

1. Define los términos Underfitting y overfitting, según se plantea en la sección 5.

Captura patrones que no se repetirán en el futuro lo que conducirá a predicciones menos precisas, no funciona bien en la validación y otros datos nuevos

No captura patrones relevantes lo que igualmente conduce a predicciones menos precisas, por lo que no se desempeña correctamente en los datos de entrenamiento

2. En tus propias palabras, explica de manera sencilla, la lógica de la regresión Random Forest según se plantea en la sección 6

Utiliza no solo un árbol de predicción, va utilizar varios arboles y sacara un promedio de todos ellos y esto hace que la predicción sea mejor, es decir que entre más arboles tenga trabajara mejor y con menos será lo contrario

3. ¿Qué ventaja tiene la regresión Random Forest en comparación con la regresión Decission Tree?

Es más exacta la predicción porque evita que haya pocos valores, y puede funcionas mejor incluso son el ajuste de los árboles de decisión

4. ¿El error medio absoluto obtenido en la regresión Random Forest, es menor, mayor o igual que el error medio absoluto obtenido en la regresión DecissionTree? Justifica tu respuesta.

Es menor el MAE en el Random Forest debido a que elimina las predicciones sobrantes, precisamente busca que no haya pocos datos, ni a la vez datos de más, para que sus resultados sean lo más precisos.

Ejecución del Script:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
import pandas as pd

from google.colab import files
uploades = files.upload()

melbourne_data = pd.read_csv('melb_data.csv')
melbourne_data = melbourne_data.dropna(axis=0)
y = melbourne_data.Price
melbourne_features = ['Rooms', 'Bathroom', 'Landsize', 'Latitude', 'Longitude']
X = melbourne_data[melbourne_features]

train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 0)
melbourne_model = RandomForestRegressor(random_state=1)
melbourne_model.fit(train_X, train_y)
val_predictions = melbourne_model.predict(val_X)
print(mean_absolute_error(val_y, val_predictions))
```



Examinar... melb_data.csv

melb_data.csv(application/vnd.ms-excel) - 13839 bytes, last modified: n/a - 100% done
Saving melb_data.csv to melb_data.csv
210016.0