

artPredict

October 24, 2022

```
[1]: import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
import math
```

0.0.1 Lector de dades

```
[2]: art = pd.read_csv("./data.csv", header=0, delimiter=',')
art.head()
```

```
[2]:
```

	TIPOLOGÍA	DENOMINACIÓN O TÍTULO	AUTOR	ESTILO \
0	PINTURA	PAISAJE ABSTRACTO	ANTONIO MARTÍNEZ MORALES	CONTEMPORÁNEO
1	PINTURA	ABSTRACCIÓN GEOMÉTRICA	J. CRUZ ILLÁN CHUS	CONTEMPORÁNEO
2	PINTURA	PAISAJE RURAL	JOSÉ LUCAS RUIZ	CONTEMPORÁNEO
3	PINTURA	PINTURA ABSTRACTA	MARÍA JOSÉ RAMÍREZ	CONTEMPORÁNEO
4	PINTURA	NIÑA EN EL BALCÓN	ANTONIO GÓMEZ CANO	CONTEMPORÁNEO

	CRONOLOGÍA	MATERIA/TÉCNICA	DIMENSIONES	ESTADO DE CONSERVACIÓN	PUNTUACIÓN
0	SIGLO XX	MIXTA SOBRE TABLA	48 X 38cm	BUENO/ENMARCADO	8.2
1	2007	ÓLEO SOBRE LIENZO	73 X 91cm	BUENO/ENMARCADO	7.9
2	SIGLO XX	ÓLEO SOBRE LIENZO	25 X 33cm	BUENO/ENMARCADO	8.9
3	SIGLO XX	ÓLEO SOBRE LIENZO	25 X 19cm	BUENO/ENMARCADO	8.7
4	1948	ÓLEO SOBRE LIENZO	59 X 49cm	BUENO/ENMARCADO	8.5

0.0.2 Modificar valores variables

```
[3]: def renameType(col):
    newCol = []
    for c in col:
        aux = c.split()
        if 'PAISAJE' in aux:
            newCol.append('PAISAJE')
        elif 'ABSTRACTA' in aux:
            newCol.append('ABSTRACTA')
        elif 'RETRATO' in aux:
```

```

        newCol.append('RETRATO')
    else:
        newCol.append('OTRO')
return newCol

```

```

[4]: def calculateDimensions(col):
    newCol = []
    for c in col:
        if '+' not in c:
            if 'x' in c:
                aux = c[:-2].split(' x ')
            else:
                aux = c[:-2].split(' X ')
            val = 1
            for v in aux:
                if '.' in v:
                    v = ''.join([w for w in v if w != '.'])
                    val *= int(v)
            else:
                aux = c[:-2].split(' + ')
                v1 = ''.join([w for w in aux[0] if w != '.'])
                v2 = ''.join([w for w in aux[1] if w != '.'])
                val = int(v1) + int(v2)
        newCol.append(val)
    return newCol

```

```

[5]: def modifyCronology(col):
    newCol = []
    for c in col:
        aux = c.split()
        if 'SIGLO' in c:
            newCol.append(c)
        elif 'DESCONOCIDA' in c or 'CIRCA' in c:
            newCol.append('SIGLO XX')
        else:
            if int(c) >= 2000:
                newCol.append('SIGLO XXI')
            elif int(c) >= 1900:
                newCol.append('SIGLO XX')
            elif int(c) >= 1800:
                newCol.append('SIGLO XIX')
    return newCol

```

```

[6]: def groupMateria(col,group):
    newCol = []
    for c in col:

```

```

    if c in group:
        newCol.append('OTRO')
    else:
        newCol.append(c)
return newCol

```

```

[7]: tipo = renameType(art['DENOMINACIÓN O TÍTULO'])
dimension = calculateDimensions(art['DIMENSIONES'])
crono = modifyCronology(art['CRONOLOGÍA'])
counts = art['MATERIA/TÉCNICA'].value_counts()
counts = counts.index[counts < 10]
print(counts)
mat = groupMateria(art['MATERIA/TÉCNICA'], counts)

art.drop(columns=['DENOMINACIÓN O TÍTULO',
↳ 'DIMENSIONES', 'CRONOLOGÍA', 'AUTOR'], inplace = True)

art['TIPO'] = tipo
art['DIMENSIONES'] = dimension
art['CRONOLOGÍA'] = crono
art['MATERIA/TÉCNICA'] = mat

art.head()

```

```

Index(['PIEDRA CALIZA TALLADA Y PULIDA', 'BRONCE A LA CERA PERDIDA',
      'GRABADO SOBRE PAPEL', 'ACRÍLICO SOBRE LIENZO', 'COLLAGE SOBRE PAPEL',
      'PIGMENTOS NATURALES, LÁTEX Y CELULOSA', 'ACRÍLICO SOBRE TABLA',
      'PASTEL SOBRE PAPEL', 'PIGMENTOS NATURALES Y RESINA DE POLIÉSTER',
      'PAPEL SOBRE PAPEL', 'CARBONCILLO SOBRE PAPEL', 'SANGUINA SOBRE PAPEL',
      'ACUARELA SOBRE PAPEL ARCHES', 'MIXTA SOBRE PAPEL',
      'HIBRIDACIONES-IMPRESIÓN DIGITAL', 'SERIGRAFIA SOBRE PAPEL',
      'PIEDRA ARENÍSCA NEGRA', 'PIEDRA, GRANITO Y MÁRMOL', 'ACERO CORTÉN',
      'ACERO CORTÉN Y BRONCE', 'CERÁMICA VIDRIADA', 'TINTA SOBRE PAPEL',
      'ACERO CORTÉN Y PIEL', 'MIXTA SOBRE PIEDRA',
      'CERAS SOBRE PAPEL DE LIJA', 'AGUA FUERTE SOBRE RESINA Y PIGMENTOS',
      'CERAS SOBRE PAPEL',
      'PIGMENTOS NATURALES DE ALTA COCCIÓN SOBRE CERÁMICA GRES',
      'PINTURA SOBRE PAPEL', 'ACERO CORTÉN, BRONCE Y METAL',
      'METAL Y PINTURA', 'MIXTA SOBRE FUSTA', 'DIBUJO Y PINTURA SOBRE PAPEL',
      'MIXTA ACUARELA', 'ARCILLA VITRIFICADA',
      'ÓLEO, TINTES Y PIGMENTOS EN POLVO SOBRE LIENZO', 'AGUADA SOBRE PAPEL',
      'VACIADO DE CEMENTO', 'ÓLEO SOBRE TELA', 'ARCILLA TALLADA Y PULIDA',
      'MADERA TALLADA Y PULIDA', 'PÁTINA DE PIGMENTOS NATURALES',
      'OLIVERA TALLADA Y PULIDA', 'PINTURA SOBRE TABLA',
      'ACRÍLICO SOBRE TELA', 'MIXTA SOBRE MADERA',
      'CARBONCILLO Y ACUARELA SOBRE PAPEL', 'YESO MATE PLATEADO SOBRE TABLA',

```

```
'MIXTA SOBRA TABLA', 'ARCILLA', 'ARCILLA INCISA Y PINTADA'],
dtype='object')
```

```
[7]: TIPOLOGÍA      ESTILO      MATERIA/TÉCNICA ESTADO DE CONSERVACIÓN \
0  PINTURA  CONTEMPORÁNEO  MIXTA SOBRA TABLA      BUENO/ENMARCADO
1  PINTURA  CONTEMPORÁNEO  ÓLEO SOBRE LIENZO      BUENO/ENMARCADO
2  PINTURA  CONTEMPORÁNEO  ÓLEO SOBRE LIENZO      BUENO/ENMARCADO
3  PINTURA  CONTEMPORÁNEO  ÓLEO SOBRE LIENZO      BUENO/ENMARCADO
4  PINTURA  CONTEMPORÁNEO  ÓLEO SOBRE LIENZO      BUENO/ENMARCADO

      PUNTUACIÓN      TIPO  DIMENSIONES CRONOLOGÍA
0          8.2      PAISAJE      1824  SIGLO XX
1          7.9        OTRO      6643  SIGLO XXI
2          8.9      PAISAJE      825  SIGLO XX
3          8.7  ABSTRACTA      475  SIGLO XX
4          8.5        OTRO      2891  SIGLO XX
```

```
[8]: counts = art['CRONOLOGÍA'].value_counts()
art = art.loc[art['CRONOLOGÍA'].isin(counts.index[counts > 100])]

counts = art['ESTILO'].value_counts()
art = art.loc[art['ESTILO'].isin(counts.index[counts > 4])]

counts = art['ESTADO DE CONSERVACIÓN'].value_counts()
art = art.loc[art['ESTADO DE CONSERVACIÓN'].isin(counts.index[counts > 1])]
```

0.0.3 Codificar Variables

```
[9]: def preprocessing(X, y=None, scaler=None):
    categorical_columns = ['TIPOLOGÍA', 'TIPO', 'ESTILO', 'CRONOLOGÍA', 'MATERIA/
    ↳TÉCNICA', 'ESTADO DE CONSERVACIÓN']
    numerical_columns = [c for c in X.columns if (c not in categorical_columns)]

    # Escalamos las variables numericas
    if scaler is None:
        # Generamos el scaler cuando los datos son los de entrenamiento
        scaler = StandardScaler()
        X[numerical_columns] = scaler.fit_transform(X[numerical_columns])
    else:
        X[numerical_columns] = scaler.transform(X[numerical_columns])

    # Aplicamos un one-hot encoding a las variables categoricas
    for column in categorical_columns:
        X_one_hot = pd.get_dummies(X[column], prefix=column)
        X = X.merge(X_one_hot, left_index=True, right_index=True)
        X = X.drop(columns=[column])
    return X, y, scaler
```

```
[10]: X = art.loc[:,art.columns != 'PUNTUACIÓN']
y = art['PUNTUACIÓN']

X_train, test, y_train, y_test = train_test_split(X, y, test_size=0.30,
↳random_state=33)
X_train, y_train, scaler = preprocessing(X_train, y_train)
X_test, y_test, _ = preprocessing(test, y_test, scaler)

X_train.head()
```

```
[10]:
```

	DIMENSIONES	TIPOLOGÍA_DIBUJO	TIPOLOGÍA_ESCULTURA	TIPOLOGÍA_FOTOGRAFÍA	\
128	-0.100018	0	0	0	
81	-0.096945	0	0	0	
312	-0.100172	0	0	0	
147	-0.100100	0	0	0	
273	-0.097474	0	0	0	

	TIPOLOGÍA_PINTURA	TIPO_ABSTRACTA	TIPO_OTRO	TIPO_PAISAJE	TIPO_RETRATO	\
128	1	0	1	0	0	
81	1	1	0	0	0	
312	1	0	0	0	1	
147	1	0	1	0	0	
273	1	0	1	0	0	

	ESTILO_CONTEMPORÁNEO	...	MATERIA/TÉCNICA_OTRO	\
128	1	...	0	
81	1	...	0	
312	1	...	0	
147	1	...	1	
273	1	...	1	

	MATERIA/TÉCNICA_PUNTA SECA SOBRE PAPEL	\
128	0	
81	0	
312	0	
147	0	
273	0	

	MATERIA/TÉCNICA_ÓLEO SOBRE LIENZO	MATERIA/TÉCNICA_ÓLEO SOBRE TABLA	\
128	0	0	
81	0	0	
312	1	0	
147	0	0	
273	0	0	

	ESTADO DE CONSERVACIÓN_BUENO/ENMARCADO	\
128	1	

81	1
312	0
147	0
273	0

	ESTADO DE CONSERVACIÓN_BUENO/SIN ENMARCAR \
128	0
81	0
312	0
147	1
273	0

	ESTADO DE CONSERVACIÓN_BUENO/SIN RESTAURAR \
128	0
81	0
312	0
147	0
273	1

	ESTADO DE CONSERVACIÓN_REGULAR/ENMARCADO \
128	0
81	0
312	1
147	0
273	0

	ESTADO DE CONSERVACIÓN_REGULAR/SIN ENMARCAR \
128	0
81	0
312	0
147	0
273	0

	ESTADO DE CONSERVACIÓN_REGULAR/SIN RESTAURAR
128	0
81	0
312	0
147	0
273	0

[5 rows x 27 columns]

0.0.4 Crear Model

```
[11]: linearRegression = LinearRegression()
linearRegression.fit(X_train,y_train)
y_pred = linearRegression.predict(X_test)
```

```
[12]: r2_test = r2_score(y_test,y_pred)
m = mean_squared_error(y_test, y_pred)

print("R2 Score :",round(r2_test,2), "\nMean Error:", round(math.sqrt(m),2))
```

R2 Score : 0.88

Mean Error: 0.34

0.0.5 Predir dades

```
[13]: print("Valors possibles per cada variable")
print(art['TIPOLOGÍA'].value_counts(), end="\n\n")
print(art['ESTILO'].value_counts(),end="\n\n")
print(art['MATERIA/TÉCNICA'].value_counts(),end="\n\n")
print(art['ESTADO DE CONSERVACIÓN'].value_counts(),end="\n\n")
print(art['TIPO'].value_counts(),end="\n\n")
print(art['CRONOLOGÍA'].value_counts(),end="\n\n")
```

Valors possibles per cada variable

PINTURA 276

ESCULTURA 26

FOTOGRAFÍA 14

DIBUJO 3

Name: TIPOLOGÍA, dtype: int64

CONTEMPORÁNEO 319

Name: ESTILO, dtype: int64

ÓLEO SOBRE LIENZO 90

OTRO 75

DIBUJO SOBRE PAPEL 43

ACUARELA SOBRE PAPEL 33

MIXTA SOBRE TABLA 28

IMPRESIÓN DIGITAL EN COLOR 14

ÓLEO SOBRE TABLA 13

PUNTA SECA SOBRE PAPEL 13

MIXTA SOBRE LIENZO 10

Name: MATERIA/TÉCNICA, dtype: int64

BUENO/ENMARCADO 225

BUENO/SIN RESTAURAR 27

BUENO/SIN ENMARCAR 25

```
REGULAR/ENMARCADO      22
REGULAR/SIN ENMARCAR   15
REGULAR/SIN RESTAURAR   5
Name: ESTADO DE CONSERVACIÓN, dtype: int64
```

```
OTRO      165
PAISAJE    90
ABSTRACTA  33
RETRATO    31
Name: TIPO, dtype: int64
```

```
SIGLO XX    173
SIGLO XXI   146
Name: CRONOLOGÍA, dtype: int64
```

Cal canviar els valors d'obra que es vol predir

```
[17]: obra = ['PINTURA', 'CONTEMPORÁNEO', 'ÓLEO SOBRE LIENZO', 'BUENO/SIN_
↳RESTAURAR', 'ABSTRACTA', '10000', 'SIGLO XX']

test.loc[-1] = obra

X_test, _, _ = preprocessing(test, scaler=scaler)
print('Puntuació Final:', round(linearRegression.predict(X_test)[-1], 2))
```

Puntuació Final: 7.37