2\ (LSTAT,1) = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 9{,}8218$$

$$fairness\ 2\ (LSTAT, 2) = |\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * (\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 5{,}1816$$

$$fairness\ 2\ (LSTAT, 3) = |\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i)\ | = 9{,}1951$$

En el caso de la segunda variable, la situación es muy similar. Manteniendo el tamaño de los tramos y con nuevos valores para las predicciones, se obtienen valores más próximos a cero, pero lo suficientemente elevados como para considerar que el modelo continúa discriminando por la variable "LSTAT".

5.4 APLICACIÓN DEL MÉTODO DE GRAM-SCHMIDT. ENTRENAMIENTO Y EVALUACIÓN DEL TERCER MODELO

En el apartado anterior se ha concluido que existe información sensible implícita en otras variables, de manera que las predicciones obtenidas con el modelo entrenado sin tener en cuenta las variables B y LSTAT sigue siendo discriminatorio. Aplicando el método de Gram-Schmidt se consigue eliminar la información sensible del conjunto de datos.

Existe una función en Python que utiliza el método de Gram-Schmidt para crear un nuevo conjunto de datos transformado a partir de los datos originales. Esta función es la función "InformationFilter". Tras aplicar este método, se puede entrenar un nuevo modelo a partir del conjunto de datos transformado. Hay que recordar que el precio a pagar por obtener un modelo que no discrimina es una pérdida de precisión en las predicciones, por lo que resulta importante obtener el valor del coeficiente de determinación que permitirá comparar todos los modelos en términos de precisión.

El valor de R^2, así como el de los coeficientes que acompañan a las variables en el modelo y otras medidas estadísticas de interés se muestran a continuación:

OLS Regression Results

Dep. Variable:	y	R-squared:	0.223
Model:	OLS	Adj. R-squared:	0.205
Method:	Least Squares	F-statistic:	12.87
Date:	Wed, 14 Jun 2023	Prob (F-statistic):	1.37e-21
Time:	08:56:06	Log-Likelihood:	-1776.5
No. Observations:	506	AIC:	3577.
Df Residuals:	494	BIC:	3628.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	22.2283	0.371	59.950	0.000	21.500	22.957
CRIM	-0.1021	0.057	-1.802	0.072	-0.213	0.009
ZN	0.0473	0.024	1.996	0.047	0.001	0.094
INDUS	0.0109	0.106	0.103	0.918	-0.197	0.219
CHAS	2.7520	1.488	1.850	0.065	-0.172	5.676
NOX	-11.9514	5.531	-2.161	0.031	-22.819	-1.084
RM	4.6367	0.509	9.104	0.000	3.636	5.637
AGE	-0.0024	0.023	-0.106	0.915	-0.047	0.042
DIS	-1.2777	0.322	-3.967	0.000	-1.910	-0.645
RAD	0.2534	0.110	2.306	0.022	0.037	0.469
TAX	-0.0112	0.006	-1.732	0.084	-0.024	0.002
PTRATIO	-0.7339	0.181	-4.056	0.000	-1.089	-0.378

Omnibus:	18.965	Durbin-Watson:	0.552
Prob(Omnibus):	0.000	Jarque-Bera (JB):	38.988
Skew:	0.173	Prob(JB):	3.42e-09
Kurtosis:	4.315	Cond. No.	2.44e+03

Ilustración 8: Modelo 3

Observando el valor del coeficiente de determinación de este nuevo modelo es fácil concluir que ha habido una pérdida significativa en la capacidad predictiva del modelo. El valor del coeficiente de determinación es de 0,223 en comparación con el valor de 0,741 del modelo original.

Para las medidas de *fairness* obtenidas con este modelo se obtienen los siguientes valores:

$$fairness\ 3\ (B,1) = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 5{,}0272$$

$$fairness\ 3\ (B,2) = |\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * (\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 0{,}7301$$

$$fairness\ 3\ (B,3) = |\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i)\ | = 3{,}3651$$

$$fairness\ 3\ (LSTAT,1) = |\frac{1}{N1} * \sum_{i=0}^{N1} \hat{y}i - \frac{1}{N2+N3} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 1{,}5909$$

$$fairness\ 3\ (LSTAT,2) = |\frac{1}{N2} * \sum_{i=0}^{N2} \hat{y}i - \frac{1}{N1+N3} * (\sum_{i=0}^{N1} \hat{y}i + \sum_{i=0}^{N3} \hat{y}i)\ | = 1{,}8662$$

$$fairness\ 3\ (LSTAT,3) = |\frac{1}{N3} * \sum_{i=0}^{N3} \hat{y}i - \frac{1}{N2+N1} * (\sum_{i=0}^{N2} \hat{y}i + \sum_{i=0}^{N1} \hat{y}i)\ | = 0{,}5154$$

Para las dos variables sensibles las medidas de *fairness* toman valores inferiores a los que se obtuvieron con los modelos anteriores, lo que indica que el modelo entrenado con el conjunto de datos transformado es el menos discriminatorio.

Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

Esta sección tiene como objetivo destacar las conclusiones más importantes a las que se han llegado tras realizar el trabajo.

La primera conclusión tiene que ver con la eficacia de las técnicas de reducción de *unfairness* utilizadas. Mirando a las medidas de equidad que se han obtenido para los tres modelos, se concluye que el modelo entrenado tras aplicar el método de Gram-Schmidt es el menos discriminatorio, ya que ha logrado eliminar también la información sensible implícita en las variables que no se consideraban problemáticas.

Como segunda conclusión, es importante destacar que, aunque se haya logrado obtener un modelo menos discriminatorio, se ha producido una pérdida en la capacidad predictiva del modelo. El valor del coeficiente de determinación del modelo original es de 0,741, después de eliminar las variables sensibles se obtiene un valor de 0,670, y tras aplicar el método de Gram-Schmidt el valor del R^2 cae hasta 0,223. Esto indica que las variables sensibles son predictivas del modelo.

Finalmente, como apunte final, si se observan las medidas de *fairness* obtenidas finalmente en el modelo que no discrimina, se aprecia que la diferencia de medias de las predicciones es más próxima a cero que en los casos anteriores, pero los valores difieren entre ellos. Recordando la finalidad del trabajo, se quiere predecir el valor medio de las viviendas ocupadas por sus propietarios, mediante un modelo que no discrimine por las variables "B" (que contiene la proporción de personas de raza negra por ciudad) y "LSTAT" (que representa el porcentaje de personas de estatus socioeconómico inferior).

- Para la variable "B" se han obtenido finalmente tres medidas de *fairness* que implican una mejora significativa respecto de las medidas iniciales. Sin embargo, las medidas difieren entre ellas más que para la segunda variable sensible. La medida que

compara las medias de las predicciones obtenidas para el tramo 1 con las medias de las predicciones obtenidas para el resto de los tramos toma un valor de 5,0272. Si se compara esta medida con las dos restantes, con valores de 0,7301 y 3,3651, se puede concluir que sigue habiendo una diferencia entre las predicciones obtenidas para el primer tramo y el resto de las predicciones, siendo este tramo el correspondiente a las observaciones con una menor proporción de personas de raza negra.

- Para la variable "LSTAT" las medidas son más similares y próximas a cero, siendo la medida que compara el tercer tramo con el resto de los tramos la que toma un valor más bajo. Este es el tramo que representa las observaciones con un porcentaje más alto de personas de estatus socioeconómico inferior.

6.2 TRABAJOS FUTUROS

Una de las conclusiones a las que se ha llegado tras realizar el trabajo, es que, si se elimina información sensible de los modelos y esta resulta ser predictiva, entonces se produce una pérdida en la capacidad de predicción del modelo en cuestión. En esta última sección se pretende ofrecer una serie de posibles trabajos futuros que permitan mejorar la capacidad predictiva de los modelos.

- Estimación real del impacto que tiene la reducción de predictibilidad en contextos específicos como la contratación, la toma de decisiones crediticias o la selección de contenido en plataformas en línea. La pérdida de precisión en las predicciones puede suponer que se tomen decisiones menos sesgadas. Por ejemplo, en el caso de la contratación, puede obtenerse una plantilla con una mayor diversidad si se sacan de la ecuación datos sobre el sexo o la raza de las personas.

- Aplicar técnicas de privacidad diferencial como el añadir ruido aleatorio a las variables de forma que se pierda la relación con el valor original, pero se mantengan las propiedades estadísticas. Esta técnica se utiliza para mantener la privacidad de las

personas de la muestra. Al añadir ruido de manera aleatoria a los datos, los valores individuales cambian, pero las características globales se mantienen.

Generación de datos sintéticos para intentar recuperar la predictibilidad que se pierde añadiendo más variables al conjunto de datos. La idea es utilizar algoritmos capaces de captar las relaciones entre los datos para crear de manera artificial otros sintéticos que sigan una distribución lo más parecida posible. De esta manera, dependiendo de la calidad y precisión de los algoritmos utilizados, se podrá recuperar parte de la capacidad predictiva que se ha perdido.

Capítulo 7. BIBLIOGRAFÍA

[1] *¿Qué es el aprendizaje supervisado?* (s.f.). (TIBCO) Recuperado el 10 de junio de 2023, de https://www.tibco.com/es/reference-center/what-is-supervised-learning

[2] Bilal Zafal, M., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017). Fairness Constraints: Mechanisms for Fair Classification. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research.* Germany. Obtenido de https://proceedings.mlr.press/v54/zafar17a.html

[3] Calders, T., Karim, A., Kamiran, F., & Zhang, X. (2013). Controlling Attribute Effect in Linear Regression. *2013 IEEE 13th International Conference on Data Mining.* Dallas. doi:10.1109/ICDM.2013.114

[4] Chitarroni, H. (2022). *La regresión logística.* Buenos Aires: IDICSO. Obtenido de https://racimo.usal.edu.ar/83/1/Chitarroni17.pdf

[5] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zennel, R. (2011). *Fairness Through Awareness.* ITCSC. Obtenido de https://arxiv.org/pdf/1104.3913.pdf

[6] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). *Certifying and removing disparate impact.* Cornell University. Obtenido de https://arxiv.org/pdf/1412.3756.pdf

[7] Fiallos, G. (2021). La Correlación de Pearson y el proceso de regresión por el Método de Mínimos Cuadrados. *Ciencia Latina, 5*(3), 3-18. Obtenido de https://ciencialatina.org/index.php/cienciala/article/view/466/573

[8] Hardt, M., Price, E., & Sebero, N. (2016). *Equality of Opportunity in Supervised Learning.* Cornell University, Advances in Neutral Information Processing Systems 29 (NIPS 2016). Obtenido de https://arxiv.org/pdf/1610.02413.pdf

[9] IBM. (s.f.). *¿Qué es Machine Learning?* Recuperado el 15 de mayo de 2023, de https://www.ibm.com/mx-es/analytics/machine-learning

[10] ICHI.PRO. (s.f.). *Medir la equidad en los modelos de aprendizaje automático*. Recuperado el 16 de junio de 2023, de https://ichi.pro/es/medir-la-equidad-en-los-modelos-de-aprendizaje-automatico-48242968147730

[11] Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2011). *Fairness-Aware Classifier with Prejudice Remover Regulizer*. Japan: National Institute of Advanced Industrial Science and Technology, University of Tsukuba. Obtenido de https://link.springer.com/content/pdf/10.1007/978-3-642-33486-3_3.pdf

[12] Maisueche Cuadrado, A. (septiembre de 2019). *Utilización del Machine Learning en la Industria 4.0* [Trabajo de Fin de Máster]. Valladolid: Universidad de Valladolid. Obtenido de https://uvadoc.uva.es/bitstream/handle/10324/37908/TFM-I-1372.pdf?sequence=1&isAllowed=y

[13] Marocho, P. (2010). *Proceso de Gram Schmidt*. (SlideShare) Recuperado el 23 de marzo de 2023, de https://www.slideshare.net/paolamorochoa/proceso-de-gram-schmidt

[14] Marr, B. (19 de febrero de 2016). A Short History of Machine Learning Every Manager Should Read. *Forbes*. Obtenido de https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=45bc222b15e7

[15] Martínez de Ibarreta Zurita, C., Álvarez Fernández, C., Borrás Palá, F., Budría Rodríguez, S., Curto González, T., & Escobar Torres, L. S. (2021). *Modelos Cuantitativos para la Economía y la Empresa en 101 ejemplos*. Madrid, España: EV Services.

[16] Molinero, L. M. (2003). *¿Qué es el método de estimación de máxima verosimilitud y cómo se interpreta?* Alce Ingeniería. Obtenido de https://www.alceingenieria.net/bioestadistica/maxverosim.pdf

[17] Mooler0410. (s.f.). *The Practical Guides for Large Language Models*. (GitHub) Recuperado el 30 de mayo de 2023

[18] Regramento (UE) 2021/0106 del Parlamento Europeo y del Consejo, de 21 de abril de 2021, por el que se establecen normas armonizadas en materia de Inteligencia Artificial (Ley de Inteligencia Artificial) y se modifican determinados actos legislativos de la Unión. (2021). *Diario Oficial de la Unión Europea*. COM/2021/206 final. https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:52021PC0206

[19] *Regresión Lineal*. (s.f.). (Probabilidad y Estadística) Obtenido de https://www.probabilidadyestadistica.net/regresion-lineal/#regresion-lineal-multipl

[20] Sabán, A. (8 de mayo de 2016). ¿Qué es la singularidad tecnológica y qué supondría para el ser humano? *Hipertextual*. Obtenido de https://hipertextual.com/2016/05/singularidad-tecnologica

[21] Starmer, J. [StatQuest with Josh Starmer]. (2017). *Linear Regression and Linear Models* [Lista de reproducción]. YouTube. Obtenido de https://www.youtube.com/watch?v=PaFPbb66DxQ&list=PLblh5JKOoLUIzaEkCLIUxQFjPIlapw8nU

[22] Starmer, J. [StatQuest with Josh Starmer]. (2018). *Logistic Regression* [Lista de reproducción]. YouTube. Obtenido de https://www.youtube.com/watch?v=yIYKR4sgzI8&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe

Capítulo 8. ANEXOS

8.1 ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

Como ya se ha mencionado en los apartados anteriores, el objetivo del trabajo es evitar la discriminación en los procesos automáticos de toma de decisiones. Existe una alta capacidad de mejora en este sector, ya que se trata de un sector relativamente nuevo. Mediante la aplicación de técnicas de cálculo de *fairness* se podrá determinar si grupos con un determinado atributo están siendo tratados de una manera no equitativa. De esta manera, será posible reformular el modelo para que las futuras predicciones sean más justas y equitativas. Esto se encuadra dentro de dos de los objetivos de la lista de ODS, en concreto, la reducción de desigualdades y el objetivo de igualdad de género.

8.2 CÓDIGO

```
[ ] # Importar fichero guardado previamente en Google Drive
    from google.colab import drive
    import pandas as pd
    drive.mount('/content/gdrive',force_remount=True) # esto solo lo tendrás que ejecutar una vez
    raw_df=pd.read_csv('/content/gdrive/MyDrive/TFG-Beatriz/data/boston_dataset.csv',sep=";",header=None)

    # Nombrar las columnas para que el código sea más legible
    raw_df = raw_df.rename(columns = {0:'CRIM', 1:'ZN', 2: 'INDUS', 3:'CHAS', 4:'NOX', 5: 'RM', 6:'AGE',7:'DIS',8:'RAD',9:'TAX',10:'PTRATIO',11:'B',12:'LSTAT',13:'MEDV'})
    raw_df.head()
```

```
Mounted at /content/gdrive
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

```
[ ] # Instalar sklego. Aquí se encuentra la función que permitirá aplicar el método de Gram-Schmidt
    %pip install sklego
```

```
[ ] #Importar el resto de librerías necesarias
    import numpy as np
    import matplotlib.pylab as plt
    import statsmodels.api as sm
    import sklego

    from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_squared_error, r2_score

    from sklego.preprocessing import InformationFilter
```

```
[ ] #Definir las variables independientes y la variable objetivo
    data = raw_df.drop(columns=['MEDV'],axis=1)
    target = raw_df[['MEDV']].values
```

ENTRENAR MODELO Y OBTENER REGRESIÓN

```
[ ] X=data
    y=target
```

```
[ ] #Entrenar modelo de regresión lineal
    modelo = LinearRegression()
    modelo.fit(X,y)
```

```
▾ LinearRegression
LinearRegression()
```

```
[ ] #predicción
    y_pred = modelo.predict(X)
```

```
[ ] # Coeficientes
    print('Coeficientes: \n', modelo.coef_)
    # Término independiente
    print('Termino independiente: \n', modelo.intercept_)
```

```
Coeficientes:
 [[-1.08011358e-01  4.64204584e-02  2.05586264e-02  2.68673382e+00
  -1.77666112e+01  3.80986521e+00  6.92224640e-04 -1.47556685e+00
   3.06049479e-01 -1.23345939e-02 -9.52747232e-01  9.31168327e-03
  -5.24758378e-01]]
Termino independiente:
 [36.45948839]
```

```
[ ] #r^2
    r_sq = modelo.score(X,y)
    print(f"coefficient of determination: {r_sq}")
```

```
coefficient of determination: 0.7406426641094095
```

```
[ ] #medidas estadísticas de interés sobre el modelo. Incluye valor de los parámetros y coeficiente de determinación
    x=X
    x=sm.add_constant(x)
    model=sm.OLS(y,x) #CUIDADO, en este caso output va primero
    results=model.fit()
    print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.734
Method:                 Least Squares   F-statistic:                     108.1
Date:                Wed, 14 Jun 2023   Prob (F-statistic):          6.72e-135
Time:                        08:55:09   Log-Likelihood:                -1498.8
No. Observations:                 506   AIC:                             3026.
Df Residuals:                     492   BIC:                             3085.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         36.4595      5.103      7.144      0.000      26.432      46.487
CRIM          -0.1080      0.033     -3.287      0.001      -0.173      -0.043
ZN             0.0464      0.014      3.382      0.001       0.019       0.073
INDUS          0.0206      0.061      0.334      0.738      -0.100       0.141
CHAS           2.6867      0.862      3.118      0.002       0.994       4.380
NOX          -17.7666      3.820     -4.651      0.000     -25.272     -10.262
RM             3.8099      0.418      9.116      0.000       2.989       4.631
AGE            0.0007      0.013      0.052      0.958      -0.025       0.027
DIS           -1.4756      0.199     -7.398      0.000      -1.867      -1.084
RAD            0.3060      0.066      4.613      0.000       0.176       0.436
TAX           -0.0123      0.004     -3.280      0.001      -0.020      -0.005
PTRATIO       -0.9527      0.131     -7.283      0.000      -1.210      -0.696
B              0.0093      0.003      3.467      0.001       0.004       0.015
LSTAT         -0.5248      0.051    -10.347      0.000      -0.624      -0.425
==============================================================================
Omnibus:                      178.041   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              783.126
Skew:                           1.521   Prob(JB):                    8.84e-171
Kurtosis:                       8.281   Cond. No.                     1.51e+04
==============================================================================
```

CORRELACIÓN LINEAL

```
[ ] data_pd=pd.DataFrame(data)
    data_pd.corr() #matriz de correlación entre todas las variables
```

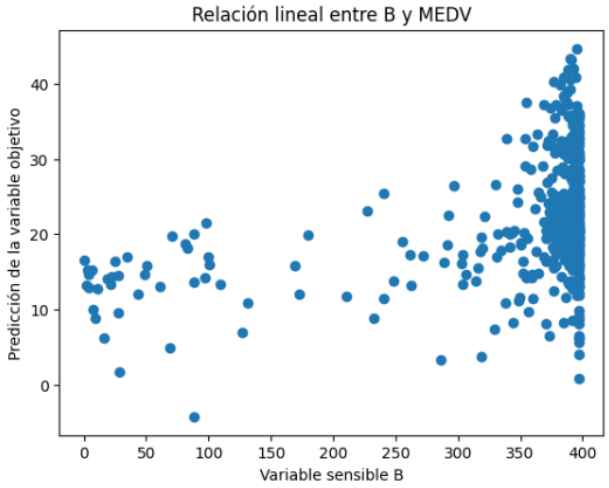
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.379670	0.625505	0.582764	0.289946	-0.385064	0.455621
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000

```
[ ] #correlación lineal entre las variables sensibles y la predicción de la variable objetivo
    y_pred_flat = y_pred.flatten()

    coef_B = np.corrcoef(data['B'].values,y_pred_flat)
    coef_LSTAT = np.corrcoef(data['LSTAT'].values,y_pred_flat)

    print(coef_B)
    print(coef_LSTAT)
```

```
[[1.         0.38747211]
 [0.38747211 1.        ]]
[[ 1.         -0.85714338]
 [-0.85714338  1.        ]]
```

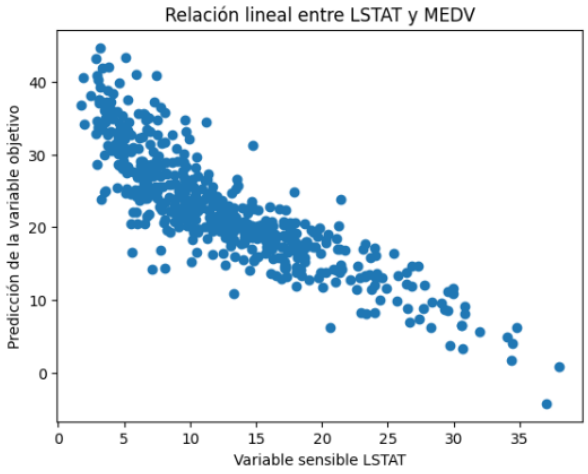
```
[ ] #scatter plot entre la variable sensible B y la variable objetivo
    plt.scatter(data['B'].values,y_pred)
    plt.xlabel("Variable sensible B")
    plt.ylabel("Predicción de la variable objetivo")
    plt.title("Relación lineal entre B y MEDV")
```

```
Text(0.5, 1.0, 'Relación lineal entre B y MEDV')
```




```
[ ] #scatter plot entre la varibale sensible LSTAT y la variable objetivo
    plt.scatter(data['LSTAT'].values,y_pred)
    plt.xlabel("Variable sensible LSTAT")
    plt.ylabel("Predicción de la variable objetivo")
    plt.title("Relación lineal entre LSTAT y MEDV")
```

```
Text(0.5, 1.0, 'Relación lineal entre LSTAT y MEDV')
```

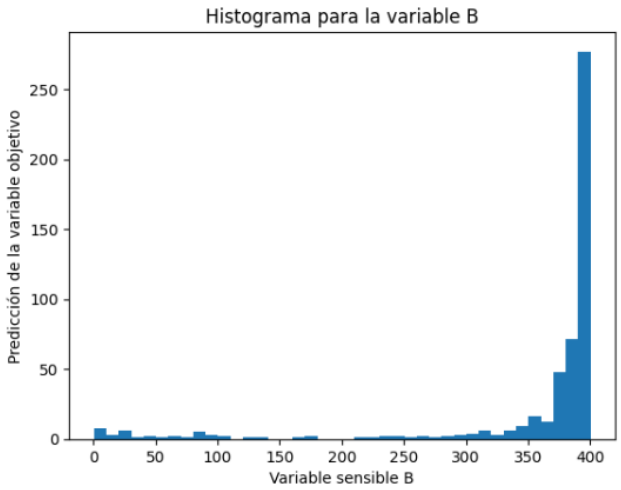


MEDIDA DE FAIRNESS

variable sensible 1 (B)

```
[ ] #Pintar histograma para la división en tramos
    B=data['B']
    bins = np.arange(min(B), max(B) + 10, 10)
    plt.hist(B, bins=bins)
    plt.xlabel("Variable sensible B")
    plt.ylabel("Predicción de la variable objetivo")
    plt.title("Histograma para la variable B")
```

```
Text(0.5, 1.0, 'Histograma para la variable B')
```




```
[ ] #División en tramos teniendo en cuenta la forma del histograma
    data['y_pred'] = y_pred #añadir columna de predicciones al dataframe
    tramo_1_B = data.loc[data['B'] <= 100]
    tramo_2_B = data.loc[(data['B'] <= 300) & (data['B'] > 100)]
    tramo_3_B = data.loc[data['B'] > 300]
```

```
[ ] #FAIRNESS

    #determinar número de filas de cada tramo
    n1_B = len(tramo_1_B)
    n2_B = len(tramo_2_B)
    n3_B = len(tramo_3_B)

    #suma de las predicciones de cada tramo
    suma_1_B = tramo_1_B['y_pred'].sum()
    suma_2_B = tramo_2_B['y_pred'].sum()
    suma_3_B = tramo_3_B['y_pred'].sum()

    #medida de fairness
    f1_B = abs(1/n1_B*suma_1_B - 1/(n2_B+n3_B)*(suma_2_B + suma_3_B))
    f2_B = abs(1/n2_B*suma_2_B - 1/(n1_B+n3_B)*(suma_1_B + suma_3_B))
    f3_B = abs(1/n3_B*suma_3_B - 1/(n2_B+n1_B)*(suma_2_B + suma_1_B))
```

```
[ ] #print de la suma y número de filas
    print(n1_B)
    print(n2_B)
    print(n3_B)
    print(suma_1_B)
    print(suma_2_B)
    print(suma_3_B)
```

```
31
23
452
408.4136221032472
359.8096829749201
10633.376694921833
```

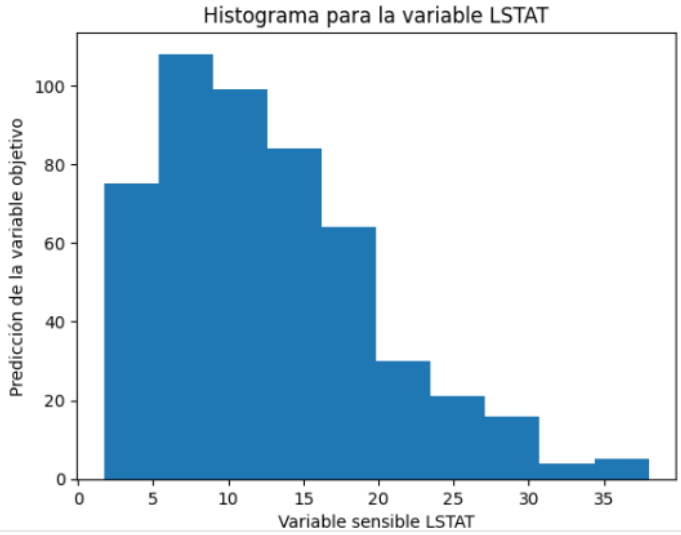
```
[ ] #print de las medidas de fairness para la primera variable
    print(f1_B)
    print(f2_B)
    print(f3_B)
```

```
9.96891729818383
7.216950257871138
9.298812177583063
```

Variable sensible 2 (LSTAT)

```
[ ] #Histograma
    lstat=data['LSTAT']
    plt.hist(lstat)
    plt.xlabel("Variable sensible LSTAT")
    plt.ylabel("Predicción de la variable objetivo")
    plt.title("Histograma para la variable LSTAT")
```

```
Text(0.5, 1.0, 'Histograma para la variable LSTAT')
```




```
[ ] #División en tramos teniendo en cuenta la forma del histograma
    tramo_1_lstat = data.loc[data['LSTAT'] < 10]
    tramo_2_lstat = data.loc[(data['LSTAT'] >= 10) & (data['LSTAT'] < 20)]
    tramo_3_lstat = data.loc[data['LSTAT'] >= 20 ]
```

```
[ ] #FAIRNESS
    #determinar número de filas de cada tramo
    n1_lstat = len(tramo_1_lstat)
    n2_lstat = len(tramo_2_lstat)
    n3_lstat = len(tramo_3_lstat)

    #suma de las predicciones de cada tramo
    suma_1_lstat = tramo_1_lstat['y_pred'].sum()
    suma_2_lstat = tramo_2_lstat['y_pred'].sum()
    suma_3_lstat = tramo_3_lstat['y_pred'].sum()

    #medida de fairness
    f1_lstat = abs(1/n1_lstat*suma_1_lstat - 1/(n2_lstat+n3_lstat)*(suma_2_lstat + suma_3_lstat))
    f2_lstat = abs(1/n2_lstat*suma_2_lstat - 1/(n1_lstat+n3_lstat)*(suma_1_lstat + suma_3_lstat))
    f3_lstat = abs(1/n3_lstat*suma_3_lstat - 1/(n1_lstat+n2_lstat)*(suma_1_lstat + suma_2_lstat))
```

```
[ ] #print del número de filas y de la suma de las predicciones
    print(n1_lstat)
    print(n2_lstat)
    print(n3_lstat)
    print(suma_1_lstat)
    print(suma_2_lstat)
    print(suma_3_lstat)
```

```
219
213
74
6329.348796005358
4221.335660876775
850.9155431178664
```

```
[ ] #print de la medida de fairness
    print(f1_lstat)
    print(f2_lstat)
    print(f3_lstat)
```

```
11.227786911980512
4.687544354121226
12.924021996445177
```

Reentrenamiento del modelo sin las variables sensibles y nueva medida de fairness

```
[ ] #Nuevo dataframe sin las columnas que contienen la información sensible
    data_2 = raw_df.drop(columns=['B','LSTAT','MEDV'],axis=1)

    #entrenamiento del modelo
    X_2=data_2
    y=target

    modelo_2 = LinearRegression()
    modelo_2.fit(X_2,y)

    y_pred_2 = modelo_2.predict(X_2)

    # Coeficientes
    print('Coeficientes: \n', modelo_2.coef_)
    # Término independiente
    print('Termino independiente: \n', modelo_2.intercept_)

    #r^2
    r_sq_2 = modelo_2.score(X_2,y)
    print(f"coefficient of determination: {r_sq_2}")
```

```
Coeficientes:
 [[-1.84032123e-01  3.90999045e-02 -4.23244973e-02  3.48752826e+00
  -2.21821095e+01  6.07574425e+00 -4.51880522e-02 -1.58385220e+00
   2.54721960e-01 -1.22126247e-02 -9.96206157e-01]]
Termino independiente:
 [27.15236786]
coefficient of determination: 0.6703140875272768
```

```
[ ] #medidas estadísticas sobre el modelo usando statsmodel
    x_2=X_2
    x_2=sm.add_constant(x_2)
    model_2=sm.OLS(y,x_2)
    results=model_2.fit()
    print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.670
Model:                            OLS   Adj. R-squared:                  0.663
Method:                 Least Squares   F-statistic:                     91.31
Date:                Wed, 14 Jun 2023   Prob (F-statistic):          1.83e-111
Time:                        08:55:49   Log-Likelihood:                -1559.5
No. Observations:                 506   AIC:                             3143.
Df Residuals:                     494   BIC:                             3194.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         27.1524      5.291      5.132      0.000      16.758      37.547
CRIM          -0.1840      0.036     -5.089      0.000      -0.255      -0.113
ZN             0.0391      0.015      2.535      0.012       0.009       0.069
INDUS         -0.0423      0.069     -0.614      0.539      -0.178       0.093
CHAS           3.4875      0.966      3.611      0.000       1.590       5.385
NOX          -22.1821      4.272     -5.193      0.000     -30.575     -13.790
RM             6.0757      0.397     15.298      0.000       5.295       6.856
AGE           -0.0452      0.014     -3.234      0.001      -0.073      -0.018
DIS           -1.5839      0.224     -7.066      0.000      -2.024      -1.143
RAD            0.2547      0.074      3.425      0.001       0.109       0.401
TAX           -0.0122      0.004     -2.887      0.004      -0.021      -0.004
PTRATIO       -0.9962      0.147     -6.777      0.000      -1.285      -0.707
==============================================================================
Omnibus:                      260.309   Durbin-Watson:                   0.938
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2458.852
Skew:                           2.039   Prob(JB):                         0.00
Kurtosis:                      13.000   Cond. No.                     1.16e+04
==============================================================================
```

```
[ ] #Nueva medida de fairness (CON VARIABLES SENSIBLES Y NUEVO MODELO ENTRENADO)
    data['y_pred_2'] = y_pred_2
    print(data)
```

```
[ ] #VARIABLE SENSIBLE 1 (B)
    tramo_1_B2 = data.loc[data['B'] <= 100]
    tramo_2_B2 = data.loc[(data['B'] <= 300) & (data['B'] > 100)]
    tramo_3_B2 = data.loc[data['B'] > 300]

    suma_1_B2 = tramo_1_B2['y_pred_2'].sum()
    suma_2_B2 = tramo_2_B2['y_pred_2'].sum()
    suma_3_B2 = tramo_3_B2['y_pred_2'].sum()

    f1_B2 = abs(1/n1_B*suma_1_B2 - 1/(n2_B+n3_B)*(suma_2_B2 + suma_3_B2))
    f2_B2 = abs(1/n2_B*suma_2_B2 - 1/(n1_B+n3_B)*(suma_1_B2 + suma_3_B2))
    f3_B2 = abs(1/n3_B*suma_3_B2 - 1/(n2_B+n1_B)*(suma_2_B2 + suma_1_B2))
```

```
[ ] #print de la suma de predicciones y medidas de fairness
    print(suma_1_B2)
    print(suma_2_B2)
    print(suma_3_B2)
    print(f1_B2)
    print(f2_B2)
    print(f3_B2)
```

```
521.7527174656731
380.12037206380285
10499.726910470526
6.074208825967364
6.2918257030980165
6.528130805395172
```

```
[ ] # VARIABLE SENSIBLE 2 (LSTAT)
    tramo_1_lstat2 = data.loc[data['LSTAT'] < 10]
    tramo_2_lstat2 = data.loc[(data['LSTAT'] >= 10) & (data['LSTAT'] < 20)]
    tramo_3_lstat2 = data.loc[data['LSTAT'] >= 20]

    suma_1_lstat2 = tramo_1_lstat2['y_pred_2'].sum()
    suma_2_lstat2 = tramo_2_lstat2['y_pred_2'].sum()
    suma_3_lstat2 = tramo_3_lstat2['y_pred_2'].sum()

    f1_lstat2 = abs(1/n1_lstat*suma_1_lstat2 - 1/(n2_lstat+n3_lstat)*(suma_2_lstat2 + suma_3_lstat2))
    f2_lstat2 = abs(1/n2_lstat*suma_2_lstat2 - 1/(n1_lstat+n3_lstat)*(suma_1_lstat2 + suma_3_lstat2))
    f3_lstat2 = abs(1/n3_lstat*suma_3_lstat2 - 1/(n2_lstat+n1_lstat)*(suma_2_lstat2 + suma_1_lstat2))
```

```
[ ] #print de la suma de las predicciones y de la medida de fairness
    print(suma_1_lstat2)
    print(suma_2_lstat2)
    print(suma_3_lstat2)
    print(f1_lstat2)
    print(f2_lstat2)
    print(f3_lstat2)
```

```
6154.705583153762
4160.395433824497
1086.4989830217444
9.821816382285697
5.181635829524673
9.195129960929602
```

MÉTODO DE GRAM-SCHMIDT. Entrenamiento y medida de fairness del modelo equitativo

```
[ ] #método de Gram-Schmidt
    X_fair = InformationFilter(["B","LSTAT"]).fit_transform(data) #devuelve una matriz, no un DataFrame

    #la matriz ha incluído las columnas B y LSTAT, hay que eliminarlas para hacer medida de fairness
    X_fair_pd = pd.DataFrame(X_fair)
    X_fair_pd = X_fair_pd.rename(columns = {0:'CRIM', 1:'ZN', 2: 'INDUS', 3:'CHAS', 4:'NOX', 5: 'RM', 6:'AGE',7:'DIS',8:'RAD',9:'TAX',10:'PTRATIO', 11:'B', 12:'LSTAT'})
    X_fair_pd = X_fair_pd.drop(columns = ['B', 'LSTAT'])
```

```
[ ] #Entrenamiento del nuevo modelo y predicción
    modelo_fair = LinearRegression()
    modelo_fair.fit(X_fair_pd, y)

    y_pred_fair = modelo_fair.predict(X_fair_pd)
```

```
[ ] #medidas estadísticas sobre el modelo. usando statsmodels.
    x_fair = X_fair_pd
    x_fair = sm.add_constant(x_fair)
    model_fair = sm.OLS(y,x_fair)
    results = model_fair.fit()
    print(results.summary())
```

```
[ ]
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.223
Model:                            OLS   Adj. R-squared:                  0.205
Method:                 Least Squares   F-statistic:                     12.87
Date:                Wed, 14 Jun 2023   Prob (F-statistic):           1.37e-21
Time:                        08:56:06   Log-Likelihood:                -1776.5
No. Observations:                 506   AIC:                             3577.
Df Residuals:                     494   BIC:                             3628.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         22.2283      0.371     59.950      0.000      21.500      22.957
CRIM          -0.1021      0.057     -1.802      0.072      -0.213       0.009
ZN             0.0473      0.024      1.996      0.047       0.001       0.094
INDUS          0.0109      0.106      0.103      0.918      -0.197       0.219
CHAS           2.7520      1.488      1.850      0.065      -0.172       5.676
NOX          -11.9514      5.531     -2.161      0.031     -22.819      -1.084
RM             4.6367      0.509      9.104      0.000       3.636       5.637
AGE           -0.0024      0.023     -0.106      0.915      -0.047       0.042
DIS           -1.2777      0.322     -3.967      0.000      -1.910      -0.645
RAD            0.2534      0.110      2.306      0.022       0.037       0.469
TAX           -0.0112      0.006     -1.732      0.084      -0.024       0.002
PTRATIO       -0.7339      0.181     -4.056      0.000      -1.089      -0.378
==============================================================================
Omnibus:                       18.965   Durbin-Watson:                   0.552
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               38.988
Skew:                           0.173   Prob(JB):                     3.42e-09
Kurtosis:                       4.315   Cond. No.                     2.44e+03
==============================================================================
```

```
[ ]  #Nueva medida de fairness
     data['y_pred_fair'] = y_pred_fair
```

```
[ ]  #VARIABLE SENSIBLE 1 (B)
     tramo_1_Bfair = data.loc[data['B'] <= 100]
     tramo_2_Bfair = data.loc[(data['B'] <= 300) & (data['B'] > 100)]
     tramo_3_Bfair = data.loc[data['B'] > 300]

     suma_1_Bfair = tramo_1_Bfair['y_pred_fair'].sum()
     suma_2_Bfair = tramo_2_Bfair['y_pred_fair'].sum()
     suma_3_Bfair = tramo_3_Bfair['y_pred_fair'].sum()

     f1_Bfair = abs(1/n1_B*suma_1_Bfair - 1/(n2_B+n3_B)*(suma_2_Bfair + suma_3_Bfair))
     f2_Bfair = abs(1/n2_B*suma_2_Bfair - 1/(n1_B+n3_B)*(suma_1_Bfair + suma_3_Bfair))
     f3_Bfair = abs(1/n3_B*suma_3_Bfair - 1/(n2_B+n1_B)*(suma_2_Bfair + suma_1_Bfair))
```

```
[ ]  #print de la suma de predicciones y medidas de fairness
     print(suma_1_Bfair)
     print(suma_2_Bfair)
     print(suma_3_Bfair)
     print(f1_Bfair)
     print(f2_Bfair)
     print(f3_Bfair)

     844.8115649991621
     534.2830854283714
     10022.505349572468
     5.027168209818402
     0.7300784253088359
     3.365105421023099
```

```
[ ] # VARIABLE SENSIBLE 2 (LSTAT)
    tramo_1_lstat_fair = data.loc[data['LSTAT'] < 10]
    tramo_2_lstat_fair = data.loc[(data['LSTAT'] >= 10) & (data['LSTAT'] < 20)]
    tramo_3_lstat_fair = data.loc[data['LSTAT'] >= 20]

    suma_1_lstat_fair = tramo_1_lstat_fair['y_pred_fair'].sum()
    suma_2_lstat_fair = tramo_2_lstat_fair['y_pred_fair'].sum()
    suma_3_lstat_fair = tramo_3_lstat_fair['y_pred_fair'].sum()

    f1_lstat_fair = abs(1/n1_lstat*suma_1_lstat_fair - 1/(n2_lstat+n3_lstat)*(suma_2_lstat_fair + suma_3_lstat_fair))
    f2_lstat_fair = abs(1/n2_lstat*suma_2_lstat_fair - 1/(n1_lstat+n3_lstat)*(suma_1_lstat_fair + suma_3_lstat_fair))
    f3_lstat_fair = abs(1/n3_lstat*suma_3_lstat_fair - 1/(n2_lstat+n1_lstat)*(suma_2_lstat_fair + suma_1_lstat_fair))
```

```
[ ] #print de la suma de las predicciones y de la medida de fairness
    print(suma_1_lstat_fair)
    print(suma_2_lstat_fair)
    print(suma_3_lstat_fair)
    print(f1_lstat_fair)
    print(f2_lstat_fair)
    print(f3_lstat_fair)
```

```
5132.294040848397
4569.317215054093
1699.9887440975128
1.590860971939101
1.8661777817723326
0.5153874034453629
```