Tecnicatura en Ciencia de Datos e Inteligencia Artificial
Modulo Analista de Datos
Evidencia de Aprendizaje N°5

Análisis de Satisfacción y Retención Laboral en BioTech Innovations S.A.

DOCENTES
Prata Nahuel
Ugarte Marcos

GRUPO DATAMINDS
1. Ayán, María Trinidad
2. Giordano, Ariel Eduardo
3. Herrera, Edgar Fabián
4. Quiroga, Fernanda

EMPRESA

BioTech Innovations S.A.

COLAB

DataMinds_Analista_De_Datos_Proyecto.ipynb [1]

[1] El presente Colab es una copia de la versión anterior. El cambio se debe a que la propietaria del archivo es una alumna que ha dejado la Tecnicatura Superior en Ciencia de Datos e Inteligencia Artificial. El cambio permitió estructurar de mejor manera el trabajo mediante el índice, títulos y subtítulos.

Índice

Introducción..... 2
Propuesta de Modelo de Regresión Lineal Simple..... 2
Procedimiento..... 2
Código..... 3
Predicciones..... 3
Código..... 4
Prueba del Modelo..... 5
Código..... 5
Medidas de Regresión..... 5
Código..... 6
Matriz de Correlación..... 6
Código..... 6
Conclusiones..... 6
Conclusiones de las Medidas de Regresión..... 7
Conclusiones sobre la Matriz de Correlación..... 7

Introducción

En el presente trabajo se propone la realización de un modelo de regresión lineal simple. Este modelo constituye una reducción de una variable (dependiente) basada en otra (independiente). Se ha utilizado la biblioteca Scikit-Learn para construir el modelo de regresión lineal. Además, se incluyen las métricas de rendimiento: error cuadrático medio (RMSE), error absoluto medio (MAE) y el valor de R-cuadrado (R^2) para evaluar el modelo. Por último, se incluye una matriz de correlación de hasta 10 variables, que representaremos visualmente sus relaciones usando un mapa de calor o gráficos de pares.

Propuesta de Modelo de Regresión Lineal Simple

El modelo de regresión lineal simple busca predecir una variable dependiente y a partir de una sola variable independiente. Se seleccionan dos variables del dataset:

- **Variable independiente ("x" predictora):** una variable que tiene un impacto sobre la variable dependiente (edad, género, antigüedad laboral, porcentaje de aumento del salario).
- **Variable dependiente ("y" respuesta):** la variable que queremos predecir

Procedimiento

1. **Definir Variables:**

    - **Variable predictora ("x" independiente):** edad, género, antigüedad laboral, porcentaje de aumento del salario
    - **Variable respuesta ("y" dependiente):** satisfacción laboral general.

2. **División del dataset:** dividimos los datos en un conjunto de entrenamiento y un conjunto de prueba. Esto ayuda a entrenar el modelo y luego probar su rendimiento con datos que no ha visto antes, en una proporción 80/20.

3. **Entrenamiento del modelo:** usamos un modelo de regresión lineal simple de scikit-learn para ajustar la relación entre la variable independiente y dependiente.

4. **Evaluación del modelo:** calculamos el RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) y el R^2 (Coeficiente de Determinación) para medir la precisión del modelo.

Código

```
#Importar la librería
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

#Carga del dataset
df = pd.read_csv('ruta_al_archivo.csv')

#Definir variables independientes (x) y variables independientes (y)
X = df[['Edad', 'Género', 'Antigüedad laboral', 'Porcentaje Aumento Del Salario']]  #Variable independiente
y = df['Satifacción laboral general']  #Variable dependiente

#Antes de dividir los datos, eliminar filas con valores faltantes
df = df.dropna(subset=['Edad', 'Género', 'Antigüedad laboral', 'Porcentaje Aumento Del Salario'])

#Dividir los datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

#Crear un imputador para rellenar los valores NaN con la media
imputer = SimpleImputer(strategy='mean') # o strategy='median'

#Ajustar el imputador a los datos de entrenamiento y transformar tanto los datos de entrenamiento como los de prueba.
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)

#Crear y entrenar el modelo de regresión lineal
model = LinearRegression() # Creamos
model.fit(X_train, y_train) # Entrenamos el modelo con los datos de entrenamiento

#Hacer predicciones sobre los datos de prueba
y_pred = model.predict(X_test)
```

Predicciones

El modelo de regresión lineal busca encontrar una relación, por ejemplo, entre los años de antigüedad y el salario, es decir, cómo el salario cambia conforme aumentan los años de experiencia.

Código

```
#Importar la librería
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

#Carga del dataset
df = pd.read_csv('ruta_al_archivo.csv')

#Explorar el dataset
print(df.head())  # Ver las primeras filas del dataset
print(df.describe())  # Descripción estadística del dataset

#Seleccionar las variables independiente (X) y dependiente (y)
X = df[['Antigüedad']] # variable predictora
y = df['Salario']  # variable respuesta
# Dividi los datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

#Crear el modelo de regresión lineal
modelo = LinearRegression()

#Entrenar el modelo con los datos de entrenamiento
modelo.fit(X_train, y_train)

# Predecir con los datos de prueba
y_pred = modelo.predict(X_test)

#Evaluar el modelo
rmse = mean_squared_error(y_test, y_pred, squared=False)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'RMSE: {rmse}')
print(f'MAE: {mae}')
print(f'R²: {r2}')

#Visualizar el modelo
plt.scatter(X_test, y_test, color='blue', label='Datos Reales')
plt.plot(X_test, y_pred, color='red', label='Predicción')
plt.xlabel('Años de Experiencia')  # variable X
plt.ylabel('Salario')  #variable Y
plt.title('Regresión Lineal Simple')
plt.legend()
plt.show()

#Matriz de correlación
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```

Prueba del Modelo

Para probar el modelo se necesitan una serie de pasos:

1. **Entrenar el modelo:** el modelo de regresión lineal simple se entrenó utilizando una variable predictora (independiente) y una variable de respuesta (dependiente). Esto lo hicimos anteriormente.
2. **Probar el modelo:** usaremos los datos de prueba para predecir la variable dependiente y evaluaremos la precisión del modelo con métricas de evaluación.

El modelo intenta establecer una relación lineal entre estas dos variables para predecir el comportamiento de la variable dependiente a partir de la independiente.

Código

```
#Predecir los resultados usando los datos de prueba. Usamos el modelo entrenado para predecir los valores de la variable dependiente usando los datos de test.

y_pred_test = modelo.predict(X_test)

#Evaluación del modelo con los datos de test
rmse_test = mean_squared_error(y_test, y_pred_test, squared=False)  # RMSE
mae_test = mean_absolute_error(y_test, y_pred_test)  # MAE
r2_test = r2_score(y_test, y_pred_test)  # R²

#Mostrar los resultados de la evaluación
print("Resultados del modelo con los datos de test:")
print(f'RMSE (Test): {rmse_test}')
print(f'MAE (Test): {mae_test}')
print(f'R² (Test): {r2_test}')

#Visualizar la predicción versus los valores reales. Generar una gráfica que compare los valores reales con las predicciones realizadas por el modelo.

import matplotlib.pyplot as plt
plt.scatter(X_test, y_test, color='blue', label='Valores Reales')
plt.plot(X_test, y_pred_test, color='red', label='Predicciones del Modelo')
plt.xlabel('Años de experiencia (X)')
plt.ylabel('Salario (Y)')
plt.title('Prueba del Modelo: Datos Reales vs Predicciones')
plt.legend()
plt.show()
```

Medidas de Regresión

- **RMSE (Root Mean Squared Error)**: es una medida de error que calcula la raíz cuadrada del error medio cuadrático. Un valor **bajo** indica que el modelo tiene un **buen** rendimiento.
- **MAE (Mean Absolute Error):** mide la diferencia promedio entre los valores predichos y los valores reales en términos absolutos. Un valor **bajo** significa un **buen** ajuste.
- **R² (Coeficiente de determinación):** indica qué tan bien el modelo puede explicar la variabilidad en los datos. Un valor de R² **cercano a 1** indica que el modelo se ajusta bien a los datos.

Código

```
#Predecir con los datos de prueba
y_pred_test = modelo.predict(X_test)

#Calcular RMSE (Root Mean Squared Error)
rmse_test = mean_squared_error(y_test, y_pred_test, squared=False)  #Error cuadrático medio raíz

# Calcular MAE (Mean Absolute Error)
mae_test = mean_absolute_error(y_test, y_pred_test)  #Error absoluto medio

#Calcular R² (Coeficiente de determinación)
r2_test = r2_score(y_test, y_pred_test)  #Coeficiente de determinación

#Mostrar los resultados de las métricas
print("Resultados de las métricas de evaluación con los datos de test:")
print(f'RMSE (Test): {rmse_test}')
print(f'MAE (Test): {mae_test}')
print(f'R² (Test): {r2_test}')
```

Matriz de Correlación

Para analizar la relación entre diferentes variables, generamos y visualizamos una matriz de correlación de un subconjunto de variables.

Código

```
#Seleccionar un subconjunto de variables
subset_df = df[['Satifacción laboral general', 'Antigüedad laboral', 'Porcentaje Aumento Del Salario', 'Edad', 'Género', 'Satisfacción de relaciones', 'Cantidad de horas trabajadas']]

#Matriz de correlación
plt.figure(figsize=(10, 6))
sns.heatmap(subset_df.corr(), annot=True, cmap='coolwarm')
plt.title('Matriz de Correlación')
plt.show()
```

Conclusiones

El presente trabajo tiene el objetivo de predecir la Satisfacción Laboral General de los empleados en función de alguna variable independiente, como Género o Porcentaje Aumento del Salario. Sin embargo, las pruebas realizadas se apartan de toda predicción. También se ha agregado sin éxito la variable Salario Total, a partir de la adecuación de la escala salarial de la Federación Argentina de Trabajadores de Industrias Químicas y Petroquímicas al dataset utilizado para el desarrollo del módulo.

Conclusiones de las Medidas de Regresión

- El Error cuadrático medio (RMSE) y Error absoluto medio (MAE) son altos. Esto indica que el modelo no se ajusta bien a los datos.
- El Coeficiente de determinación (R^2) es bajo, puede que necesitemos incluir más variables o transformar las existentes. Si el R^2 hubiese estado cercano a 1 el modelo explicaría bien la variabilidad en la satisfacción laboral general por ejemplo.
- Los resultados no son satisfactorios, por tal motivo se debe explorar con otras variables que puedan influir en la satisfacción laboral.

Conclusiones sobre la Matriz de Correlación

- Se pueden identificar variables que están altamente correlacionadas con la satisfacción laboral general, lo cual puede ayudar a enriquecer el modelo. Por ejemplo, un alto porcentaje de aumento del salario podría estar relacionado con una mayor satisfacción.
- La antigüedad laboral y la cantidad de horas trabajadas también pueden tener relaciones significativas, lo que podría ser útil para futuras investigaciones.
- Variables como Porcentaje Aumento Del Salario, Cantidad de horas trabajadas y Nivel de Educación podrían tener una fuerte asociación, lo que indica que son factores relevantes a considerar en el análisis de satisfacción y retención laboral.
- Hemos probado que el análisis de regresión lineal simple con una única variable independiente puede influir en la predicción de una variable dependiente. Este análisis ha sido útil para demostrar la importancia de una sola variable en la predicción de otra, proporcionando un punto de partida para un análisis más exhaustivo.