
Predicción del Rendimiento Académico en Secundaria mediante Aprendizaje Automático

Álvaro Rodríguez Mesa
2ºIMAT B ICAI
202304364@alu.comillas.edu

Abstract

Este informe muestra la predicción de la nota final de estudiantes de secundaria mediante modelos de aprendizaje automático. Se siguen los cinco pasos fundamentales de modelado y se presentan resultados visuales e interpretaciones para apoyar la toma de decisiones educativas.

1 Introducción

El rendimiento académico de los estudiantes está influenciado por numerosos factores. Comprenderlos puede ser clave para mejorar la calidad educativa. Este informe muestra la predicción de la nota final de estudiantes de secundaria mediante modelos de aprendizaje automático y mediante dichas técnicas con sus respectivas estadísticas y gráficas propone soluciones y predicciones representativas para hacer cambios. En este proyecto se han seguido los 5 pasos (dedos de la mano) del primer tema de forma clara. Entre estos pasos se encuentra la recogida de datos (cargar archivos dados), el preprocesamiento y limpieza de dichos datos, la elección del método (se trata de escoger el modelo más adecuado y que mejores resultados ofrezca) y el posterior entrenamiento de los modelos escogidos junto con su adaptación de parámetros. Finalmente, se generaliza y como paso adicional de este proyecto se implementa una parte creativa.

2 Descripción del conjunto de datos

Los datos incluyen información académica y socio-demográfica de estudiantes de dos institutos de Madrid. La variable objetivo es la nota final (T3) en Matemáticas o Lengua. Entre las variables se encuentran: sexo, edad, nivel educativo de los padres, apoyo escolar y familiar, consumo de alcohol, salud, faltas, etc. En el enunciado se encuentran dos archivos (test y otro para train y validation) con los que se trabajan dichas variables. Entre estas, algunas son de mayor importancia que otras para la predicción de la nota final T3, esto se podrá apreciar posteriormente gracias a diversas gráficas de importancia de variables.

3 Análisis exploratorio

Previo a entrenar los modelos y sacar conclusiones se analizan y exploran los datos dados. Primeramente, en el preproceso y limpieza de datos se hace un `data.info()`, `data.describe()` y `data.isnull().sum()` para ver los tipos de datos que se manejan, sus estadísticas principales y sobre todo, si existen o no datos nulos. Para este tipo de datasets no muy grandes, se consideran los datos nulos como informativos. Es por esto, que no se deben de eliminar y se opta por sustituirlos por la media. Además se muestra mediante un histograma como es la variable a predecir (ver figura 1) para hacerse a una idea de los datos con los que se van a trabajar.

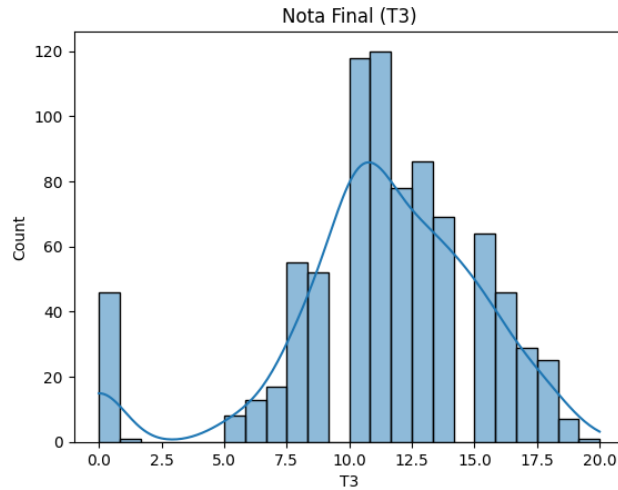


Figure 1: Histograma nota final (T3)

Tras esto, se analizan los posibles datos atípicos (outliers) de forma informativa y únicamente para saber la forma de los datos (análisis básico, ver figura 2). No se tratan de forma especial dichos datos ya que los modelos escogidos (Random Forest tanto en i como ii, se ver posteriormente) son robustos ante estos datos atípicos/outliers.

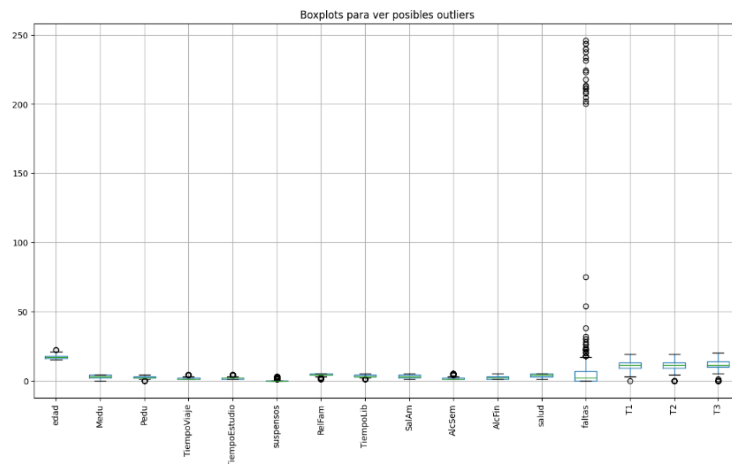


Figure 2: Boxplots variables

Después, se rellenan los datos nulos con `IterativeImputer` y se vuelven a ejecutar `data.info()`, `data.describe()` y `data.isnull().sum()` para verificar que dichos nulos han sido sustituidos correctamente por su media. Se ha usado esta función de `scikit-learn` en vez de técnicas simples como imputar con la media, ya que utiliza modelos de regresión para predecir iterativamente los valores faltantes de cada columna numérica basándose en los valores presentes y en las relaciones entre las demás variables del dataset. El proceso comienza imputando de manera básica (por ejemplo, con la media) y luego, en sucesivas iteraciones, entrena un modelo predictivo para cada columna con datos faltantes, actualizando de forma progresiva los valores estimados hasta que convergen o se alcanza el número máximo de iteraciones. Esto se traduce en imputaciones más realistas y representativas, especialmente cuando las variables están correlacionadas, reduciendo el sesgo y mejorando la calidad de los datos de entrada para cualquier análisis posterior. En el código, tras importar `IterativeImputer`, se seleccionan las columnas numéricas del dataframe y luego se ajusta y transforma el conjunto utilizando `imputer.fit.transform`, reemplazando de manera automática todos los valores perdidos por estimaciones informadas por el propio patrón de los datos. Finalmente, se transforman

las variables categóricas a cuantitativas mediante onehotencoding y se verifica que las clases están balanceadas (se confirma con la distribución normal resultante).

4 Modelado predictivo

Para escoger el modelo se ha hecho cross-validation para varios modelos y se han obtenido las métricas básicas RMSE y R^2 para compararlas. Este cross-validation se ha hecho con un n de Kfolds de 5 y un random_state de 42 (datos inicializados en estos valores por ser los más típicos y sencillos para este método comparativo), se comparan los modelos RandomForest, GradientBoosting, LinearRegression, DecisionTree y SVR. Los resultados obtenidos son los siguientes:

Modelo: RandomForest - R^2 promedio: 0.8549 - RMSE promedio: 1.4943

Modelo: GradientBoosting - R^2 promedio: 0.8611 - RMSE promedio: 1.4678

Modelo: LinearRegression - R^2 promedio: 0.8292 - RMSE promedio: 1.6403

Modelo: DecisionTree - R^2 promedio: 0.7126 - RMSE promedio: 2.1011

Modelo: SVR - R^2 promedio: 0.7791 - RMSE promedio: 1.8673

Analizando los resultados, aunque el mejor desempeño lo tiene Gradient Boosting, Random Forest también muestra buenos resultados ($R^2=0.8549$, $RMSE=1.4943$). Se escoge Random Forest porque es menos propenso al sobreajuste por su estructura de ensamblado de árboles independientes y típicamente requiere menos ajuste fino de hiperparámetros, lo que lo hace más robusto ante diferentes tipos de datos y más fácil de usar si se dispone de poco tiempo para ajuste. Además, Random Forest suele ser más rápido de entrenar y menos sensible a los cambios en los datos o a la presencia de ruido, mientras que Gradient Boosting, aunque a veces logra mejores resultados, puede ser más vulnerable a sobreajuste y requiere un ajuste cuidadoso para evitarlo. Si las diferencias de desempeño entre ambos son pequeñas, Random Forest se convierte en la opción preferible por su mayor estabilidad, simplicidad y rapidez.

4.1 Modelo (i): con T1 y T2

Para este primer modelo se han usado todas las variables y se ha empleado RandomForest por lo mencionado anterior. Como mayor explicación, se ha escogido este modelo ya que Random Forest es un algoritmo que combina múltiples árboles de decisión para mejorar la precisión y estabilidad de las predicciones. Sus principales ventajas (aplicadas para este proyecto) son su resistencia al sobreajuste, y su habilidad para trabajar con datos faltantes o variables categóricas. Además, Random Forest puede estimar la importancia de cada variable en la predicción, ofreciendo resultados robustos y fáciles de interpretar en comparación con otros modelos individuales. Para medir cómo de preciso y fiable es este modelo se calculan las métricas básicas: RMSE y R^2 . Los resultados son buenos ya que tiene un error promedio de apenas 1.36 puntos sobre 20 y explica el 88.9% de la variabilidad del rendimiento final de los estudiantes. Para apoyar esto se muestra la curva de aprendizaje (ver figura 3) donde los errores de entrenamiento y validación tienden a parecerse:

4.2 Modelo (ii): sin T1 ni T2

Para este segundo modelo se vuelve a emplear RandomForest pero quitando las variables T1 y T2. En este caso se obtienen unos resultados peores: RMSE: 3.2 y R^2 : 0.38. Finalmente, esto se muestra con los valores tan altos de error que se aprecian en la escala del eje y de la curva de calibración (ver figura 4):

Comentario adicional: Al no disponer de T1 y T2 (motivo de tanto error) se muestra con un histograma la importancia del resto de variables donde destacan suspensos y faltas (ver figura 5):

En el apéndice se muestran las gráficas comparativas de ambos modelos.

5 Modelo optimizado

En este apartado se busca optimizar el modelo ii mediante cross-validation, en este caso se ha escogido RandomizedSearch del GradientBoostingRegressor ya que puede superar a Random

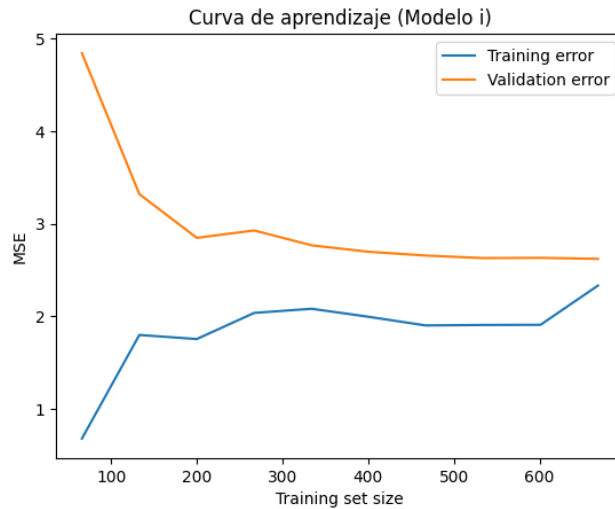


Figure 3: Curva de aprendizaje de modelo i

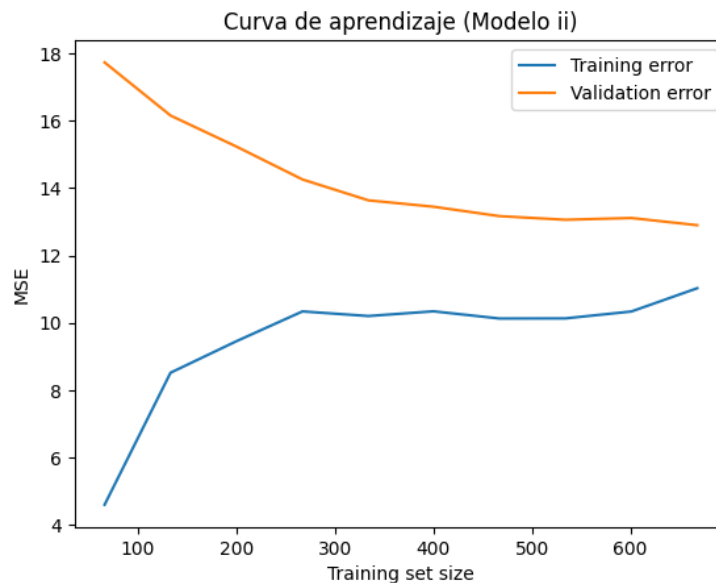


Figure 4: Curva de aprendizaje modelo ii

Forest simple porque, en lugar de simplemente promediar muchos árboles independientes, va construyendo cada nuevo árbol enfocándose específicamente en corregir los errores cometidos por los árboles anteriores. Gracias a esto, logra capturar patrones complejos y relaciones no evidentes en los datos de forma más precisa. Además, Gradient Boosting es muy flexible y permite ajustar parámetros finos como la tasa de aprendizaje o la cantidad máxima de errores a corregir, lo que puede reducir el sobreajuste y mejorar la capacidad de generalización del modelo. También, emplea un rango continuo en vez de categórico para encontrar los mejores hiperparámetros adaptables en el modelo. Al utilizar este modelo y carecer de una función de coste no se puede hacer el descenso gradiente que buscaría el mínimo tras varias derivaciones.

Los parámetros optimizados son:

- n_estimators: rango (100, 300),
- max_depth: rango (3, 20),

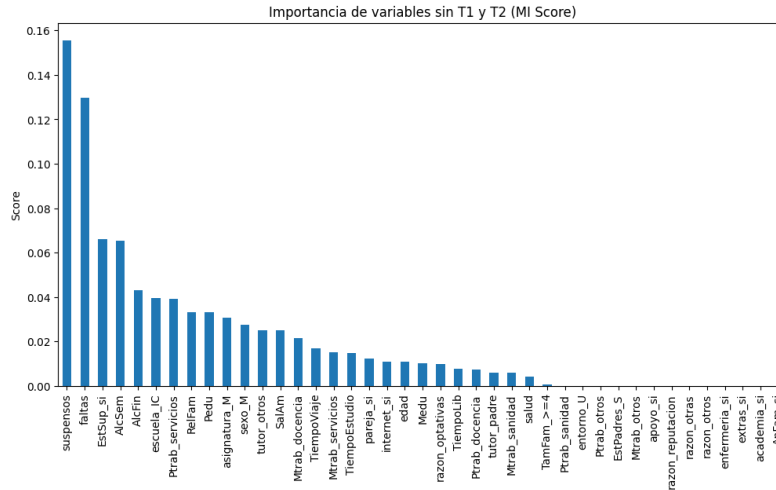


Figure 5: Histograma de importancia de variables

- learning_rate: rango (0.01, 0.3),
- subsample: rango (0.6, 0.4),
- min_samples_split: rango (2, 10),
- min_samples_leaf: rango (1, 10)

Este modelo obtiene mejoras en los resultados respecto al modelo ii normal. Como se puede apreciar, los resultados han sido de: RMSE: 3.02 y R2: 0.45. La curva de aprendizaje es muy similar a la anterior pero se pueden apreciar las mejoras en las gráficas comparativas ya vistas en el apéndice.

Además, se vuelven a analizar las variables más importantes saliendo las mismas que antes pero en diferente orden (ver figura 6). Esta vez, se han mostrado más gráficas para verificar esta hipótesis como la matriz de correlación o violinplots (ver apéndice).

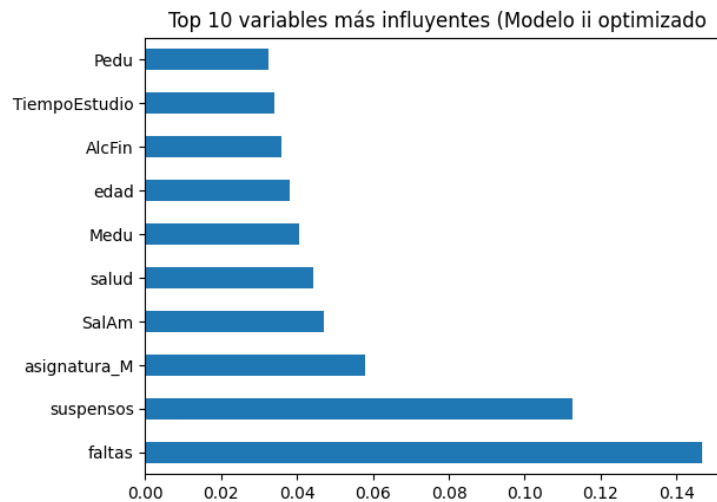


Figure 6: Importancia variables modelo ii optimizado

6 Interpretación de resultados

Una vez vistas las gráficas del apéndice se pueden interpretar sus resultados y tratar de obtener unas soluciones concluyentes para aquellos estudiantes nece-

sitados de ayuda para mejorar su rendimiento final (T3). Como soluciones se proponen:

En cuanto al aspecto relacionado con las faltas, una solución que podría frenar y bajar el índice de notas bajas por número alto de faltas es poner un sistema de asistencia más duro. Se puede proponer un porcentaje de clases obligatorias a las que ir o un número mínimo de asistencia a clases para poder presentarse a la convocatoria final de la asignatura.

Para el tema de los suspensos, como solución se puede proponer un sistema de apoyo típico de centros escolares en los que aquellos alumnos con asignaturas suspensas tengan clases de refuerzo o ayuda de profesores particulares adjudicados. Clases obligatorias en horario no lectivo puede incentivar a dichos alumnos a estudiar más y reducir el número de suspensos.

7 Parte creativa: aprendizaje no supervisado y explicabilidad

Para explorar más a fondo estos datos se aplican como técnicas de aprendizaje no supervisado PCA y KMeans. PCA es útil para reducir la dimensionalidad y obtener efectividad y simplicidad al mismo tiempo, se ha usado un número de componentes de 2 para reducir la dimensionalidad a 2 (el entorno bidimensional es sencillo para interpretar resultados y sacar conclusiones) Además, esta técnica es de gran utilidad para representar resultados en contextos académicos. También se emplea KMeans para agrupar los estudiantes en clusters con características simples y obtener patrones ocultos.

Para realizar PCA se estandarizan los datos (condición indispensable) y en cuanto a KMeans se realizan tanto el método del codo como el de la silueta para obtener el número óptimo de clusters (en este caso 3). En el método del codo se puede intuir que es 3 ya que es el valor a partir del cual la inercia deja de disminuir significativamente al aumentar el número de clústeres. Sin embargo, en el método de la silueta se puede ver que es 3 de forma muy clara ya que es el valor en el que se maximiza la puntuación de la silueta. El resultado es el siguiente (ver figura 7):

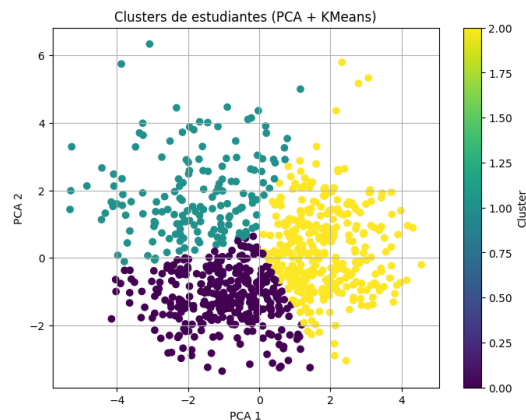


Figure 7: Clusters de estudiantes

8 Conclusiones

Finalmente, se hacen las predicciones de T3 con el archivo dado tanto para el modelo i como para el modelo ii optimizado. Los resultados se pueden ver en el archivo csv creado. Como resumen de cierre, los modelos predictivos permiten anticipar el rendimiento escolar con buena precisión. El modelo (i) es más preciso, pero el (ii) permite intervenir desde el inicio del curso. Factores como el número de faltas y de suspensos son críticos para el desarrollo de las calificaciones finales del estudiante. El modelo optimizado y las herramientas de interpretabilidad refuerzan la utilidad del enfoque. Con ayuda de las gráficas informativas sobre la importancia de las variables se han propuesto diversas soluciones para optimizar el rendimiento de los estudiantes. Algunas de estas soluciones ya mencionadas son las de incluir un sistema de asistencia más estricto así como una implementación

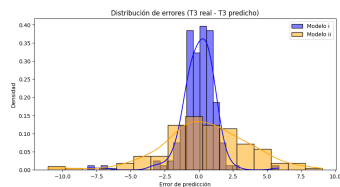


Figure 8: Imagen 10

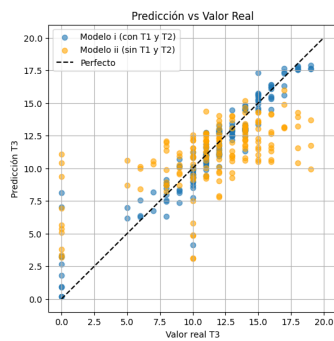


Figure 9: Imagen 11

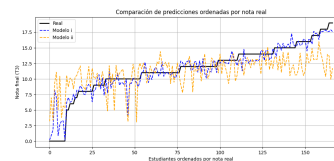


Figure 10: Imagen 12

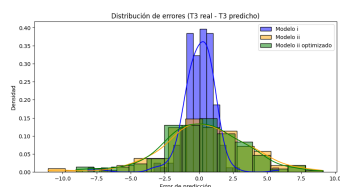


Figure 11: Imagen 13

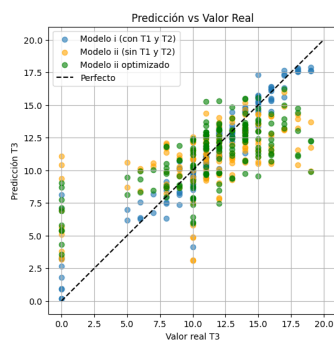


Figure 12: Imagen 14

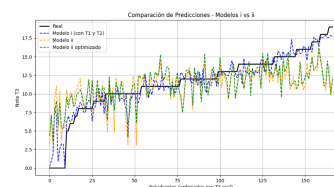


Figure 13: Imagen 15

de clases de refuerzo o tutorías personales para analizar personalmente a los estudiantes y aumentar sus horas de estudio y por supuesto, su eficiencia.

9 Apéndice

En esta sección del apéndice se muestran el resto de imágenes del notebook. Entre estas se muestran: gráficas comparativas modelo i y modelo ii (figuras 8, 9 y 10), gráficas comparativas modelo i, modelo ii y modelo ii optimizado (figuras 11, 12 y 13), gráficas de importancia de variables (ver figuras 14, 15 y 16) y métodos clusters kmeans (ver figuras 17 y 18).

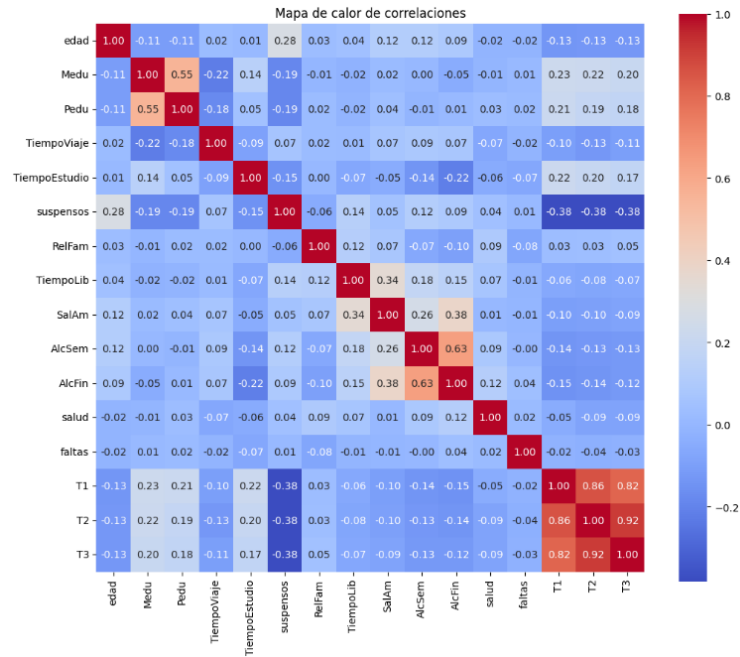


Figure 14: Imagen 16

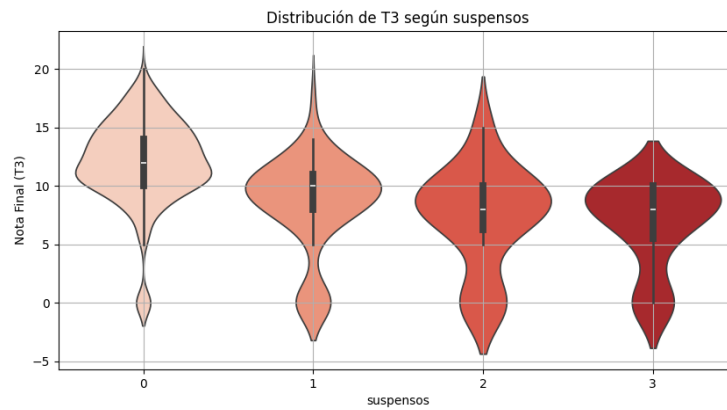


Figure 15: Imagen 17

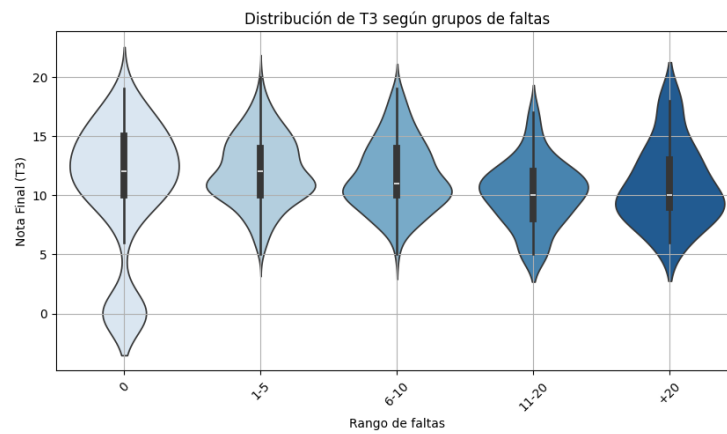


Figure 16: Imagen 18

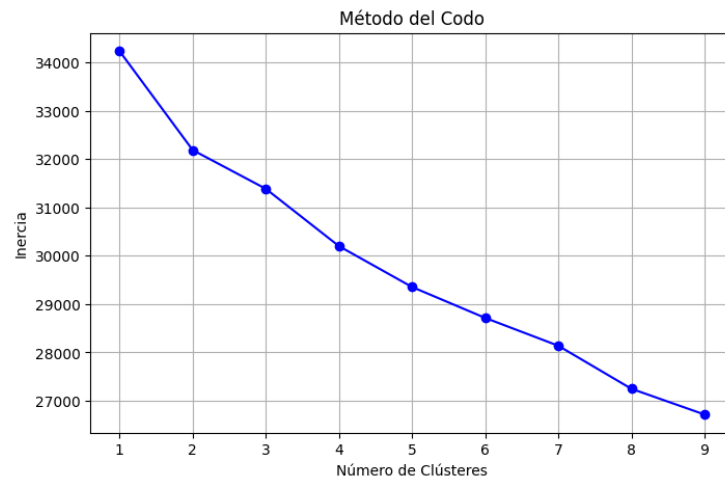


Figure 17: Imagen 19

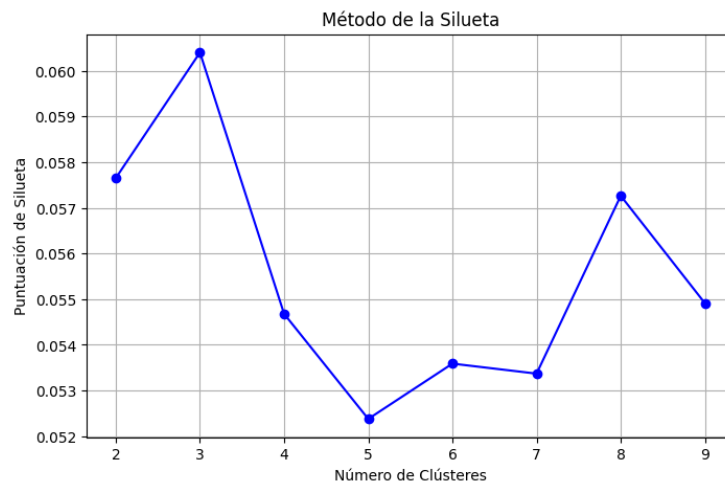


Figure 18: Imagen 20