Proyecto IA

Inteligencia Artificial

Departamento de Electrónica

Profesor: Francisco Carlos Calderón Bocanegra

Video Games Sales Prediction

Facultad de Ingeniería, Pontificia Universidad Javeriana, Bogotá D.C, Colombia.
Sergio Enrique González. sergioegonzalez@javeriana.edu.co

En este proyecto se encontrará la predicción de ventas de videojuegos en Estados Unidos basado en el dataset obtenido en la página de Kaggle. Este dataset se puede descargar y encontrar en el siguiente enlace: https://www.kaggle.com/datasets/migeruj/videogames-predictive-model

Ilustración 1. Dataset obtenido en Kaggle

Este proyecto utilizo la herramienta Visual Studio Code con un lenguaje de programación en Python, en donde se realizaron diferentes tipos de pruebas y modelos para obtener el mejor método de análisis el cual este problema de aprendizaje correspondía a uno de no supervisado y de regresión.

1. El archivo fue creado en .ipynb
2. Se procedió hacer la limpieza respectiva del dataset.
3. Se normalizaron los datos donde posteriormente se realizó PCA
4. Al comprobar los datos obtenidos en PCA fue necesario utilizar los 5 pliegues ya que cada uno de los componentes de la variable X aportan información importante en la predicción que se va a desarrollar durante el código.

PCA

```
PCA = decomposition.PCA(n_components=5)
PCA.fit(X_train)
X_train = PCA.transform(X_train)

print("Pesos n de PCA:",PCA.explained_variance_ratio_)
print("Suma total de pesos n de PCA",sum(PCA.explained_variance_ratio_))
```

```
Pesos n de PCA: [0.24404026 0.21613138 0.20139132 0.17850932 0.15992771]
Suma total de pesos n de PCA 0.9999999999999999
```

Ilustración 2.PCA de la variable X

5. Después se realizó un gridsearch y cross-validación utilizando GridSearchCV para optimizar los hiperparametros de los diferentes modelos para R2, arrojando los mejores parámetros del modelo seleccionado para obtener el mejor Score.

6. Los modelos utilizados fueron KNN donde se varió la métrica y peso de la distancia, Maquina de vectores de soporte "SMV" y para finalizar se utilizó Decision Tree se varió el criterio y profundidad de los árboles

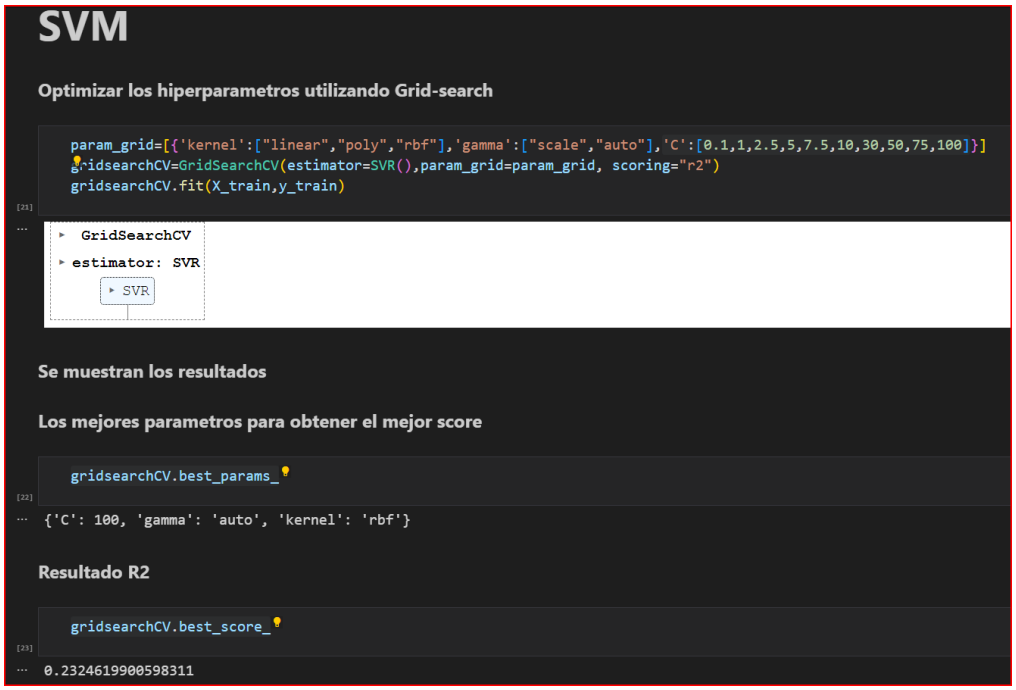

Ilustración 3. Máquina de vectores de Soporte con GridSearch

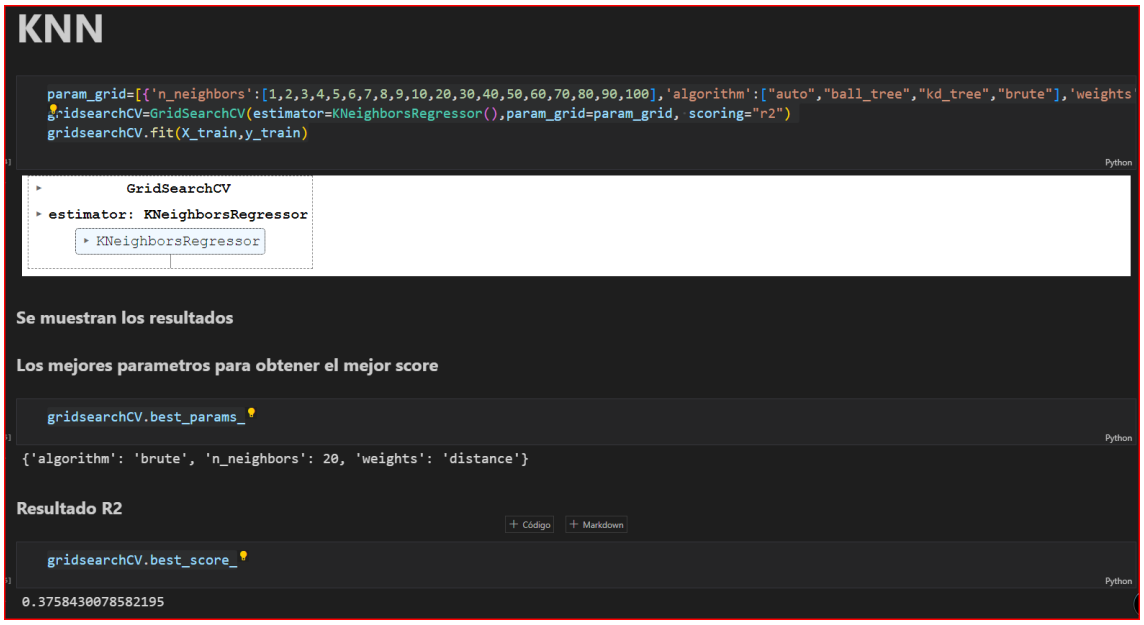

Ilustración 4. K-Vecinos más cercanos con GridSearchCV

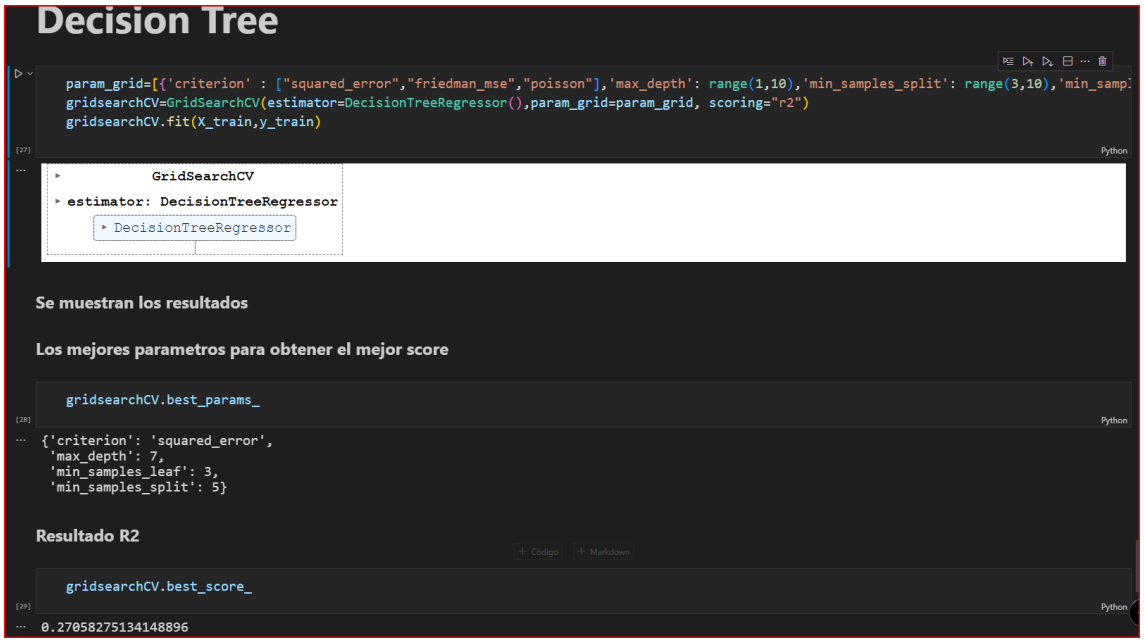

Ilustración 5. Decision Tree con GridSearchCV

Los datos proporcionados en el conjunto de datos están bastante descorrelacionados entre sí, para obtener un mejor resultado el conjunto de datos puede mejorar si desea analizarlo mediante de un modelo de aprendizaje automático.

Enlace para video con explicación más detallada:
https://www.youtube.com/watch?v=9gelV7wpy8Q