UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Maestría en Ciencia de Datos

Aprendizaje Automático

Alvaro Pequeño Mondragón
1726520

Reporte #2

21 de julio de 2024

Para la practica 2 se requiere realizar un modelo de regresión y encontrar los mejores parámetros para el modelo usando la validación cruzada. Esto se hará usando gridsearch que esta disponible en la librería de sklearn.

Se realizó un modelo de regresión para poder determinar el pago diario promedio de las dependencias tomando el número de empleados que estas tienen. El modelo en especifico es de knn vecinos regresor.

Primero se elaboró el modelo sin validación cruzada:

```
#Se separan los datos en X y Y
model1_x = model1.drop(['num_empleados'], axis = 1, inplace = False)
model1_y = model1.drop(['pago_diario_promedio'], axis = 1, inplace = False)
```
✓ 0.0s

```
#Se entrena el primer modelo
model1_xtrain, model1_xtest, model1_ytrain, model1_ytest = train_test_split(model1_x, model1_y, test_size=0.2, random_state=73)
model1_knn_regressor = KNeighborsRegressor(n_neighbors=5)
model1_knn_regressor.fit(model1_xtrain, model1_ytrain)
```
✓ 0.0s

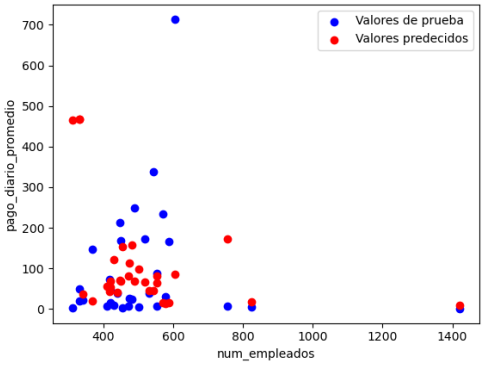
```
KNeighborsRegressor
KNeighborsRegressor()
```

+ Code + Markdown

```
#Se hace una la predicción de los datos de prueba y se obtiene la evaluación del modelo
model1_ypred = model1_knn_regressor.predict(model1_xtest)
model1_mse = mean_squared_error(model1_ytest, model1_ypred)
model1_r2 = r2_score(model1_ytest, model1_ypred)
print('MSE: '+ str(model1_mse))
print('R2 Score: ' + str(model1_r2))
```
✓ 0.0s

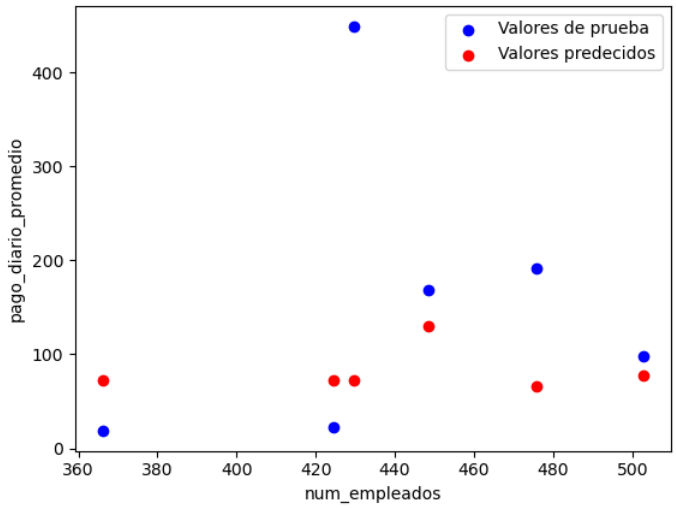
```
MSE: 43367.32387096775
R2 Score: -1.093764766653937
```

A continuación, se muestra gráficamente los resultados del modelo anterior:



Como se puede observar la grafica está muy saturada de puntos por lo que se decidió hacer un nuevo modelo tomando en cuento solamente los datos de las preparatorias para ver mas claro algunos puntos y ver que tan acertadas son las predicciones:

La grafica obtenida es la siguiente:



Por ultimo se utilizan los datos del ultimo modelo para hacer la validación cruzada y poder obtener los parámetros que obtengan un mejor resultado en cuanto al error medio cuadrado. Se eligió esta métrica ya que esta penaliza los errores más grandes por lo que me permite ver que tan alejadas están saliendo las predicciones de los valores verdaderos:

```
#Se usarán los datos anteriores para encontrar el mejor modelo usando la validación cruzada, primero se4 hace el modelo y los parametris a probar
knn_vc = KNeighborsRegressor()
knn_vc_para = [{'n_neighbors': [1,2,4,6,8], 'weights':['uniform', 'distance']}]
```

0.0s

```
#Se define el modelo para la validación cruzada y se obtiene el mejor modelo
best_knn_vc = GridSearchCV(knn_vc, knn_vc_para, cv=5, scoring='neg_mean_squared_error')

best_knn_vc.fit(model2_xtrain, model2_ytrain)
print(best_knn_vc.best_params_)
print(best_knn_vc.best_score_)
```

0.4s

```
{'n_neighbors': 6, 'weights': 'uniform'}
-5538.8741666666665
```

Como se puede observar en la imagen anterior el mejor modelo para la métrica del error medio cuadrado el mejor modelo de knn medias regresor es aquel con los siguientes parámetros:

- n_neighbors = 6
- weights = uniform

Los parámetros anteriores nos dan un resultado de MSE = -5538.8741