File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Distribución porcentual de la variable objetivo
df['left'].value_counts(normalize=True) * 100
```

[7] ✓ 0.0s

```
left
0    76.191746
1    23.808254
Name: proportion, dtype: float64
```

El análisis de la variable objetivo muestra que la base de datos presenta un desbalance entre las clases, siendo mayor el número de empleados que permanecen en empresa en comparación con aquellos que la abandonan. Este comportamiento es común en problemas de rotación de personal y debe considerarse al evaluar el desempeño de los modelos de clasificación, ya que métricas como la precisión pueden verse influenciadas por dicho desbalance.

Preparación de los datos para el modelado

Antes de entrenar los modelos de Máquinas de Soporte Vectorial, es necesario preparar el conjunto de datos. Esta etapa incluye la transformación de variables cate la separación de la variable objetivo y el escalado de las variables numéricas.

La correcta preparación de los datos es especialmente importante en modelos SVM, ya que estos son sensibles a la escala de las variables.

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

la separación de la variable objetivo y el escalado de las variables numéricas.

La correcta preparación de los datos es especialmente importante en modelos SVM, ya que estos son sensibles a la escala de las variables.

```
# Separación de variables predictoras y variable objetivo
X = df.drop('left', axis=1)
y = df['left']
```

[8] ✓ 0.0s

Generate + Code + Markdown

```
# Codificación de variables categóricas
X_encoded = pd.get_dummies(X, drop_first=True)
```

[9] ✓ 0.0s

La transformación de las variables categóricas mediante codificación permite convertir la información cualitativa en un formato numérico adecuado para los model
Machine Learning. Con esto, el conjunto de datos queda listo para aplicar técnicas de escalado y posteriormente entrenar los modelos de clasificación.

Escalado de variables y partición del conjunto de datos

Las Máquinas de Soporte Vectorial son sensibles a la escala de las variables, por lo que es necesario aplicar un proceso de estandarización antes de entrenar los mo
El escalado permite que todas las variables contribuyan de manera equilibrada al proceso de aprendizaje.

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Las Máquinas de Soporte Vectorial son sensibles a la escala de las variables, por lo que es necesario aplicar un proceso de estandarización antes de entrenar los mo
El escalado permite que todas las variables contribuyan de manera equilibrada al proceso de aprendizaje.

Posteriormente, el conjunto de datos se divide en subconjuntos de entrenamiento y prueba, con el fin de evaluar el desempeño de los modelos sobre datos no util
durante el entrenamiento.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

[10] ✓ 0.6s

```
# División del conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded, y, test_size=0.3, random_state=42, stratify=y
)
```

[11] ✓ 0.0s

```
# Escalado de las variables
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
# Escalado de las variables
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[12] ✓ 0.0s

Generate + Code + Markdown

Con el escalado de las variables se asegura que todas las características se encuentren en una misma escala, lo cual es fundamental para el correcto funcionamient modelos SVM. Asimismo, la partición del conjunto de datos permite evaluar de manera objetiva el desempeño de los modelos al utilizar información no vista durar entrenamiento.

Entrenamiento de modelos SVM con distintos kernels

En esta etapa se entrenan distintos modelos de Máquinas de Soporte Vectorial utilizando los cuatro tipos de *kernel* más comunes: lineal, polinomial, radial (RBF) y sigmoide. El objetivo es comparar su desempeño y analizar cuál de ellos resulta más adecuado para el problema de clasificación planteado.

```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

[13] ✓ 2.0s

Generate + Code + Markdown

Modelo SVM con kernel lineal

En este primer modelo se utiliza una Máquina de Soporte Vectorial con kernel lineal. Este tipo de kernel es adecuado cuando las clases pueden separarse de forma aproximadamente lineal en el espacio de características. Su simplicidad permite establecer una línea base de desempeño para comparar con kernels más complejo

```
# Entrenamiento del modelo SVM con kernel lineal
svm_linear = SVC(kernel='linear', random_state=42)
svm_linear.fit(X_train_scaled, y_train)
```

[14] ✓ 2.2s

```
SVC
▸ Parameters
```

```
# Predicciones con el modelo lineal
y_pred_linear = svm_linear.predict(X_test_scaled)
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Predicciones con el modelo lineal
y_pred_linear = svm_linear.predict(X_test_scaled)
```

[15] ✓ 0.2s

Generate + Code + Markdown

```python
# Matriz de confusión
cm_linear = confusion_matrix(y_test, y_pred_linear)
cm_linear
```

[16] ✓ 0.0s

```
array([[3202,  227],
       [ 789,  282]])
```

```python
# Mapa de calor de la matriz de confusión
plt.figure()
sns.heatmap(cm_linear, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicción')
plt.ylabel('Valor real')
plt.title('Matriz de confusión - SVM Kernel Lineal')
plt.show()
```

[17] ✓ 0.2s

Matriz de confusión - SVM Kernel Lineal

File Edit Selection View Go Run ···

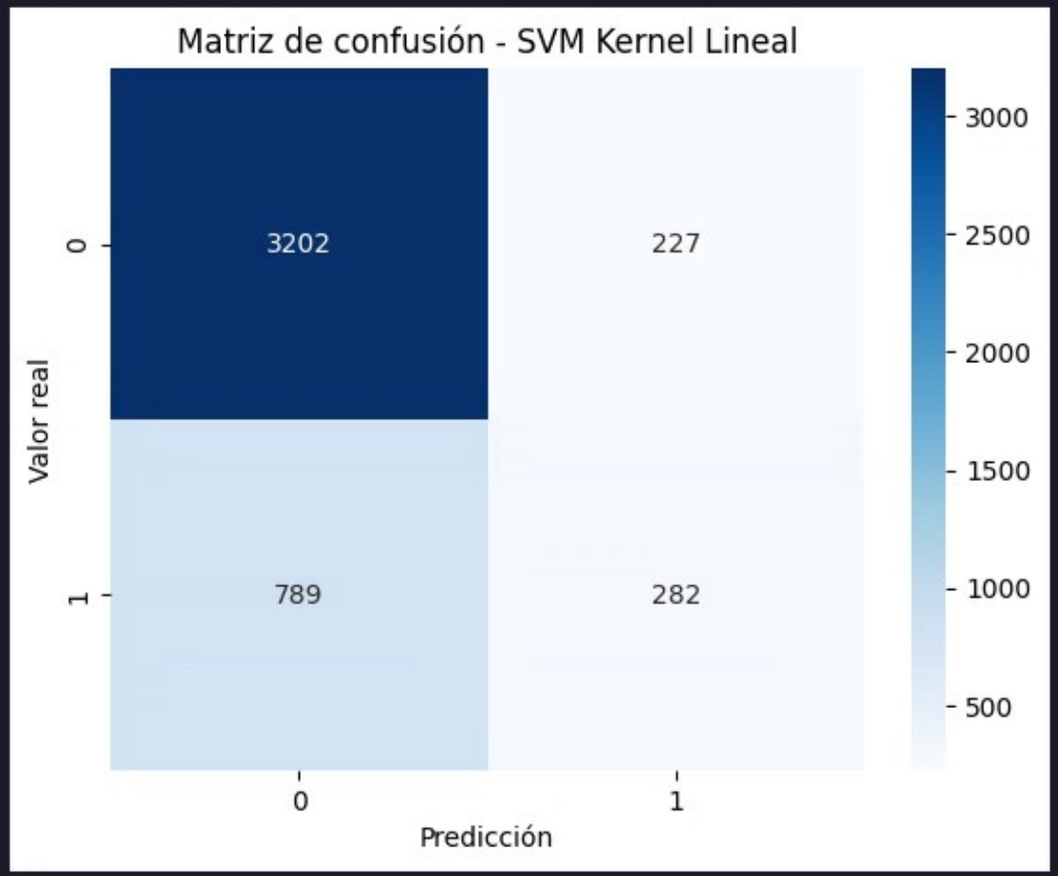
Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

[17] ✓ 0.2s




```
print(classification_report(y_test, y_pred_lineal))
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Generate + Code + Markdown

```python
# Reporte de clasificación
print(classification_report(y_test, y_pred_linear))
```

[18] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.80      0.93      0.86      3429
           1       0.55      0.26      0.36      1071

    accuracy                           0.77      4500
   macro avg       0.68      0.60      0.61      4500
weighted avg       0.74      0.77      0.74      4500
```

Los resultados del modelo SVM con kernel lineal muestran el desempeño del clasificador al separar las clases mediante una frontera lineal. A partir de la matriz de confusión y el reporte de clasificación es posible evaluar la capacidad del modelo para identificar correctamente a los empleados que abandonan la empresa y aqu que permanecen en ella.

Este modelo sirve como referencia inicial para comparar el impacto de kernels más complejos en la capacidad predictiva del algoritmo.

Modelo SVM con kernel polinomial

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Modelo SVM con kernel polinomial

En este segundo modelo se utiliza una Máquina de Soporte Vectorial con kernel polinomial. Este tipo de kernel permite capturar relaciones no lineales entre las var al proyectar los datos a un espacio de mayor dimensión. Su uso resulta útil cuando la separación entre clases no puede representarse adecuadamente mediante un frontera lineal.

```
# Entrenamiento del modelo SVM con kernel polinomial
svm_poly = SVC(kernel='poly', degree=3, random_state=42)
svm_poly.fit(X_train_scaled, y_train)
```

[19] ✓ 0.7s

```
SVC
▸ Parameters
```

```
# Predicciones con el modelo polinomial
y_pred_poly = svm_poly.predict(X_test_scaled)
```

[20] ✓ 0.1s

```
# Matriz de confusión
cm_poly = confusion_matrix(y_test, y_pred_poly)
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Matriz de confusión
cm_poly = confusion_matrix(y_test, y_pred_poly)
cm_poly
```

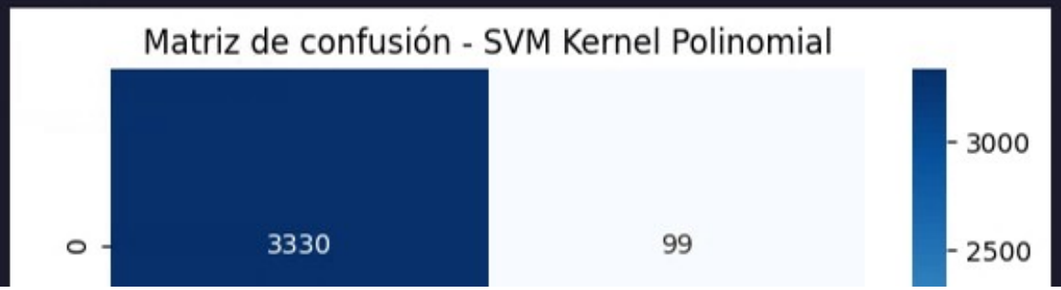
[21] ✓ 0.0s

```
array([[3330,   99],
       [ 179,  892]])
```

Generate + Code + Markdown

```python
# Mapa de calor de la matriz de confusión
plt.figure()
sns.heatmap(cm_poly, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicción')
plt.ylabel('Valor real')
plt.title('Matriz de confusión - SVM Kernel Polinomial')
plt.show()
```

[22] ✓ 0.0s



File Edit Selection View Go Run ···

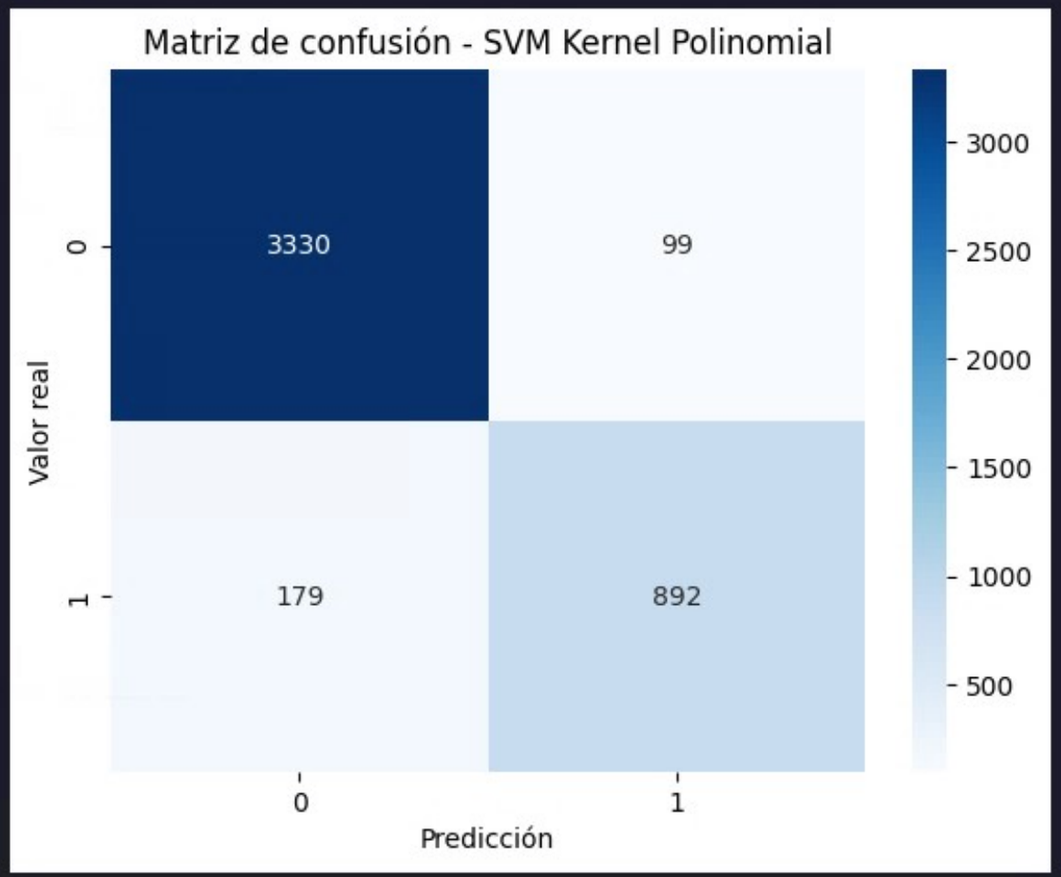
Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

[22] ✓ 0.0s



File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Reporte de clasificación
print(classification_report(y_test, y_pred_poly))
```

[23] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.95      0.97      0.96      3429
           1       0.90      0.83      0.87      1071

    accuracy                           0.94      4500
   macro avg       0.92      0.90      0.91      4500
weighted avg       0.94      0.94      0.94      4500
```

El modelo SVM con kernel polinomial permite capturar relaciones más complejas entre las variables en comparación con el kernel lineal. A partir de los resultados obtenidos, se puede evaluar si la incorporación de no linealidades mejora la capacidad del modelo para clasificar correctamente a los empleados según su probabi abandono.

Este modelo se analiza en conjunto con los demás kernels para determinar si su mayor complejidad se traduce en un mejor desempeño predictivo.

Modelo SVM con kernel radial (RBF)

En este modelo se utiliza una Máquina de Soporte Vectorial con kernel radial (RBF). Este kernel es ampliamente utilizado en problemas de clasificación, ya que perm

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Modelo SVM con kernel radial (RBF)

En este modelo se utiliza una Máquina de Soporte Vectorial con kernel radial (RBF). Este kernel es ampliamente utilizado en problemas de clasificación, ya que perm modelar fronteras de decisión altamente no lineales y adaptarse mejor a conjuntos de datos complejos.

El kernel RBF suele ofrecer un buen balance entre flexibilidad y capacidad de generalización, por lo que resulta un candidato natural para comparar con los modelo y polinomial.

```
# Entrenamiento del modelo SVM con kernel RBF
svm_rbf = SVC(kernel='rbf', random_state=42)
svm_rbf.fit(X_train_scaled, y_train)
```

[24] ✓ 0.6s

```
SVC
▸ Parameters
```

```
# Predicciones con el modelo RBF
y_pred_rbf = svm_rbf.predict(X_test_scaled)
```

[25] ✓ 0.4s

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Predicciones con el modelo RBF
y_pred_rbf = svm_rbf.predict(X_test_scaled)
```

[25] ✓ 0.4s

```python
# Matriz de confusión
cm_rbf = confusion_matrix(y_test, y_pred_rbf)
cm_rbf
```

[26] ✓ 0.0s

```
array([[3324,  105],
       [ 108,  963]])
```

```python
# Mapa de calor de la matriz de confusión
plt.figure()
sns.heatmap(cm_rbf, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicción')
plt.ylabel('Valor real')
plt.title('Matriz de confusión - SVM Kernel RBF')
plt.show()
```

[27] ✓ 0.0s

Matriz de confusión - SVM Kernel RBF

File Edit Selection View Go Run ···

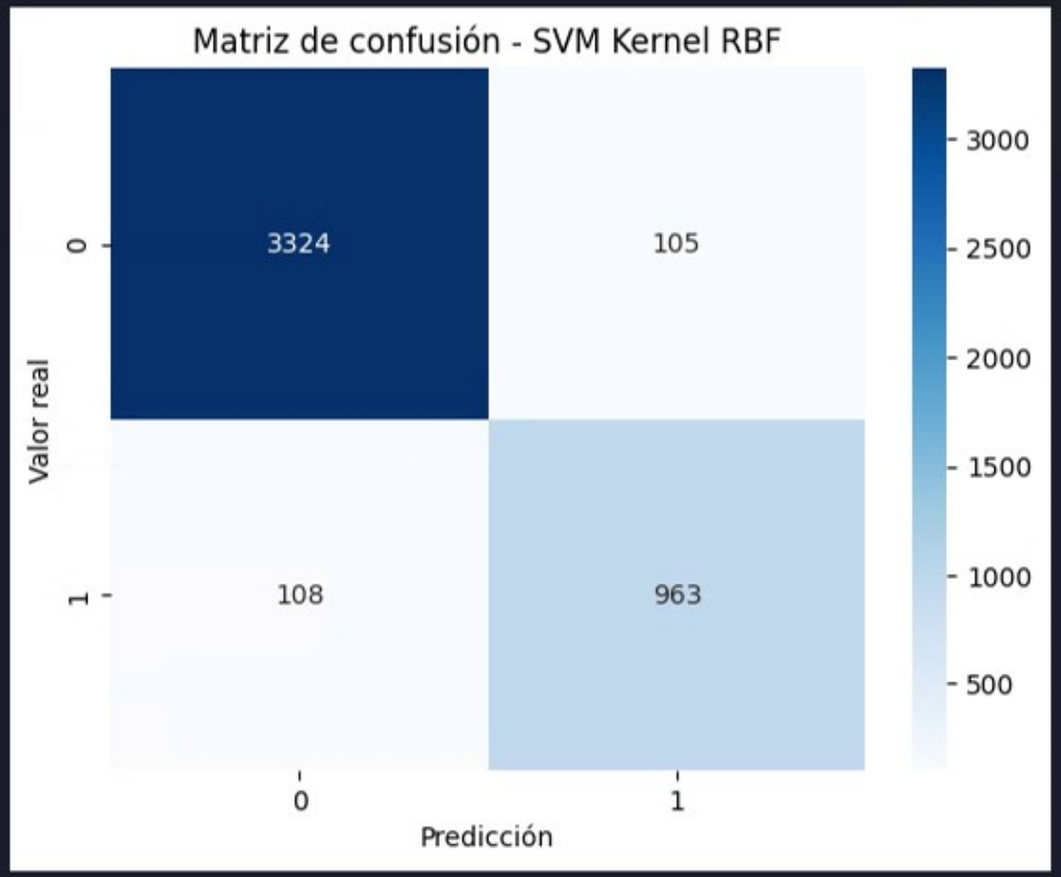
Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

[27] ✓ 0.0s




```
print(classification_report(y_test, y_pred_rbf))
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Reporte de clasificación
print(classification_report(y_test, y_pred_rbf))
```

[28] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      3429
           1       0.90      0.90      0.90      1071

    accuracy                           0.95      4500
   macro avg       0.94      0.93      0.93      4500
weighted avg       0.95      0.95      0.95      4500
```

El modelo SVM con kernel RBF muestra un comportamiento más flexible al permitir fronteras de decisión no lineales. A partir de los resultados obtenidos se puede si este enfoque mejora la identificación de empleados con riesgo de abandono, en comparación con los kernels lineal y polinomial.

El desempeño de este modelo se analizará de manera conjunta con los demás kernels para seleccionar la alternativa más adecuada para el problema planteado.

Modelo SVM con kernel sigmoide

En este modelo se emplea una Máquina de Soporte Vectorial con kernel sigmoide. Este kernel está inspirado en el comportamiento de funciones de activación utili

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Modelo SVM con kernel sigmoide

En este modelo se emplea una Máquina de Soporte Vectorial con kernel sigmoide. Este kernel está inspirado en el comportamiento de funciones de activación utili en redes neuronales y puede resultar útil en ciertos contextos específicos.

Su desempeño suele ser más sensible a la escala de los datos y a la selección de parámetros, por lo que su evaluación permite completar el análisis comparativo er distintos tipos de kernel.

```
# Entrenamiento del modelo SVM con kernel sigmoide
svm_sigmoid = SVC(kernel='sigmoid', random_state=42)
svm_sigmoid.fit(X_train_scaled, y_train)
```

[29] ✓ 1.8s

```
SVC
▸ Parameters
```

```
# Predicciones con el modelo sigmoide
y_pred_sigmoid = svm_sigmoid.predict(X_test_scaled)
```

[30] ✓ 0.5s

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Predicciones con el modelo sigmoide
y_pred_sigmoid = svm_sigmoid.predict(X_test_scaled)
```

[30] ✓ 0.5s

```python
# Matriz de confusión
cm_sigmoid = confusion_matrix(y_test, y_pred_sigmoid)
cm_sigmoid
```

[31] ✓ 0.0s

```
array([[2645,  784],
       [ 858,  213]])
```

```python
# Mapa de calor de la matriz de confusión
plt.figure()
sns.heatmap(cm_sigmoid, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicción')
plt.ylabel('Valor real')
plt.title('Matriz de confusión - SVM Kernel Sigmoide')
plt.show()
```

[32] ✓ 0.0s

Matriz de confusión - SVM Kernel Sigmoide

File Edit Selection View Go Run ···

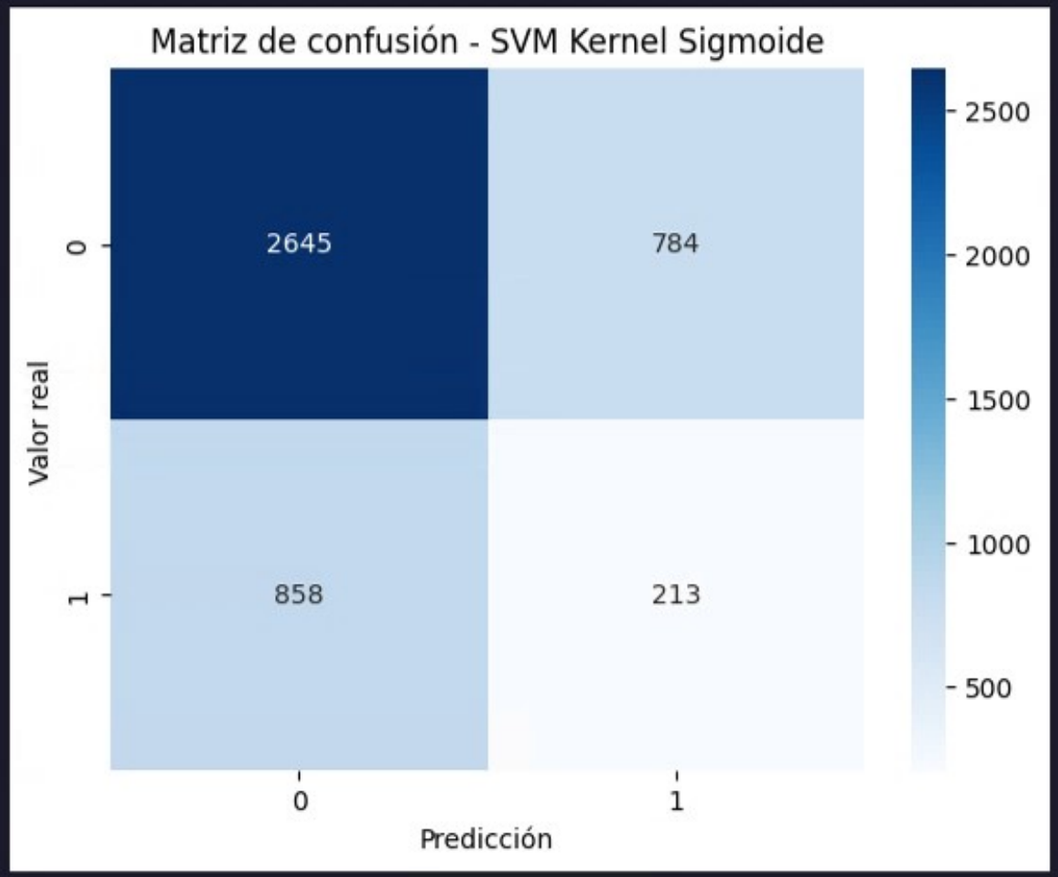
Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

[32] ✓ 0.0s




```
print(classification_report(y_test, y_pred_sigmoid))
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Reporte de clasificación
print(classification_report(y_test, y_pred_sigmoid))
```

[33] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.76      0.77      0.76      3429
           1       0.21      0.20      0.21      1071

    accuracy                           0.64      4500
   macro avg       0.48      0.49      0.48      4500
weighted avg       0.63      0.64      0.63      4500
```

El modelo SVM con kernel sigmoide completa el análisis comparativo entre los distintos enfoques evaluados. A partir de sus métricas de desempeño se puede obs
este kernel logra una clasificación adecuada o si presenta limitaciones frente a los modelos previamente analizados.

Los resultados obtenidos se utilizarán para seleccionar el modelo más apropiado considerando precisión, capacidad de generalización e interpretación de errores.

Comparación de modelos SVM

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Comparación de modelos SVM

Con el objetivo de seleccionar el modelo predictivo más adecuado, se evaluaron cuatro configuraciones de Máquinas de Soporte Vectorial, cada una utilizando un kernel distinto: lineal, polinomial, radial (RBF) y sigmoide.

La comparación se realizó considerando las métricas de clasificación obtenidas sobre el conjunto de prueba, tales como precisión (*accuracy*), *recall*, *f1-score* y la distribución de errores reflejada en las matrices de confusión. Este enfoque permite no solo evaluar el desempeño global, sino también analizar la capacidad de ca modelo para identificar correctamente a los empleados que abandonan la empresa.

Selección del modelo predictivo más adecuado

A partir del análisis comparativo de los cuatro modelos evaluados, el modelo SVM con kernel RBF presentó el mejor desempeño predictivo, alcanzando una precisi superior al resto de los enfoques analizados.

Este kernel demostró una mayor capacidad para capturar relaciones no lineales presentes en los datos, lo que se reflejó tanto en la accuracy obtenida como en la distribución de errores observada en su matriz de confusión. En particular, el modelo RBF logró identificar de forma más consistente a los empleados que abandon empresa, reduciendo errores críticos para la toma de decisiones en el área de Recursos Humanos.

Pronóstico para un empleado con características específicas

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Pronóstico para un empleado con características específicas

Una vez seleccionado el modelo SVM con mejor desempeño, se procede a utilizarlo para realizar un pronóstico individual. Este análisis permite estimar la probabili abandono de un empleado a partir de sus indicadores laborales y constituye una herramienta de apoyo para la toma de decisiones en el área de Recursos Humanc

```python
# Tomar un registro real como plantilla
empleado = X_test.iloc[[0]].copy()

# Modificar valores numéricos para simular un empleado específico
empleado['satisfaction_level'] = 0.45
empleado['last_evaluation'] = 0.60
empleado['average_montly_hours'] = 160
empleado['time_spend_company'] = 3

# Asegurar variables binarias
if 'work_accident' in empleado.columns:
    empleado['work_accident'] = 0

if 'promotion_last_5years' in empleado.columns:
    empleado['promotion_last_5years'] = 0

# Escalar
empleado_scaled = scaler.transform(empleado)

# Predicción con SVM RBF
prediccion = svm_rbf.predict(empleado_scaled)
```

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
# Predicción con SVM RBF
prediccion = svm_rbf.predict(empleado_scaled)

prediccion
```

[34] ✓ 0.0s

```
··· array([0])
```

El pronóstico obtenido indica que el empleado analizado no presenta una probabilidad significativa de abandonar la empresa. De acuerdo con el modelo SVM seleccionado, su nivel de satisfacción, desempeño y condiciones laborales actuales no lo ubican dentro de un perfil de riesgo.

Este resultado permite concluir que, bajo las condiciones evaluadas, no sería necesario implementar acciones correctivas inmediatas, aunque se recomienda mante monitoreo continuo de sus indicadores laborales.

Generate + Code + Markdown

Conclusiones finales

En esta práctica se desarrolló un modelo de clasificación utilizando Máquinas de Soporte Vectorial aplicado a una base de datos de Recursos Humanos, con el obje analizar los factores asociados al abandono de empleados dentro de una organización.

Se evaluaron cuatro configuraciones distintas de SVM mediante el uso de kernels lineal, polinomial, radial (RBF) y sigmoide. A partir de métricas cuantitativas, matri confusión y reportes de clasificación, se determinó que el modelo con kernel RBF presentó el mejor desempeño predictivo, logrando una mayor precisión y una me identificación de empleados en riesgo de abandono.

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

[34] ✓ 0.0s

```
array([0])
```

El pronóstico obtenido indica que el empleado analizado no presenta una probabilidad significativa de abandonar la empresa. De acuerdo con el modelo SVM seleccionado, su nivel de satisfacción, desempeño y condiciones laborales actuales no lo ubican dentro de un perfil de riesgo.

Este resultado permite concluir que, bajo las condiciones evaluadas, no sería necesario implementar acciones correctivas inmediatas, aunque se recomienda mante monitoreo continuo de sus indicadores laborales.

Conclusiones finales

En esta práctica se desarrolló un modelo de clasificación utilizando Máquinas de Soporte Vectorial aplicado a una base de datos de Recursos Humanos, con el obje analizar los factores asociados al abandono de empleados dentro de una organización.

Se evaluaron cuatro configuraciones distintas de SVM mediante el uso de kernels lineal, polinomial, radial (RBF) y sigmoide. A partir de métricas cuantitativas, matri confusión y reportes de clasificación, se determinó que el modelo con kernel RBF presentó el mejor desempeño predictivo, logrando una mayor precisión y una me identificación de empleados en riesgo de abandono.

Finalmente, el modelo seleccionado fue utilizado para generar un pronóstico individual, demostrando su aplicabilidad práctica como herramienta de apoyo a la tor decisiones estratégicas en el área de Recursos Humanos. Los resultados obtenidos evidencian el valor del uso de modelos de Machine Learning para la prevención rotación de personal y la gestión basada en datos.

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM)

Autor: RobertScience
Curso: Profesión Científico de Datos v2
Práctica: M22 – Clasificación con SVM

Objetivo del proyecto

El objetivo de esta práctica es desarrollar un modelo predictivo de clasificación utilizando Máquinas de Soporte Vectorial (SVM) aplicado a una base de datos de re
humanos, con el fin de analizar los factores que influyen en la decisión de los empleados de abandonar la empresa.

A través de la evaluación de distintos tipos de *kernel*, se busca identificar el modelo con mejor desempeño predictivo y analizar sus resultados mediante métricas d
clasificación, matrices de confusión y una interpretación orientada a la toma de decisiones en el área de Recursos Humanos.

Preparación del entorno de trabajo

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

Preparación del entorno de trabajo

Antes de iniciar el análisis de los datos y la construcción de los modelos de clasificación, se verifica que el entorno de ejecución esté correctamente configurado. Es validación permite asegurar que el análisis se ejecute dentro del entorno virtual del proyecto y que las librerías necesarias se encuentren disponibles.

La correcta configuración del entorno es un paso fundamental para garantizar la reproducibilidad de los resultados obtenidos a lo largo del proyecto.

```python
import sys
import pandas as pd
import numpy as np
import sklearn

print("Python:", sys.version)
print("Executable:", sys.executable)
print("pandas:", pd.__version__)
print("numpy:", np.__version__)
print("sklearn:", sklearn.__version__)
print("ENTORNO OK")
```

[1] ✓ 11.0s

```
Python: 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)]
Executable: c:\Users\ROBERTSCIENCE\Documents\Tarea M22-CD –RobertScience\.venv\Scripts\python.exe
pandas: 3.0.0
```

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
Python: 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)]
Executable: c:\Users\ROBERTSCIENCE\Documents\Tarea M22-CD –RobertScience\.venv\Scripts\python.exe
pandas: 3.0.0
numpy: 2.4.2
sklearn: 1.8.0
ENTORNO OK
```

La verificación del entorno confirma que el proyecto se está ejecutando dentro del entorno virtual correspondiente y que las principales librerías necesarias para el
y el modelado predictivo se encuentran correctamente instaladas. Con esto, se puede avanzar de forma segura a las siguientes etapas del análisis.

Carga del conjunto de datos

En esta etapa se carga la base de datos `recursos_humanos.csv`, la cual contiene la información histórica de empleados utilizada para el análisis de rotación de p
La carga del conjunto de datos permite iniciar el proceso de exploración y comprender la estructura general de la información disponible.

A partir de este punto se trabajará con los datos originales, realizando posteriormente las transformaciones necesarias para su uso en modelos de clasificación.

```
# Carga del dataset de recursos humanos
df = pd.read_csv("../data/recursos_humanos.csv")

# Visualización de las primeras observaciones
df.head()
```

[2] ✓ 0.0s

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
df.head()
```

[2] ✓ 0.0s

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

La visualización de las primeras observaciones permite confirmar que la base de datos fue cargada correctamente y que las variables corresponden a la informació esperada. Esta revisión inicial facilita la comprensión del tipo de datos con los que se trabajará en las siguientes etapas del análisis.

Inspección general del conjunto de datos

Una vez cargada la información, se realiza una inspección general del conjunto de datos con el objetivo de conocer su dimensión, el tipo de variables presentes y c estadísticas descriptivas básicas. Este análisis preliminar permite identificar posibles consideraciones a tener en cuenta antes de la preparación de los datos.

```
# Dimensiones del conjunto de datos
df.shape
```

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```python
# Dimensiones del conjunto de datos
df.shape
```

[3] ✓ 0.0s

```
(14999, 10)
```

Generate + Code + Markdown

```python
# Información general del dataset
df.info()
```

[4] ✓ 0.0s

```
<class 'pandas.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   satisfaction_level     14999 non-null  float64
 1   last_evaluation        14999 non-null  float64
 2   number_project         14999 non-null  int64
 3   average_montly_hours   14999 non-null  int64
 4   time_spend_company     14999 non-null  int64
 5   Work_accident          14999 non-null  int64
 6   left                   14999 non-null  int64
 7   promotion_last_5years  14999 non-null  int64
 8   sales                  14999 non-null  str
 9   salary                 14999 non-null  str
```

File Edit Selection View Go Run ··· Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

```
 7   promotion_last_5years  14999 non-null  int64
 8   sales                  14999 non-null  str
 9   salary                 14999 non-null  str
dtypes: float64(2), int64(6), str(2)
memory usage: 1.1 MB
```

```python
# Estadísticas descriptivas de las variables
df.describe()
```

[5] ✓ 0.0s

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

A partir de la inspección general se observa que el conjunto de datos no presenta valores faltantes y que contiene una combinación de variables numéricas y categ
Las estadísticas descriptivas permiten identificar rangos y comportamientos generales de las variables, lo cual resulta relevante para las siguientes etapas de prepar

File Edit Selection View Go Run ···

Tarea M22-CD –RobertScience [Administrator]

Tarea_M22-CD_robertscience.ipynb ● | Tarea_M22-CD_robertscience.ipynb

notebooks > Tarea_M22-CD_robertscience.ipynb > M↓ Tarea M22-CD – Análisis de Clasificación con Máquinas de Soporte Vectorial (SVM) > M↓ Conclusiones finales

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· .venv (Pyth

A partir de la inspección general se observa que el conjunto de datos no presenta valores faltantes y que contiene una combinación de variables numéricas y categ
Las estadísticas descriptivas permiten identificar rangos y comportamientos generales de las variables, lo cual resulta relevante para las siguientes etapas de prepar
modelado de los datos.

Análisis de la variable objetivo

La variable objetivo del análisis es `left`, la cual indica si un empleado abandonó la empresa (1) o permaneció en ella (0). El análisis de su distribución permite iden
el conjunto de datos se encuentra equilibrado o desbalanceado entre las clases, lo cual es un aspecto relevante para el desempeño de los modelos de clasificación

```
# Distribución de la variable objetivo
df['left'].value_counts()
```

[6] ✓ 0.0s

```
left
0    11428
1     3571
Name: count, dtype: int64
```

```
# Distribución porcentual de la variable objetivo
df['left'].value_counts(normalize=True) * 100
```