APRENDIZAJE AUTOMÁTICO

Predicción de notas de alumnos de secundaria

Albert Barreiro Díaz
Ramón Cano Aparicio
Grupo 13

Índice

Introducción

Este trabajo se propone estudiar si es posible predecir las notas de los alumnos a partir de sus datos y situación personal. Para ello hemos usado el dataset Student Performance Data Set [1], que contiene datos sobre estudiantes de secundaria de dos centros educativos portugueses.

Dicho dataset contiene información respecto a la situación de los alumnos y sus familias, así como sus resultados en cada uno de los tres años de enseñanza en portugues y matemáticas. Los resultados se encuentran en los atributos G1, G2 y G3, correspondientes a la nota del primer, segundo y tercer año respectivamente.

Para este trabajo hemos predecir solamente las notas de la asignatura de matemáticas. Además, solo usamos únicamente como target la variable G3, la correspondiente al último año, y descartamos totalmente las variables G1 y G2. Esto se debe a que la correlación entre G1, G2 y G3 es demasiado alta por lo que se podría decir G3 solamente con estas dos variables. Eso haría que perdiéramos el objetivo de predecir las notas según la información de la situación personal de los alumnos.

Finalmente aclaramos que la variable G3, es decir, la nota está representada según el sistema educativo de secundaria portugués como un valor numérico del 0 al 20 [2].

En cuanto al pre-procesamiento de datos, después de estudiar los datos hemos creado dos versiones. Una de ellas contiene todas los atributos del dataset salvo los anteriormente mencionados, la otra versión tampoco con tiene otros datos que hemos considerado que añadían ruido a los datos.

Hemos usado ambas versiones del dataset para probar en diferentes modelos lineales (Linear Regression, LASSO y Quadratic SVM) y no lineales (MLP y SVM usando un kernel RBF).

Teniendo en cuenta el score resultante y el tiempo de ejecución, finalmente nos hemos decantado por usar el modelo Quadratic SVM con el preprocesamiento que elimina las variables que añaden ruido.

Nuestro modelo elegido, con un score R^2 de 0.149 sobre la partición de test no es capaz de predecir las notas de los alumnos a partir de la información de su situación personal.

[1] http://archive.ics.uci.edu/ml/datasets/Student+Performance

[2] https://pt.wikipedia.org/wiki/Nota_escolar

Trabajos previos relacionados

Mismo dataset:

En el artículo Using Data Mining to Predict Secondary School Student Performance, de P. Cortez y A. Silva [3] utilizan el mismo dataset que nosotros. Prueban a clasificar y a hacer una regresión del dataset y además de todo esto, prueban como predicen los modelos utilizando las columnas (G1,G2), (G2), ().

En el cuaderno de Kaggle[4] del usuario onizukaharuto también se intenta predecir la nota de los estudiantes obviando las notas G1 y G2.

Otros datasets pero tematica similar:

Esta otra página[5] intentan analizar las notas de los estudiantes desde una vista más analítica para finalmente responder una serie de preguntas que pueden ayudar a entender mejor nuestro dataset de estudiantes.

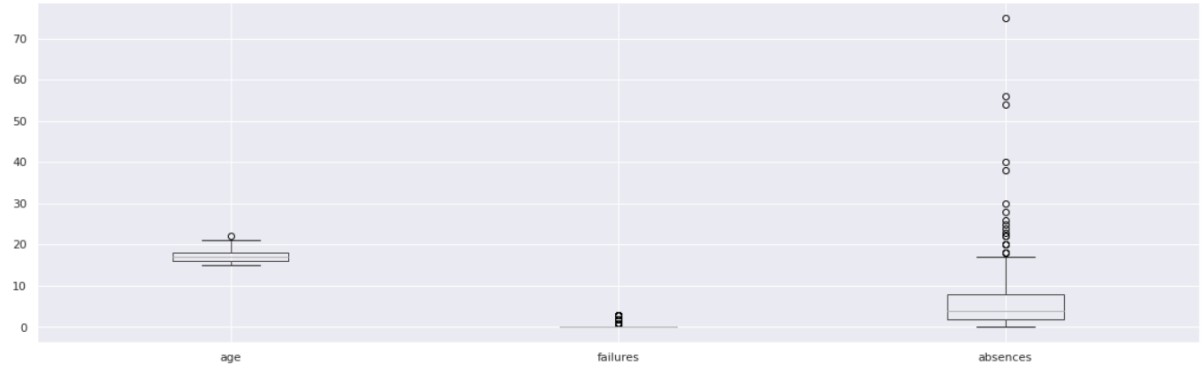
[3] http://www3.dsi.uminho.pt/pcortez/student.pdf

[4] https://www.kaggle.com/onizukaharuto/studentperformance

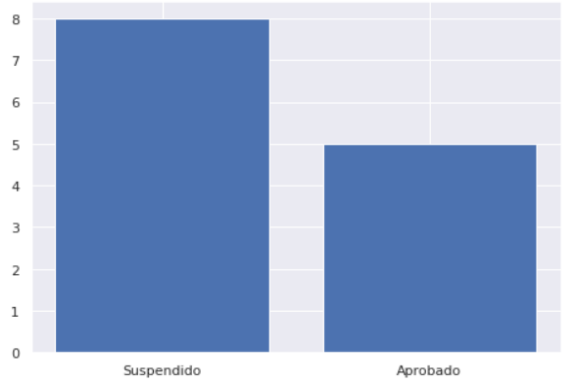
[5] https://rstudio-pubs-static.s3.amazonaws.com/517704_199c58baabf44be287a93a4c1aacd4d9.html

Proceso de exploración de datos

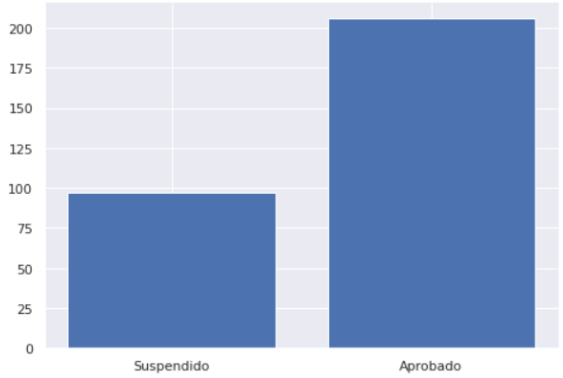
Empezamos comprobando que el dataset no contiene missing values.
A continuación, de nuestras variables numéricas miramos que no contengan outliers o datos incoherentes. Estos atributos son: **age**, **failures** y **absences**.



En el caso de **age** y **failures** no vimos nada que fuera extremadamente extraño. Pero, en **absences** no es habitual que un niño de secundaria falte a tantas clases.
Para comprobar si son datos incoherentes o no, comprobamos cuántos de estos alumnos aprueban o suspenden. Nos damos cuenta de que la tasa de alumnos que suspenden teniendo más de 20 ausencias es mayor que la de los demás alumnos.

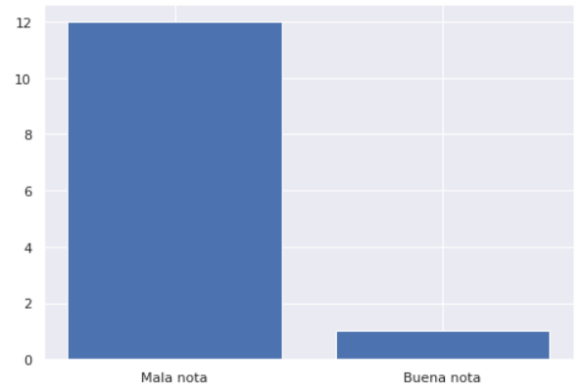


Notas de alumnos con más de 20 ausencias

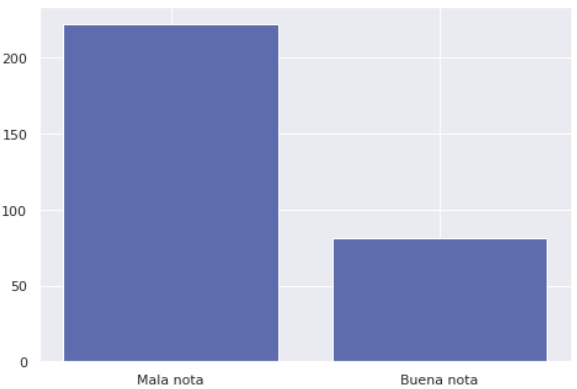


Notas de alumnos con menos de 20 aus.

No solo eso, la mayoría de estos tienen una nota igual o menos a un suficiente.



Notas de alumnos con más de 20 ausencias

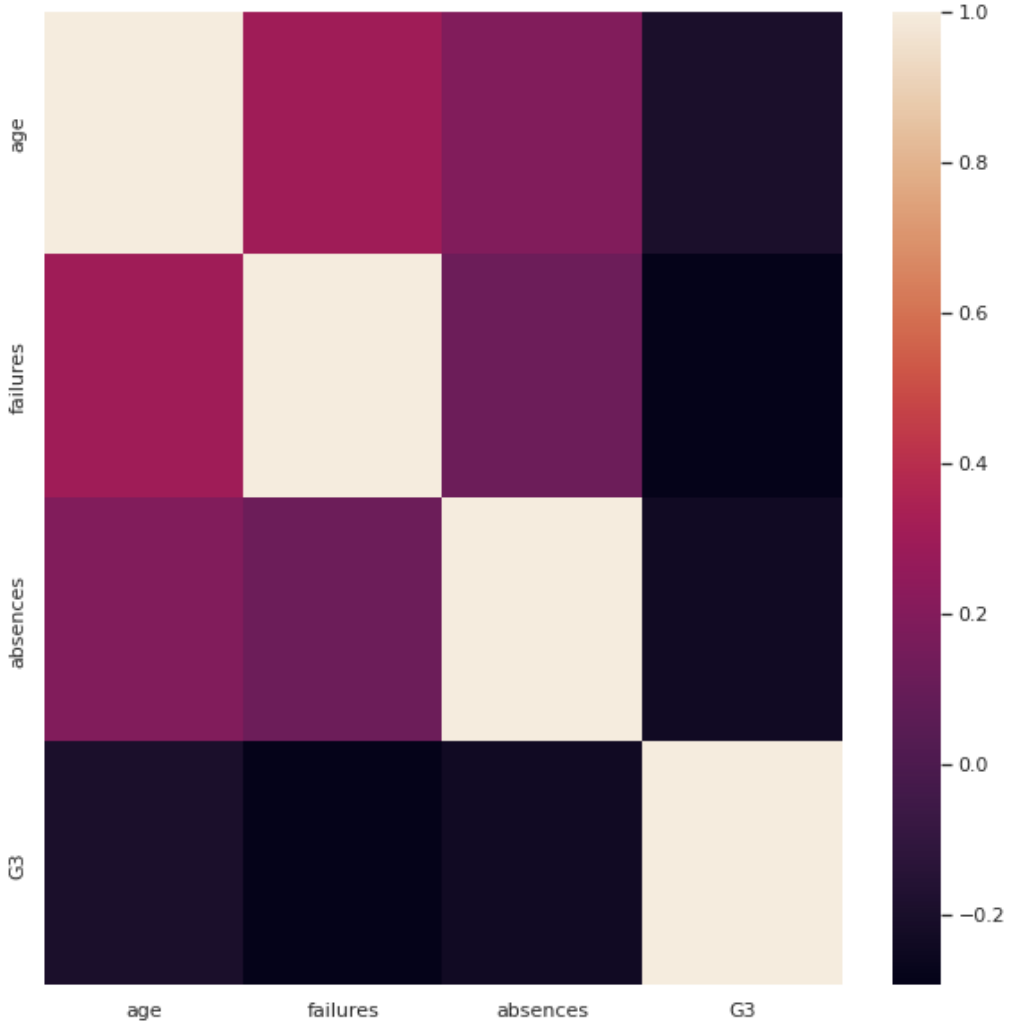


Notas de alumnos con menos de 20 aus.

Por todo esto, finalmente decidimos que estas ausencias son coherentes con el dataset.

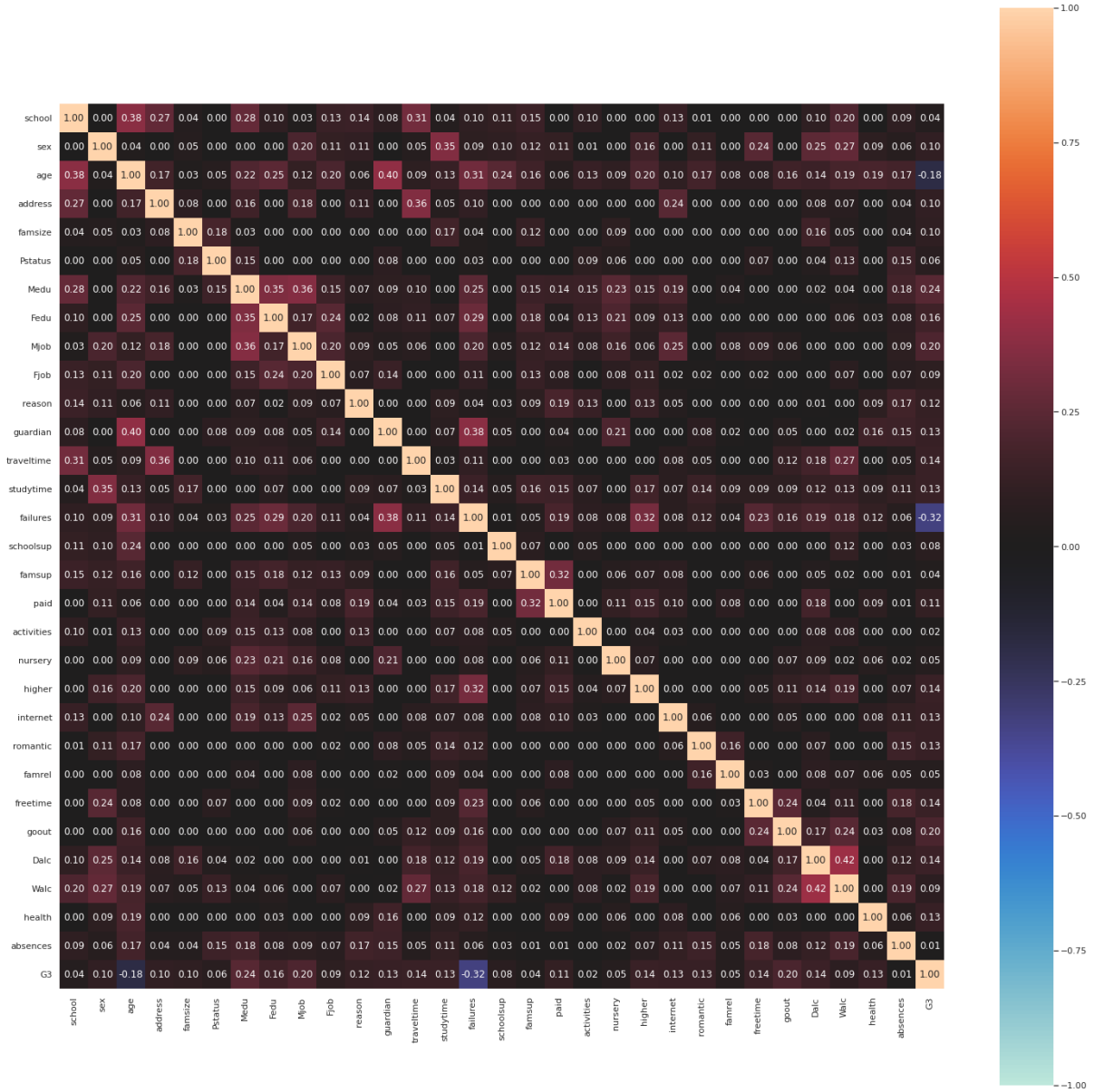
Para la codificación de las variables binarias hemos puesto a 0 y 1 cada una de sus dos categorías. Las variables que son binarias són: **school**, **sex**, **address**, **famsize**, **Pstatus**, **schoolsup**, **famsup**, **paid**, **activities**, **nursery**, **higher**, **internet**, **romantic**.
En el resto de variables categóricas no codificadas (**Mjob**, **Fjob**, **reason** y, **guardian**) utilizamos one-hot encoding ya que no tiene sentido representarlas como variables numéricas.
Para finalizar con la codificación, dejamos como están las variables categóricas que ya venían codificadas en el dataset (**Medu**, **Fedu**, **traveltime**, **studytime**, **famrel**, **freetime**, **goout**, **Dalc**, **Walc**, **health**). Como expresan un orden, preferimos dejarlas así ya que si usaramos one-hot encoding estaríamos perdiendo información útil.

En la correlación entre variables numéricas.



Vemos que no parece haber redundancia en los datos. Además, vemos que la correlación entre nuestra variable target 'G3' y el atributo 'absences' es bastante baja. Quizás esto es debido a los outliers que vimos anteriormente, ya que observamos que, aunque los valores fueran muy dispares, afectan poco al número de aprobados.

Correlación entre variables categóricas:



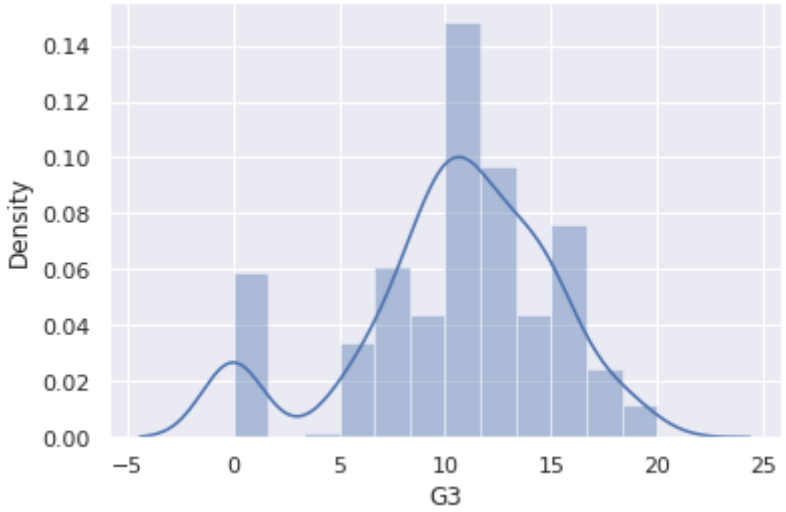
Para la correcta visualización de este plot mejor hacerlo en el pdf del código.

Por poner unos ejemplos una de las variables con más correlación es Dalc y Walc (correlación de 0.41). Esto tiene sentido ya que los alumnos que consumen alcohol entre semana probablemente también lo hagan los fines de semana.
Otros ejemplos serían la variables failures con G3(-0.29) y age con G3(-0.20). Los alumnos que suspenden más asignaturas y repiten cursos son los que tienden a tener peores notas también.

Basándonos en la correlación, vemos que no hay nada excesivamente correlacionado directamente y por lo tanto suponemos que no hay variables redundantes.

Finalmente. para la normalización y transformación de las variables numéricas (**age**, **failures, absences**). Para simplificar un poco el preprocesado hemos terminado haciendo una transformación logarítmica y después, un escalado entre 0 y 1.

Más adelante encontramos que borrar las variables desbalanceadas puede servir de ayuda. Así que entrenamos los modelos sin borrar y borrando estas columnas.



Al hacer el histograma del target encontramos que la distribución de nuestros datos se asemejan bastante a una curva gaussiana, aunque hay una cantidad de 0 un poco extraña.

Protocolo de remuestreo

Como el dataset con el que contamos es bastante pequeño, hemos cogido un 80% de las muestras como training set, y el 20% como test set.

Además, para la elección de hiperparámetros de cada modelo y posterior evaluación hemos usado la técnica de grid-search con un cross-validation de 5 particiones excepto para las redes neuronales en las que que usamos 3 particiones. Esto es debido a que hemos intentado reducir el tiempo de entrenamiento. Después de hacer grid-search siempre comprobamos el modelo con un cross-validation de 5 particiones. Así, podemos comparar todos los modelos de regresión lineal con los demás.

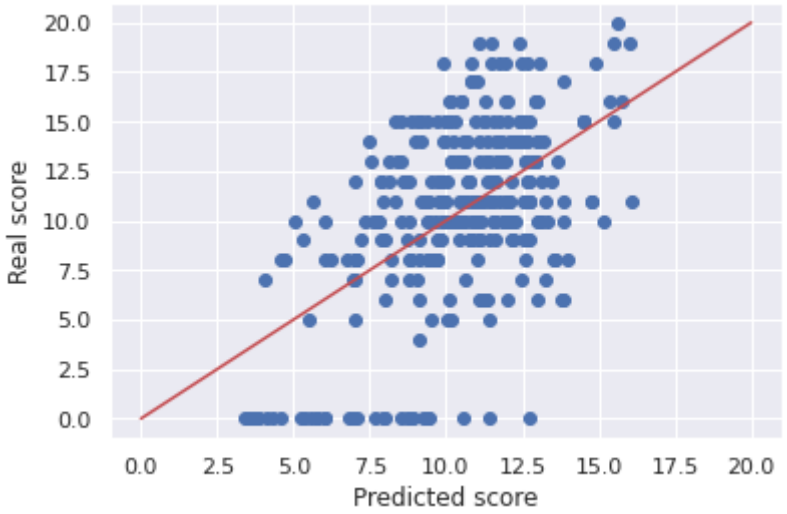
Finalmente, una vez elegido el modelo que hemos considerado mejor, lo hemos probado con el test set del 20% para ver qué tan bueno era nuestro resultado.

Resultado de modelos lineales/cuadráticos

Regresión lineal

Resultados obtenidos con la regresión lineal:

Modelo	R^2 score	Tiempo (s)
Regresión Lineal	0.063	0.011
Regresión Lineal regularización	0.054	0.006
Regresión Lineal regularización sin one-hot encoding	**0.206**	**0.004**
Regresión Lineal regularización sin one-hot encoding sin absencias	0.119	0.004



Valor real vs.predicho en Regresión Lineal usando regularización sin variables desbalanceadas

LASSO

Mejores parámetros para LASSO con todos los atributos:
`{'alpha': 0.01}`

Mejores parámetros para LASSO sin variables desbalanceadas:
`{'alpha': 0.0}`

Modelo	R^2 score	Tiempo (s)
LASSO	0.141	0.017
LASSO sin one-hot encoding	**0.206**	**0.015**

Valor real vs.predicho en el modelo LASSO sin variables desbalanceadas

SVM cuadratica

Mejores parámetros para Quadratic SVM con todos los atributos:
`{'C': 20, 'degree': 1, 'epsilon': 5, 'gamma': 'auto'}`

Mejores parámetros para Quadratic SVM sin variables desbalanceadas:
`{'C': 10, 'degree': 1, 'epsilon': 5, 'gamma': 'auto'}`

Modelo	R^2 score	Tiempo (s)
Quadratic SVM	0.137	0.014
Quadratic SVM sin one-hot encoding	**0.216**	**0.011**

Valor real vs.predicho en el modelo Quadratic SVM sin variables desbalanceadas

Resultado de modelos no lineales

MLP

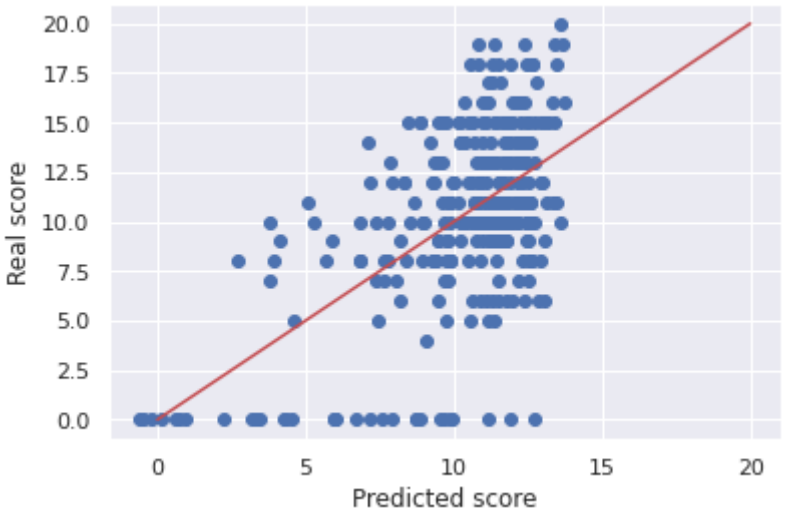
Mejores parámetros para MLP sin exploración del parámetro max_iter:
`{'activation': 'tanh', 'alpha': 0.1, 'hidden_layer_sizes': (100, 50), 'learning_rate_init': 0.1, 'solver': 'adam'}`

Mejores parámetros para MLP con exploración del parámetro max_iter:
`{'activation': 'relu', 'alpha': 0.3, 'hidden_layer_sizes': (1500, 800, 200), 'learning_rate_init': 0.001, 'max_iter': 30, 'solver': 'lbfgs'}`

Mejores parámetros para MLP con exploración del parámetro max_iter y sin variables desbalanceadas:
`{'activation': 'tanh', 'alpha': 0.1, 'hidden_layer_sizes': (4000, 200, 1000, 500), 'learning_rate_init': 0.3, 'max_iter': 20, 'solver': 'lbfgs'}`

Tiempos:

MLP	-0.086	0.184
MLP-max-iter	-0.101	23.476
MLP-no-ohe	**0.223**	**19.296**



Valor real vs.predicho en el modelo MLP sin variables desbalanceadas

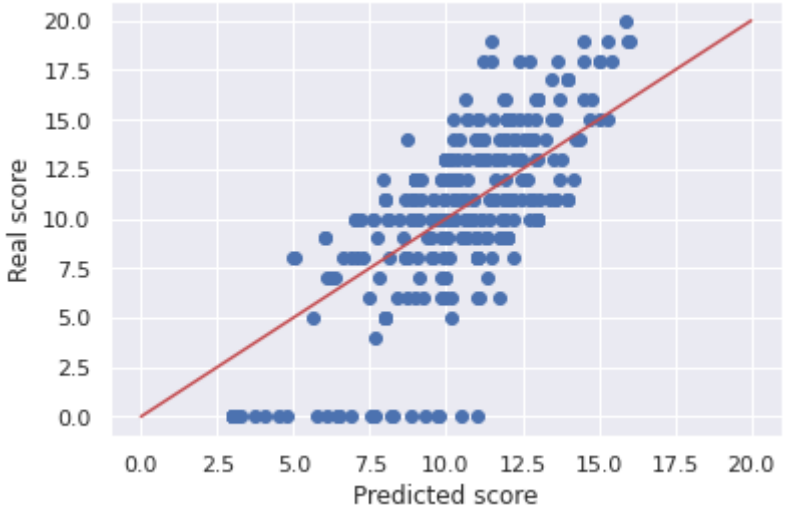
SVM con kernel RBF

Mejores parámetros para SVM con variables desbalanceadas:
`{'C': 5, 'degree': 1, 'epsilon': 1, 'gamma': 'scale'}`

Mejores parámetros para SVM sin variables desbalanceadas:
`{'C': 5, 'degree': 1, 'epsilon': 3, 'gamma': 'scale'}`

Tiempos:

SVMwithK	0.073	0.017
SVMwithK-no-ohe	**0.2069**	**0.0149**



Valor real vs.predicho en el modelo SVM con kernel RBF sin variables desbalanceadas

Modelo final

Parece que la elección de qué datos usar para entrenar los modelos ha sido más decisivo en la puntuación que el propio modelo en si. Para todos los modelos en los que no hemos usado las variables desbalanceadas hemos obtenido unos scores muy parecidos.

El modelo con mejor score ha sido MLP, con un R^2=0.223 y, en segundo lugar, Quadratic SVM con un R^2=0.216. La diferencia entre los scores es muy pequeña, de hecho, ambos resultados son bastante poco satisfactorios siendo los dos mejores modelos que tenemos.

Si comparamos el tiempo de entrenamiento de los dos modelos la diferencia es mucho mayor, necesitando 23.476s para entrenar el modelo MLP y solamente 0.011s para entrenar el modelo Quadratic SVM.

Por ello, hemos decidido decantar nos por el modelo Quadratic SVM, sacrificando un poco de precisión a cambio de ahorrarnos mucho tiempo de computación.

Interpretación de los modelos

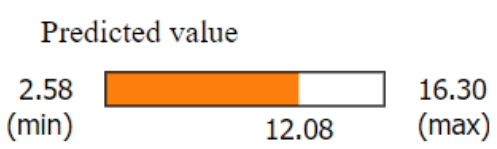
Tener una mayor cantidad de suspensos, ser hombre y salir mucho con tus amigos indican una correlación negativa respecto a la nota.
Faltar mucho a clase, un alto nivel de educación de la madre y estudiar mucho tiempo influyen positivamente en la nota del alumno.

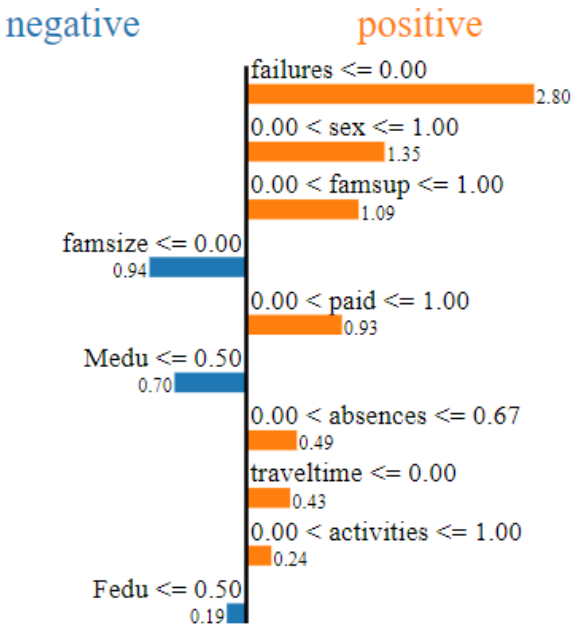
La importancia de los atributos no varía demasiado entre modelos lineales y no lineales como podemos ver a continuación.

Los suspensos (failures) influyen negativamente en la nota. Ser mujer y tener soporte de la familia influye positivamente. Las ausencias siempre son algo bueno y una familia pequeña es algo negativo. No salir mucho con tus amigos y gastar muchas horas de estudio es bueno para la nota.

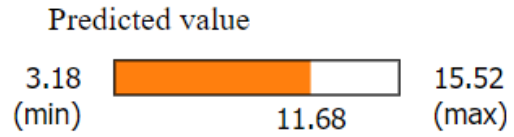
<u>Quadratic SVM</u>

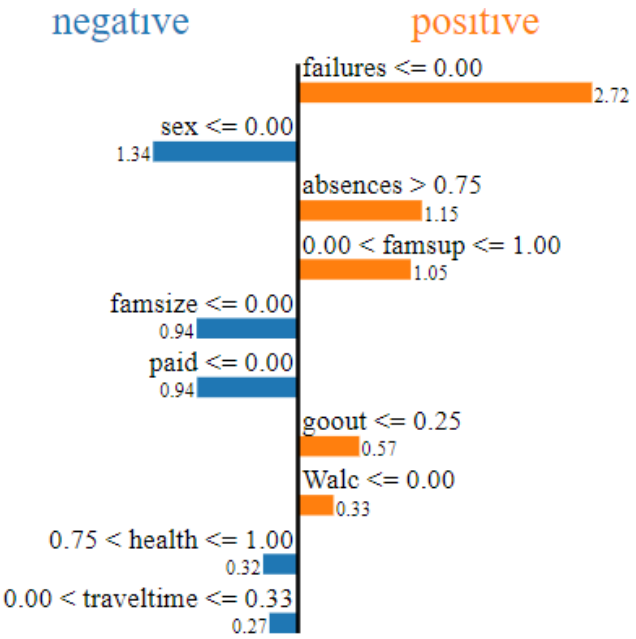
Alumno tipico:



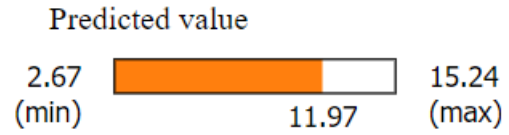


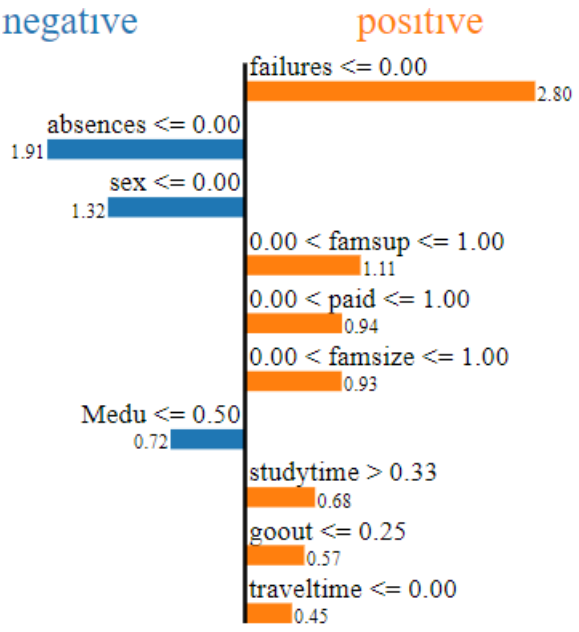
Alumno con 75 ausencias:



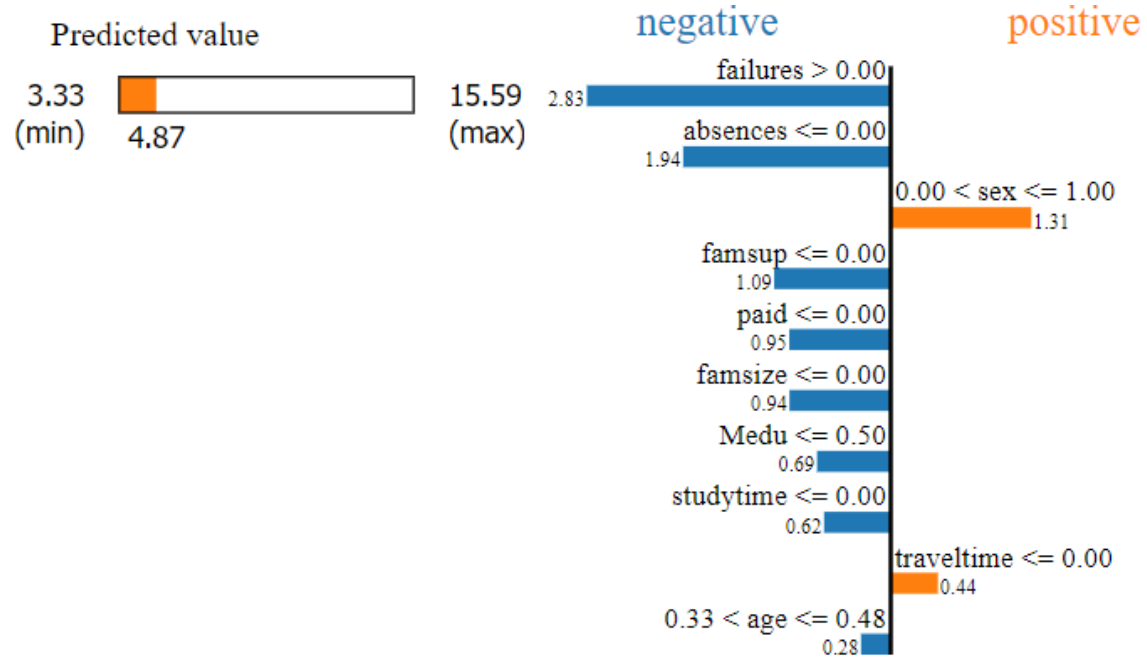


Alumno con un 0 de nota y con muy alto error de predicción:



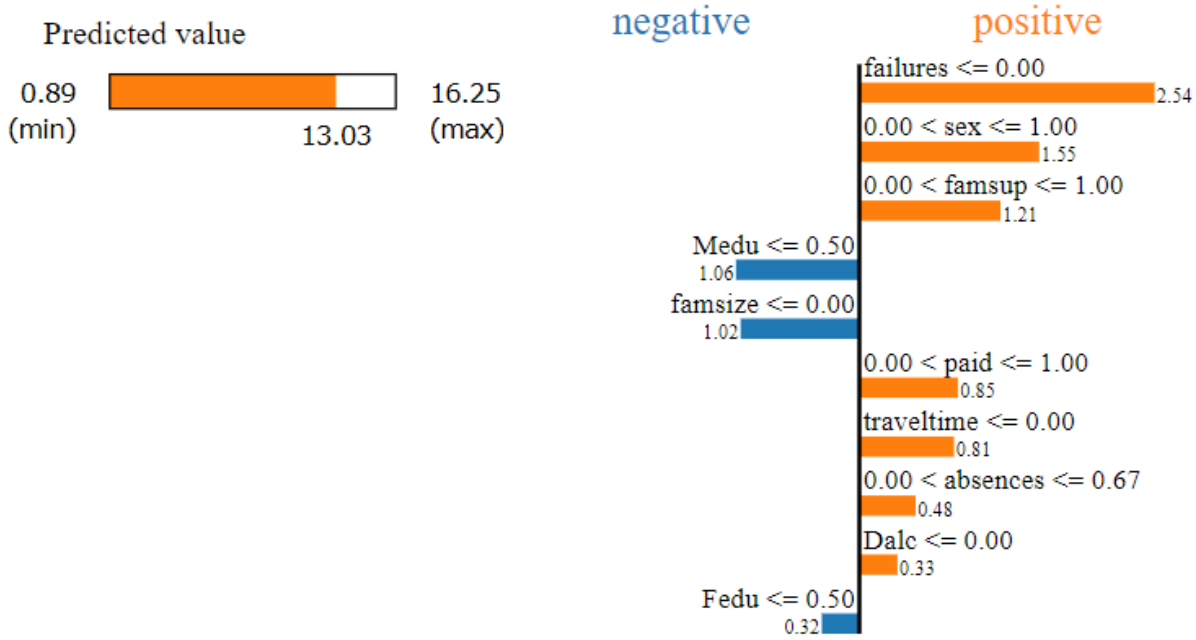


Alumno con un 0 de nota y con un bajo error de predicción:

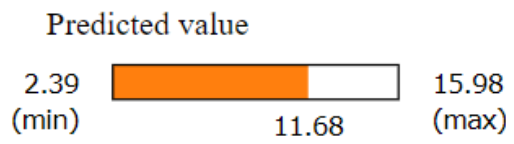


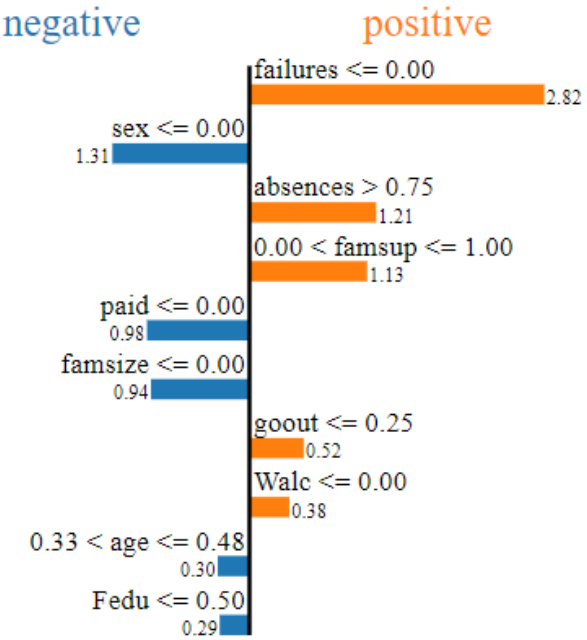
<u>SVM con kernel rbf</u>

Alumno típico:

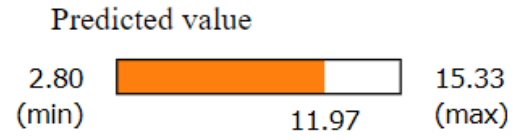


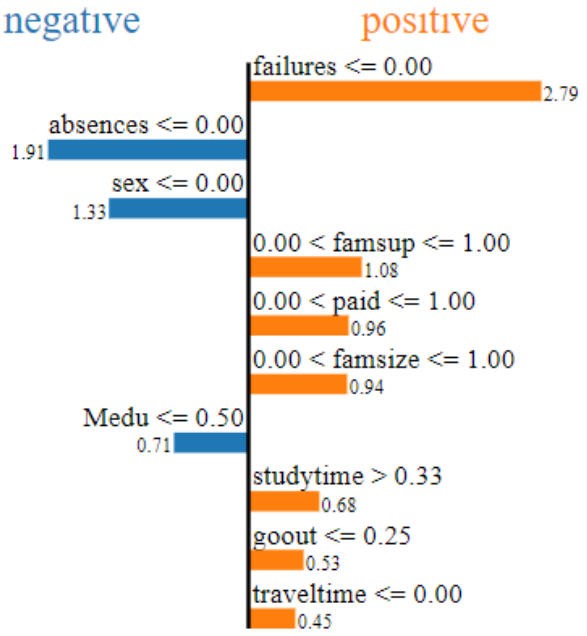
Alumno con 75 ausencias:



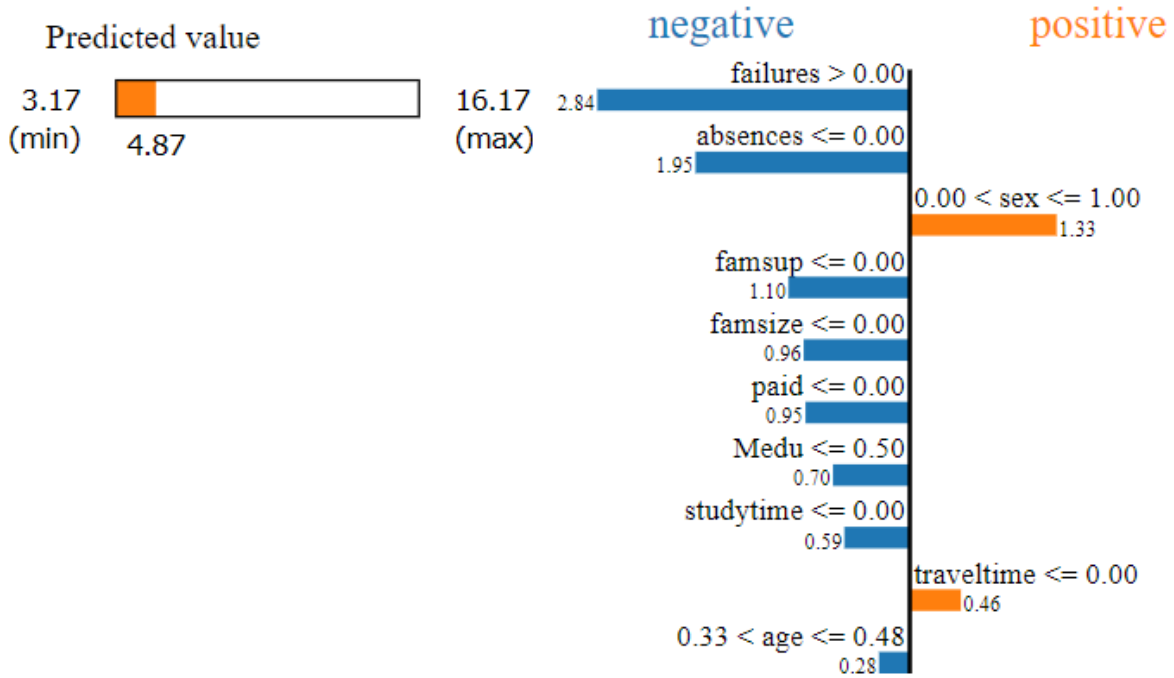


Alumno con un 0 de nota y con muy alto error de predicción:





Alumno con un 0 de nota y con un bajo error de predicción:



Conclusiones y autovaloración

Autovaloración

Nuestra poca experiencia al trabajar estudiando datos nos ha pasado factura, teniendo muchas dudas durante el preprocesamiento de estos y obviando errores de los que no nos hemos dado cuenta hasta que ya estábamos entrenando los modelos.

Nuestro proceso de trabajo ha sido iterativo, es decir, a medida que avanzábamos teníamos que volver atrás a cambiar cosas que al inicio no habíamos tenido en cuenta.

Conclusiones

No hemos podido demostrar que podemos predecir el desempeño de los estudiantes basándonos en sus condiciones personales.

Aun así, no pensamos que quede demostrado lo contrario, es decir, no sabemos si es posible o no predecir las notas de los alumnos teniendo unos datos mejores, sobretodo en lo relativo a la cantidad de estos.

Extensiones y limitaciones

Los datos con los que hemos trabajado han sido bastante escasos teniendo en cuenta la complejidad del problema. Proponemos realizar un estudio similar pero usando un conjunto de datos más amplio, para poder así entrenar modelos más complejos que sean capaces de aprender aquellos casos que nuestros modelos no han sido capaces.

Otra posible extensión de este estudio es el de, con el fin de simplificar el problema, clasificar las notas de los alumnos como suspenso, aprobado, bien, notable y excelente, convirtiendo así el problema en uno de clasificación.

En la misma línea que la propuesta anterior, se podría intentar clasificar a los alumnos según aprobados o suspensos.

Referencias

UCI Machine Learning Repository: Student Performance Data Set. (2014, 27 noviembre). UCI Machine Learning Repository. Recuperado 9 de enero de 2022, de http://archive.ics.uci.edu/ml/datasets/Student+Performance

Wikipedia contributors. (2018, 12 diciembre). Academic grading in Portugal. Wikipedia. Recuperado 9 de enero de 2022, de https://en.wikipedia.org/wiki/Academic_grading_in_Portugal

User guide: contents. (s. f.). Scikit-Learn. Recuperado 9 de enero de 2022, de https://scikit-learn.org/stable/user_guide.html

USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE. (s. f.). Universidade do Minho. Recuperado 9 de enero de 2022, de http://www3.dsi.uminho.pt/pcortez/student.pdf

Pyplot tutorial — Matplotlib 3.5.1 documentation. (s. f.). Matplotlib. Recuperado 9 de enero de 2022, de https://matplotlib.org/stable/tutorials/introductory/pyplot.html

O. (2020, 3 febrero). StudentPerformance. Kaggle. Recuperado 9 de enero de 2022, de https://www.kaggle.com/onizukaharuto/studentperformance

Molnar, Christoph. “Interpretable machine learning. A Guide for Making Black Box Models Explainable”, 2019. https://christophm.github.io/interpretable-ml-book/.

STUDENT PERFORMANCE ANALYSIS. (s. f.). rstudio-pubs-static. Recuperado 9 de enero de 2022, de https://rstudio-pubs-static.s3.amazonaws.com/517704_199c58baabf44be287a93a4c1aacd4d9.html

Xu, W. (2021, 11 diciembre). What’s the difference between Linear Regression, Lasso, Ridge, and ElasticNet in sklearn? Medium. Recuperado 9 de enero de 2022, de https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29

Donthi, S. (2021, 30 diciembre). Support Vector Machines, Dual Formulation, Quadratic Programming & Sequential Minimal Optimization. Medium. Recuperado 9 de enero de 2022, de https://towardsdatascience.com/support-vector-machines-dual-formulation-quadratic-programming-sequential-minimal-optimization-57f4387ce4dd

Yıldırım, S. (2021, 14 diciembre). Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters. Medium. Recuperado 9 de enero de 2022, de https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167

Eby, M. (2021, 12 diciembre). Hyperparameter Tuning - Analytics Vidhya. Medium. Recuperado 9 de enero de 2022, de https://medium.com/analytics-vidhya/hyperparameters-80cb4f442e5

How to Use LIME to Understand sklearn Models Predictions [Python]? by Sunny Solanki. (s. f.). Medium. Recuperado 9 de enero de 2022, de https://coderzcolumn.com/tutorials/machine-learning/how-to-use-lime-to-understand-sklearn-models-predictions