

Taller de iniciación a Inteligencia Artificial Explicable (XAI) para educacion

Explorando modelos y explicaciones con
Google Colab y SHAP



3 de Julio de 2025



Universitat d'Alacant
Universidad de Alicante

David Gil Méndez
Carmen García Barceló



Objetivos del taller

- Entender qué es la IA y la IA explicable (XAI)
- Cargar y usar datos en Google Colab
- Entrenar un modelo de IA básico
- Visualizar e interpretar explicaciones con SHAP



¿Qué es la IA?

- Sistemas que aprenden patrones a partir de datos
- Se usan para hacer predicciones o clasificaciones
- Casos en educación: abandono escolar, recomendadores, corrección automática



¿Qué es la IA explicable (XAI)?

- Técnicas para entender cómo decide un modelo
- Permiten saber qué variables influyen más
- Clave para confianza, ética y toma de decisiones

Herramientas del taller

- Google Colab: cuadernos interactivos en la nube
- Kaggle: repositorio de datos públicos
- SHAP: biblioteca para explicar decisiones de modelos

Todo es online y en la nube → sin necesidad de instalar nada



Flujo del ejercicio

- 1. Cargar datos desde Kaggle
- 2. Entrenar modelo de clasificación (Random Forest)
- 3. Evaluar precisión
- 4. Visualizar importancia de variables
- 5. Analizar predicciones individuales



¿Qué es Kaggle?

- Plataforma para ciencia de datos con miles de datasets públicos
- Útil para explorar, descargar o usar directamente conjuntos de datos educativos
- Permite crear notebooks en la nube y compartir proyectos



Registro en Kaggle

- Ir a <https://www.kaggle.com>
- Crear una cuenta gratuita (se puede usar cuenta de Google)

Sólo parte avanzada, si quiero acceder directamente a los datasets desde programación

- Para descargar datasets, ir a 'My Account' y generar un API token
- Descargar kaggle.json y subirlo a Google Colab (si se usa API)



Registro en Kaggle

- Dentro de kaggle, y los datasets, selecciono students

The screenshot shows the Kaggle website interface. On the left is a sidebar with navigation links: Home, Competitions, Datasets (selected), Models, Code, Discussions, Learn, and More. The main content area displays a search for 'students' with filters for 'All datasets', 'Computer Science', 'Education', 'Classification', 'Computer Vision', 'NLP', 'Data Visualization', and 'Pre-Trained Model'. A list of datasets is shown, with 'Students Performance' by Joakim Arvidsson selected. The dataset details page for 'Students Performance' is displayed, showing a 'Data Card' with a 'Code' tab selected. The 'Data Card' shows a table of student performance data with columns: STUDENT ID, Student Age, Sex, Graduated high-s..., and Scholarshi. The table has 22 rows of data. The 'Data Explorer' on the right shows a histogram of the 'Student Age' column and a summary of the dataset.

Students Performance (11.65 kB)

Download

10 of 33 columns

Detail Compact Column

STUDENT ID	Student Age	Sex	Graduated high-s...	Scholarshi
Student ID	Age group	Gender Group	See description for class labels	See description for class labels
STUDENT11	1	1	1	3
STUDENT12	1	1	1	4
STUDENT13	1	1	1	4
STUDENT14	2	1	2	5
STUDENT15	3	2	2	4
STUDENT16	2	2	2	3
STUDENT17	1	1	2	5
STUDENT18	2	2	2	3
STUDENT19	1	1	2	4
STUDENT20	1	2	1	3
STUDENT21	1	2	2	5
STUDENT22	1	2	2	5

Data Explorer
Version 2 (22.67 kB)

highereducation: StudentsPerform

Summary

2 files

66 columns

Elegimos uno y lo descargamos

¿Qué es Google Colab?

- Entorno gratuito de notebooks en la nube
- No requiere instalación ni configuración
- Permite ejecutar código Python paso a paso
- Ideal para prácticas educativas e investigación
- Se accede desde:
<https://colab.research.google.com>



Cómo abrir el cuaderno de ejemplo

- 1. Descargar el archivo .ipynb desde el enlace proporcionado
- 2. Acceder a <https://colab.research.google.com>
- 3. Ir a 'Archivo' > 'Subir notebook' y seleccionar el archivo
- 4. Hacer clic en las celdas para ejecutar paso a paso



Cómo abrir el cuaderno de ejemplo

- Pero en nuestro caso vamos a hacerlo más sencillo:
- <https://github.com/davidogm/TalleresULPGC>
— XAI (Podemos abrir uno de los cuadernos .ipynb)

The screenshot shows the GitHub interface for the repository `davidogm / TalleresULPGC`. The left sidebar displays the file structure, with the `XAI` folder expanded, showing files like `StudentPerformanceFactors.csv`, `Student_Success_Analysis.ipynb`, `XAI_Colab_Kaggle_StudentPerfo...`, `readme`, `student_performance_dataset.csv`, and `README.md`.

The main content area shows the `TalleresULPGC / XAI` directory. It includes a table of files with their commit history:

Name	Last commit message	Last commit date
..		
StudentPerformanceFactors.csv	Add files via upload	8 minutes ago
Student_Success_Analysis.ipynb	Creado con Colab	now
XAI_Colab_Kaggle_StudentPerformance.ipynb	Creado con Colab	9 minutes ago
readme	Create readme	19 hours ago
student_performance_dataset.csv	Add files via upload	8 minutes ago

Below the table, the `readme` file is visible, with an edit icon.



Cómo abrir el cuaderno de ejemplo

- <https://github.com/davidogm/TalleresULPGC>

TalleresULPGC / XAI / Student_Success_Analysis.ipynb

davidogm Creado con Colab 9228634 · 2 minutes ago History

Preview Code Blame 2770 lines (2770 loc) · 1020 KB Code 55% faster with GitHub Copilot Raw Download Edit

Open in Colab

```
In [ ]: # Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
import matplotlib.gridspec as gridspec
from sklearn.preprocessing import OrdinalEncoder, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import shap
```

Lo abrimos aquí

Student Performance Factors Dataset

```
In [ ]: df_performance = pd.read_csv("StudentPerformanceFactors.csv")
df_performance.head()
```

```
Out[ ]: 
```

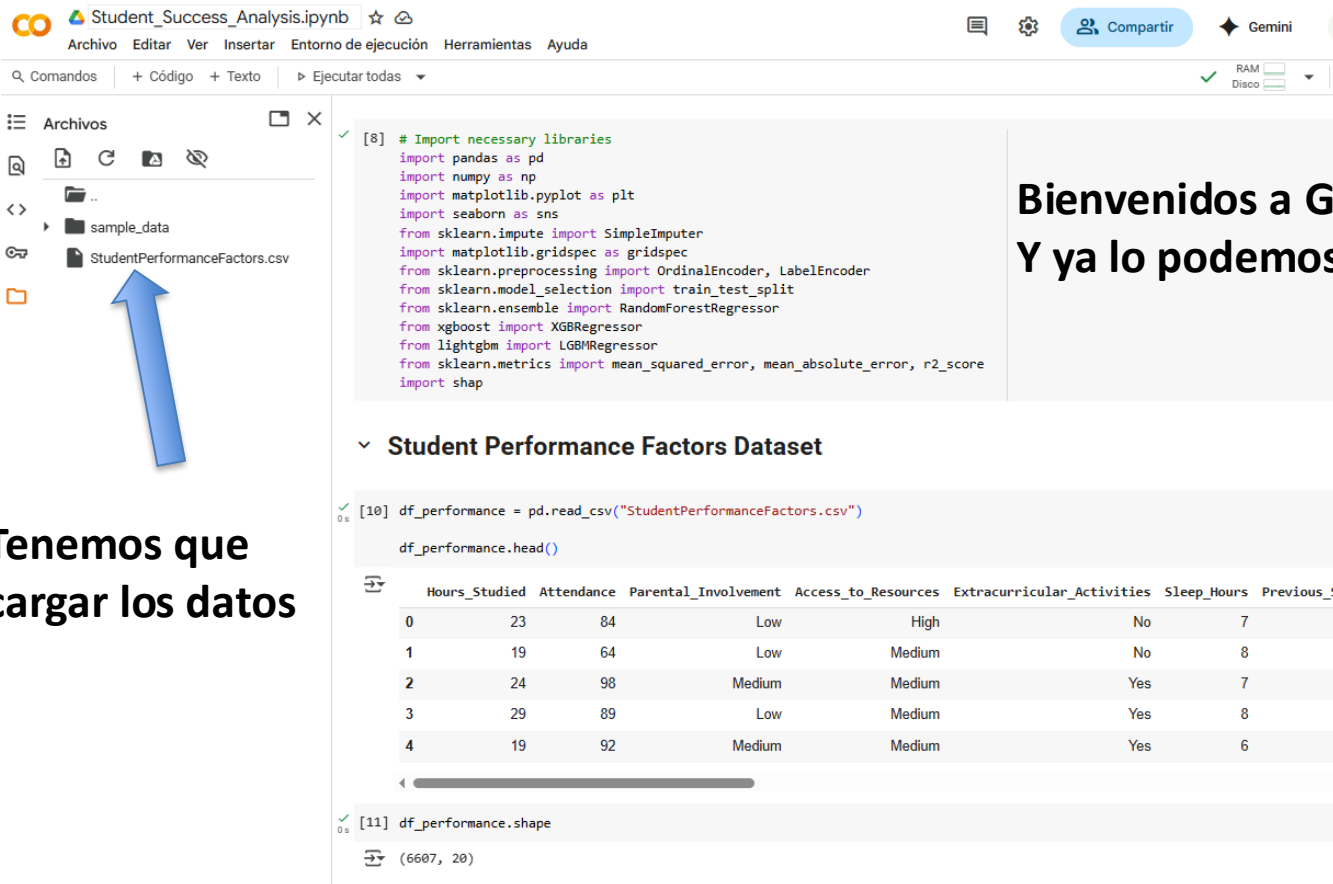
	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Mo
0	23	84	Low	High	No	7	73	
1	19	64	Low	Medium	No	8	59	
2	24	98	Medium	Medium	Yes	7	91	
3	29	89	Low	Medium	Yes	8	98	
4	19	92	Medium	Medium	Yes	6	65	

```
In [ ]: df_performance.shape
```



Cómo abrir el cuaderno de ejemplo

- <https://github.com/davidogm/TalleresULPGC>



Archivos

sample_data

StudentPerformanceFactors.csv

```
[8] # Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
import matplotlib.gridspec as gridspec
from sklearn.preprocessing import OrdinalEncoder, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import shap
```

**Bienvenidos a Google colab !!!
Y ya lo podemos probar y ejecutar**

Student Performance Factors Dataset

```
[10] df_performance = pd.read_csv("StudentPerformanceFactors.csv")
df_performance.head()
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_5
0	23	84	Low	High	No	7	
1	19	64	Low	Medium	No	8	
2	24	98	Medium	Medium	Yes	7	
3	29	89	Low	Medium	Yes	8	
4	19	92	Medium	Medium	Yes	6	

```
[11] df_performance.shape
```

(6607, 20)

Tenemos que
cargar los datos

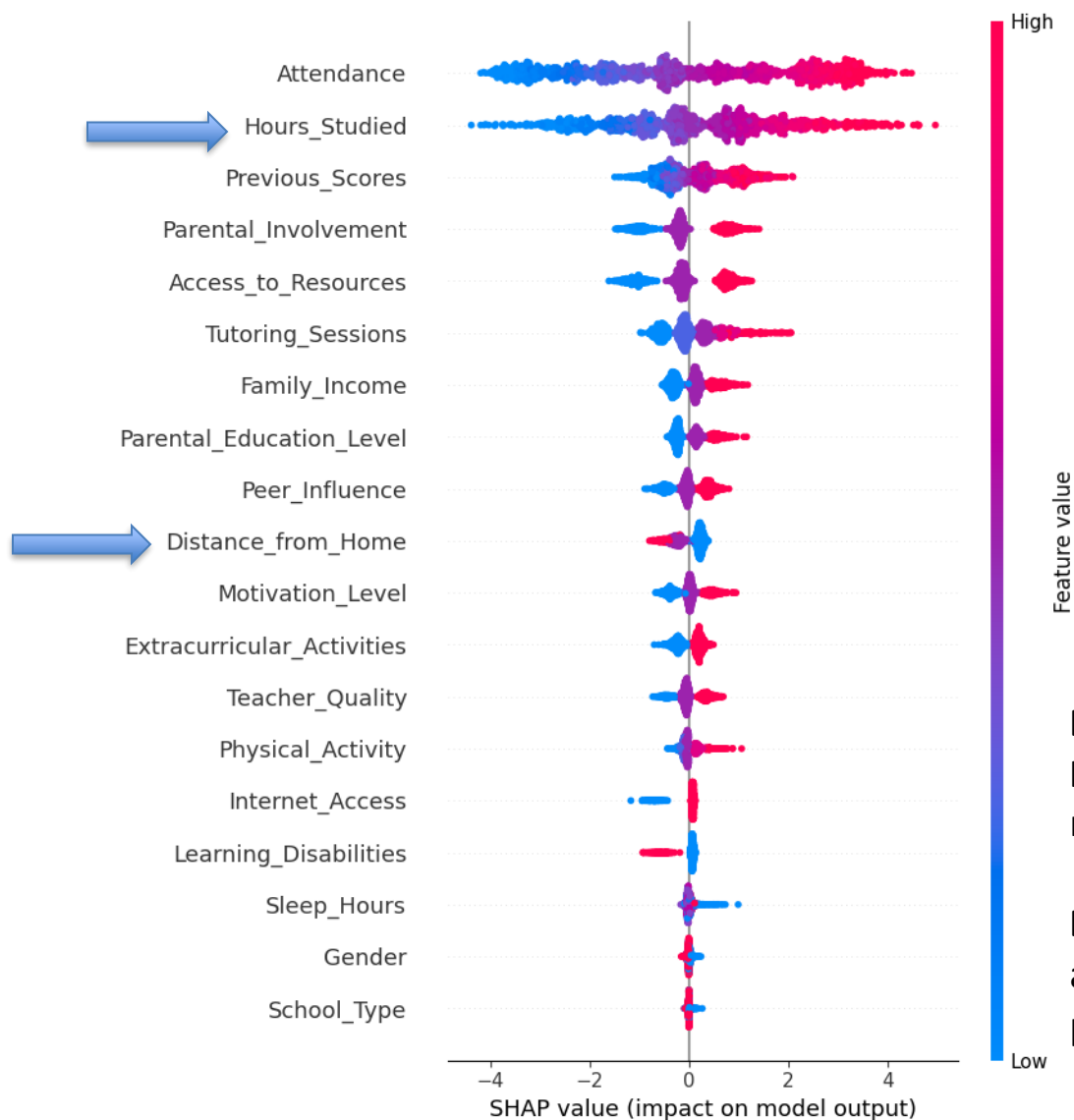


Interpretación con SHAP

- Summary plot: importancia global de variables
- Force plot: explicación local de una predicción
- ¿Por qué este estudiante fue clasificado así?

Veamos un caso real de explicabilidad con SHAP

Explicabilidad **global** del modelo:



Cada punto es un estudiante

Puntos rojos: Valor alto de la característica

Puntos azules: Valor bajo de la característica

Puntos a la derecha: Característica contribuye a aprobar

Puntos a la izquierda: Característica contribuye a suspender

Ejemplo:

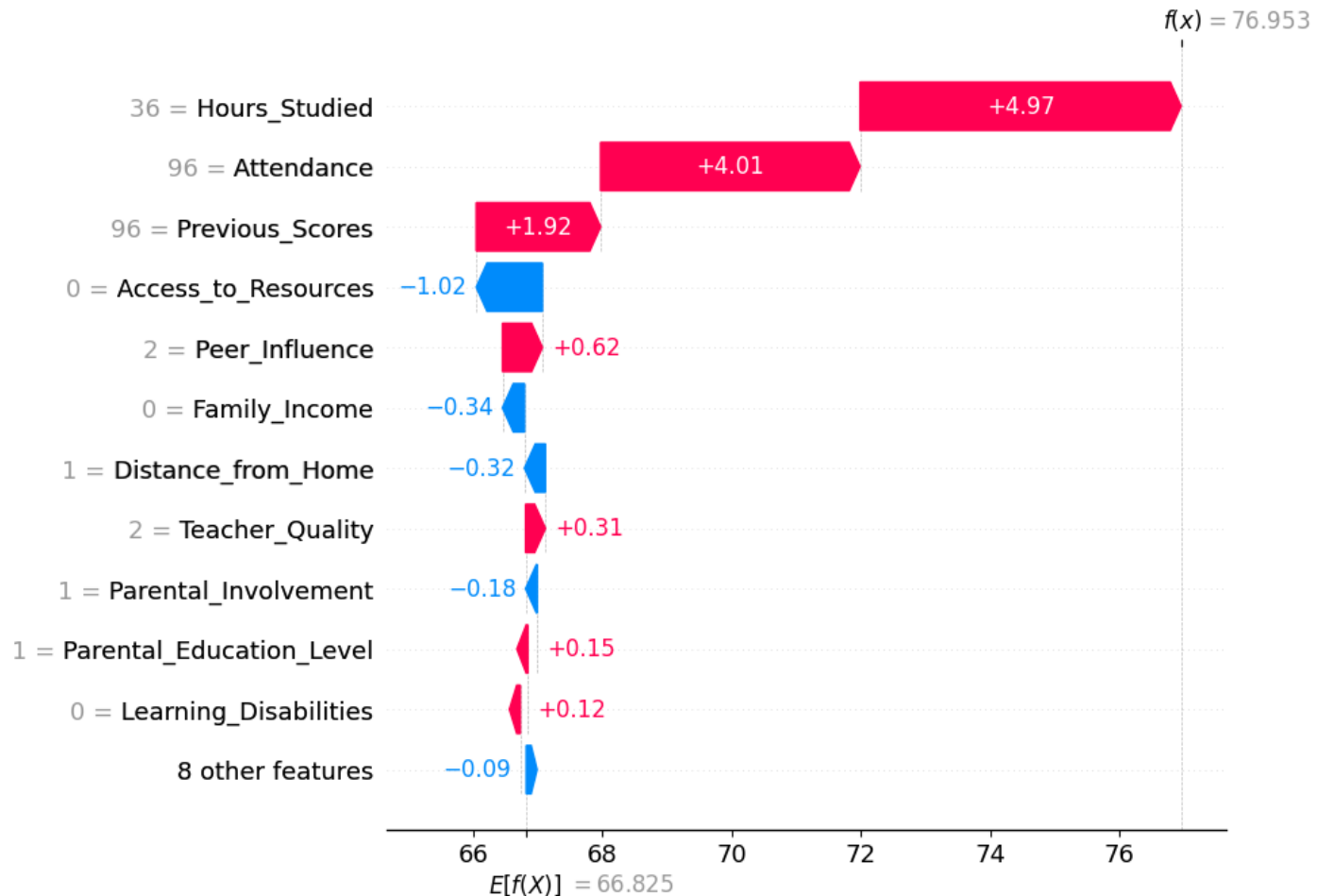
Horas de estudio: Puntos **rojos** a la derecha = más horas, más posibilidades de aprobar

Distancia de casa a la escuela: Puntos **azules** a la derecha = más distancia, menos posibilidades de aprobar

Veamos un caso real de explicabilidad con SHAP

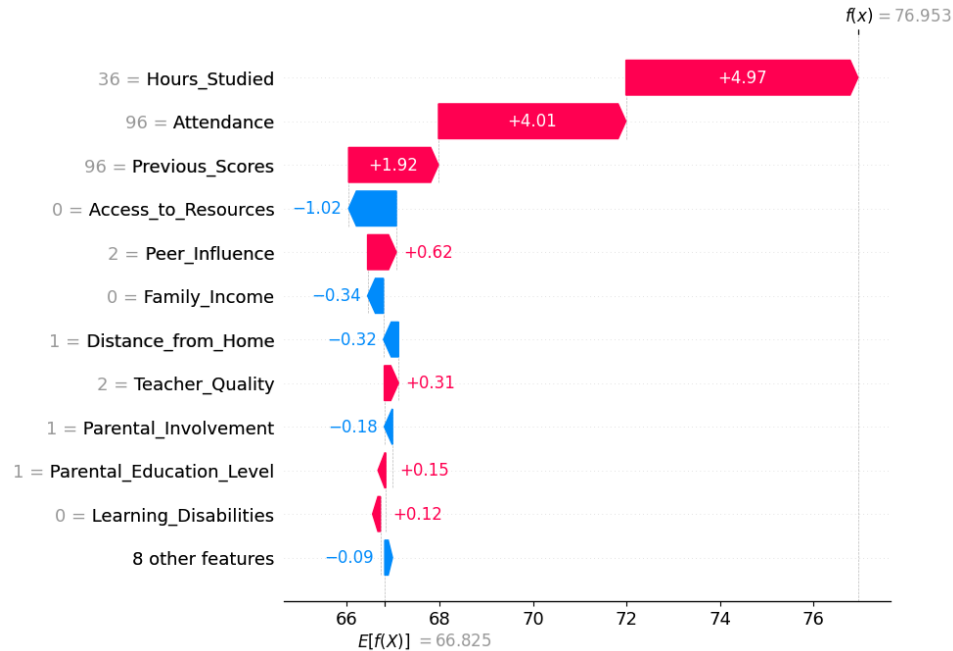
Explicabilidad **local**: Para estudiar casos de estudiantes concretos

Estudiante con nota alta:



Veamos un caso real de explicabilidad con SHAP

Estudiante con nota alta:



¿Qué representa este gráfico?

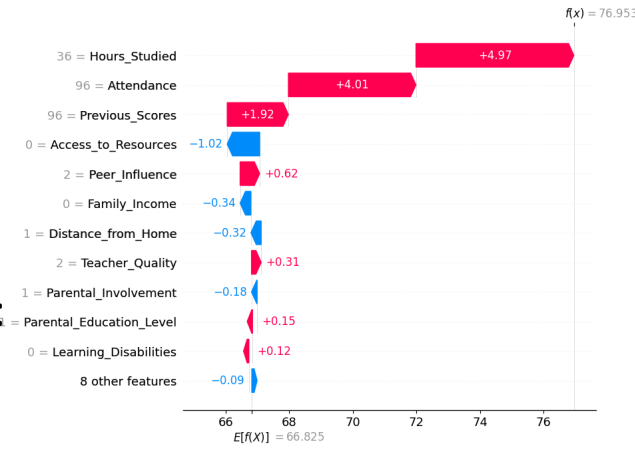
- Predicción final del modelo: $f(x) = 76.953$ → Es la nota estimada para un estudiante concreto.
- Valor base: $E[f(X)] = 66.825$ → Es la media de predicción del modelo para todos los estudiantes. Es como un punto de partida “neutral”.
- El modelo parte de esta media y suma o resta el efecto de cada característica para llegar a la predicción final.

Estudiante con nota alta:

Cómo se interpreta?

Cada barra representa una característica (feature) y cómo esta influye en la predicción individual de este estudiante:

- Barras rojas: características que aumentan la predicción.
- Barras azules: características que disminuyen la predicción.
- La longitud de la barra indica la magnitud del impacto.



Análisis del caso concreto: “Estudiante con nota alta”

- Hours_Studied (36 horas): contribuye con +4.97 puntos. Es el mayor factor positivo.
- Attendance (96%): añade +4.01 puntos. También muy relevante.
- Previous_Scores (96): suma +1.92 puntos.
- Access_to_Resources (0): resta -1.02 puntos. No tiene acceso, lo cual penaliza.
- Otras variables tienen efectos más pequeños, positivos o negativos.

¿Qué nos dice este gráfico?

Este estudiante obtiene una nota estimada bastante alta (76.953) principalmente gracias a:

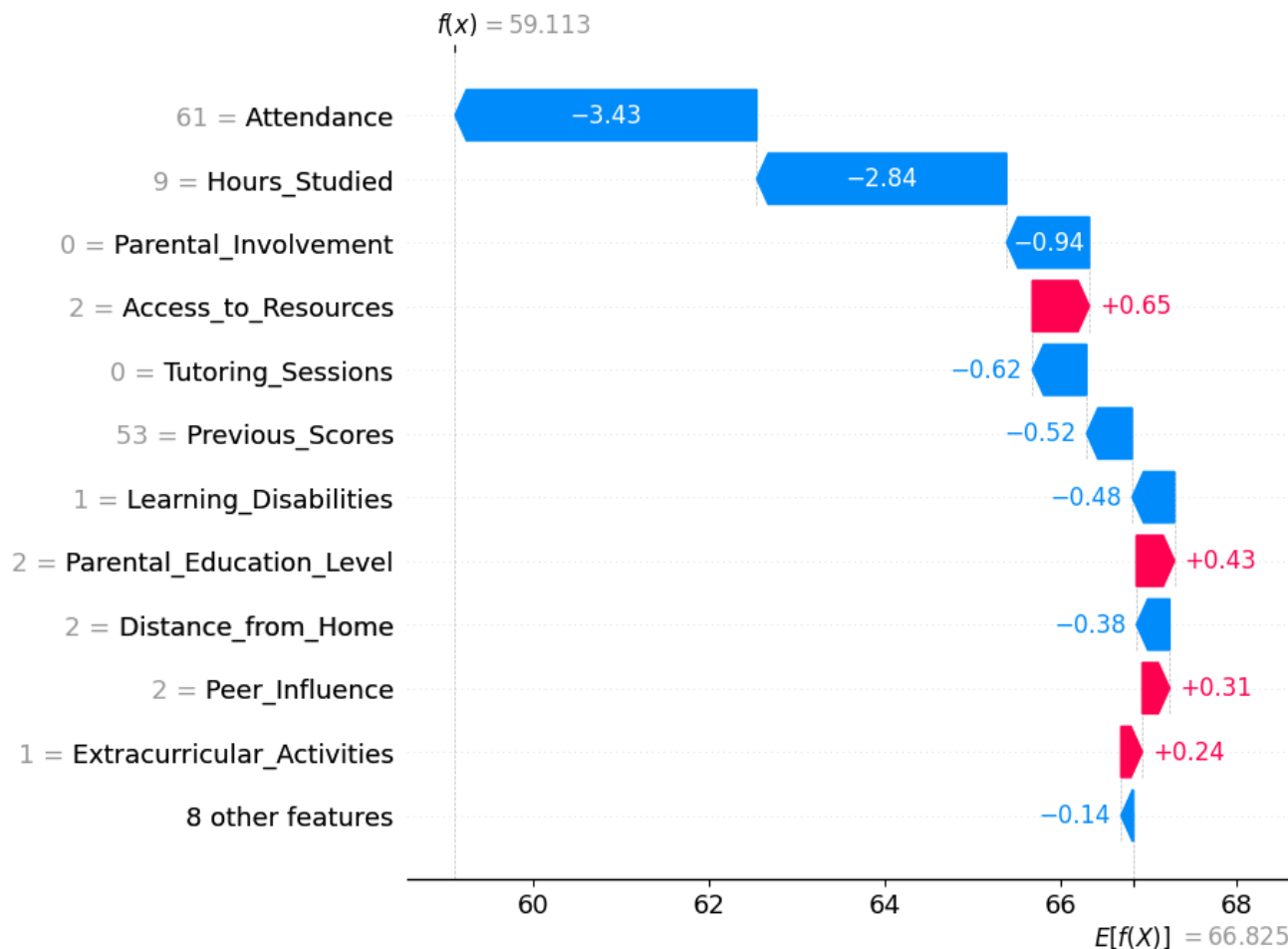
- Su alta dedicación al estudio (Hours_Studied),
- Su asistencia regular,
- Y su buen desempeño previo.

Aunque hay factores negativos como la falta de acceso a recursos, su efecto queda compensado

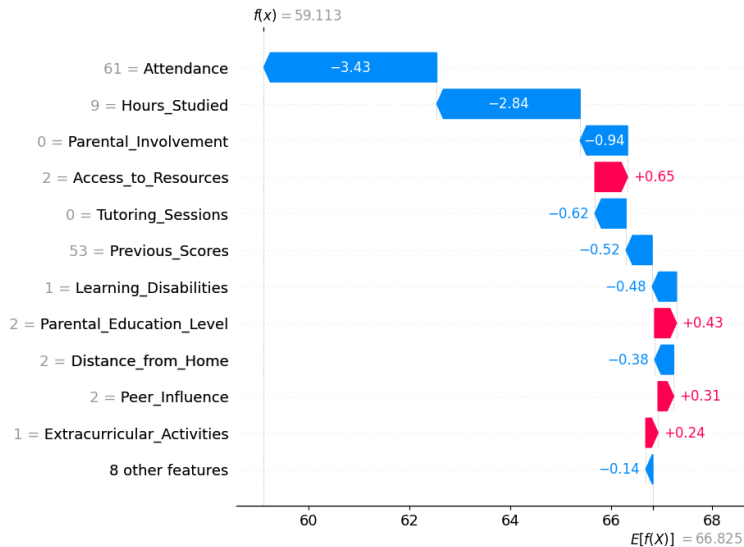
Veamos un caso real de explicabilidad con SHAP

Explicabilidad **local**: Para estudiar casos de estudiantes concretos

Estudiante con nota baja:



Estudiante con nota baja:



Este gráfico de tipo waterfall (cascada) muestra cómo se parte del valor medio de predicción del modelo

$E[f(x)] = 66.825$ y, paso a paso, cada característica del estudiante va restando o sumando puntos hasta llegar al valor final predicho $f(x) = 59.113$, que es una nota baja.

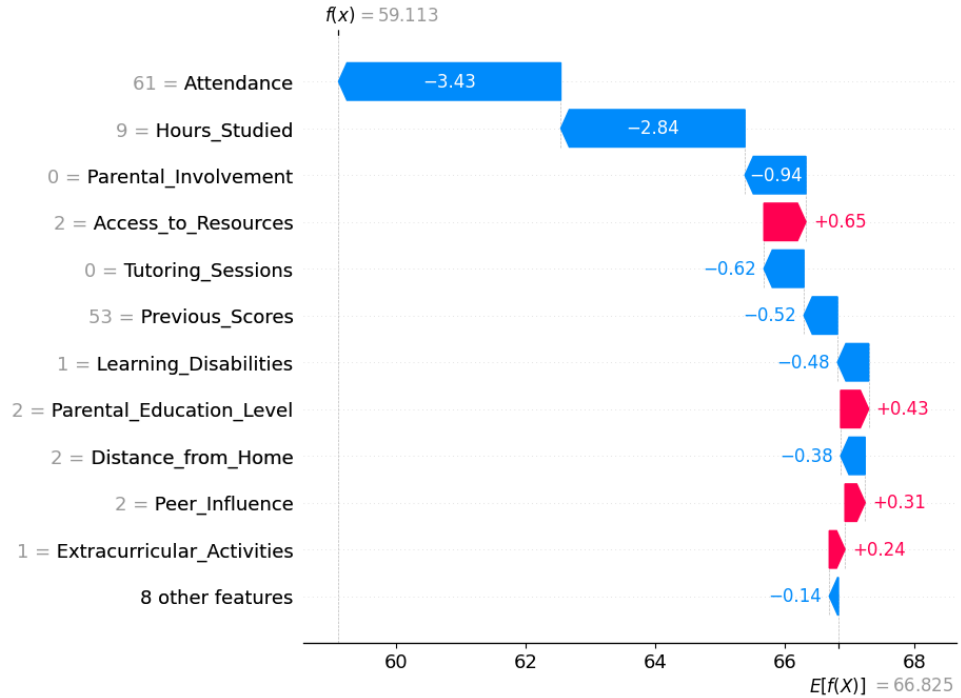
- Barras azules → factores que bajan la predicción (contribuyen a una nota baja).
- Barras rosas → factores que suben la predicción (mitigan un poco la nota baja).

Estudiante con nota baja:

Interpretación paso a paso

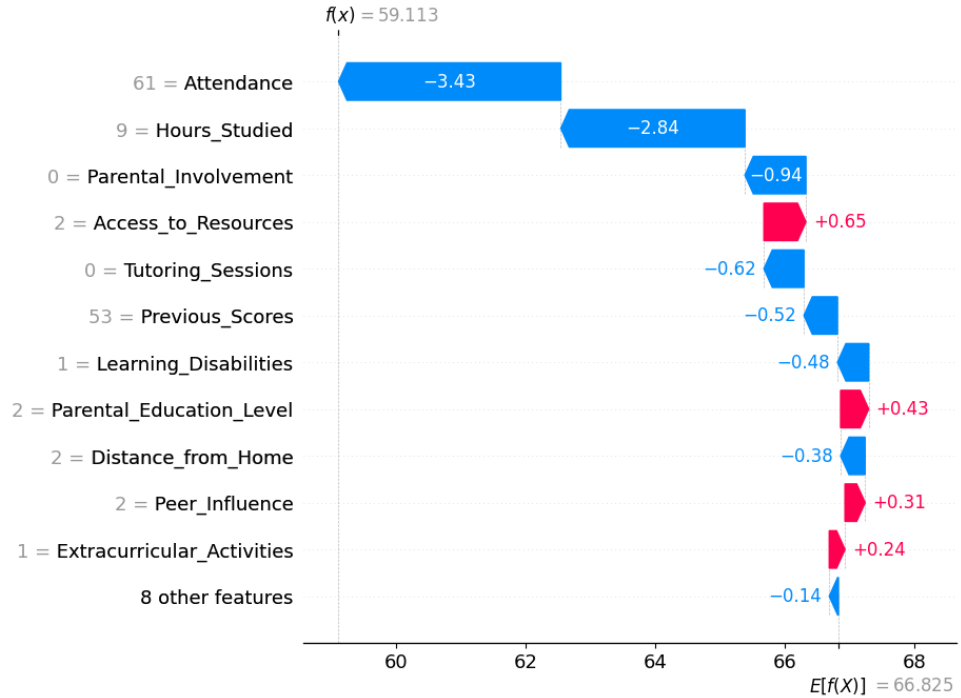
● Factores que reducen la nota:

- 1.Attendance (-3.43): El nivel de asistencia del estudiante fue bajo (valor = 61), lo cual tuvo un gran impacto negativo.
- 3.Hours_Studied (-2.84): Solo estudió 9 horas, lo que también contribuyó significativamente a su baja nota.
- 4.Parental_Involvement (-0.94): Sin implicación parental (valor = 0), otro factor negativo.
- 5.Tutoring_Sessions (-0.62): No recibió sesiones de refuerzo.
- 6.Previous_Scores (-0.52): Su historial académico fue relativamente bajo (nota previa = 53).
- 7.Learning_Disabilities (-0.48): Tiene algún tipo de dificultad de aprendizaje (valor = 1).
- 8.Distance_from_Home (-0.38): Vive a cierta distancia del centro (valor = 2), lo cual parece influir negativamente.
- 9.Extracurricular_Activities (-0.14): Participación baja en actividades extracurriculares (valor = 1)



Estudiante con nota baja:

Interpretación paso a paso



Factores que subieron un poco la nota:

Access_to_Resources (+0.65): Accede a buenos recursos (valor = 2).

Parental_Education_Level (+0.43): Buen nivel educativo de los padres.

Peer_Influence (+0.31): Influencia positiva del entorno.

8 other features (+0.24): Otros factores menores también aportaron algo positivo.