REGRESSÃO

Cada problema tem o seu domínio próprio e suas características. Estas características se refletem no conjunto de dados deste domínio. Para exemplificar, vamos supor que queiramos criar um sistema de regressão para predizer o valor de UPDRS baseado nos dados fornecidos. (“*Unified Parkinson’s Disease Rating Scale (**UPDRS**) is the most widely used measure to assess motor symptoms and signs in Parkinson’s disease (PD”).(*https://www.hindawi.com/journals/pd/2012/719167/).

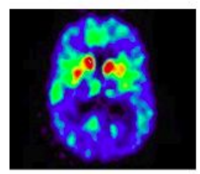
Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Parkinsons Telemonitoring Data Set

Download: Data Folder, Data Set Description

Abstract: Oxford Parkinson's Disease Telemonitoring Dataset



Data Set Characteristics:	Multivariate	**Number of Instances:**	5875	**Area:**	Life
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	26	**Date Donated**	2009-10-29
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	111555

https://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring

O segundo dataset é o descrito abaixo:

About Citation Policy Donate a Data S

Repository Web

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

View ALL D

Physicochemical Properties of Protein Tertiary Structure Data Set

Download: Data Folder, Data Set Description

Abstract: This is a data set of Physicochemical Properties of Protein Tertiary Structure. The data set is taken from CASP 5-9. There are 45730 decoys and size varying from 0 to 21 armstrong.

Data Set Characteristics:	Multivariate	**Number of Instances:**	45730	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	9	**Date Donated**	2013-03-31
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	37712

https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure#

Problema a ser resolvido

Nesta caso, o problema é construir um componente de machine learning que seja capaz de estimar os valores de saída baseado nos valores de entrada.

Solução do Problema

As principais métricas para regressão são:

MSE (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html)
R2   (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html)

Como avaliar estas métricas?

	MSE	R2
Melhor valor	0	1

Tabela 2: Métricas para Regressão.

Escolha do Algoritmo de Regressão

```python
regress = {"Boost": ensemble.GradientBoostingRegressor(**params),
             "RF": RandomForestRegressor(n_estimators=1000,n_jobs=-1),
            "LINR": linear_model.SGDRegressor(max_iter=500, tol=1e-6),
            "MLP": MLPRegressor(hidden_layer_sizes=(100 ),max_iter=500),
             "svr_rbf": SVR(kernel='rbf', C=1e3, gamma=0.1),
             'svr_li':SVR(kernel='linear', C=1e3),
             'svr_poly':SVR(kernel='poly', C=1e3, degree=2)

            }
```

Testamos vários algoritmos de regressão e verificamos qual deles tem o melhor R2 e o menor MSE.

Fazendo a codificação em Python 3.6 temos:

```python
#REGRESSAO
import pandas as pd
from sklearn import linear_model
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split, KFold
from sklearn.neural_network import MLPClassifier, MLPRegressor
from sklearn.svm import SVR
from sklearn import ensemble
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale

df = pd.read_csv('parkinsons_updrs.csv')
cols = df.columns.tolist()
motor_UPDRS = cols.pop(4)
total_UPDRS = cols.pop(4)
cols.append(motor_UPDRS)
cols.append(total_UPDRS)
df = df[cols]
df = df.dropna()
y1 = df['motor_UPDRS'].values
X1 = df.loc[:,'Jitter(%)':'PPE'].values

df = pd.read_csv('CASP.csv')
df = df.dropna()
y = df['RMSD']
X = df.loc[:,'F1':'F9']


X = scale(X)
pca = PCA(n_components=6)
Xpca = pca.fit_transform(X)
X = Xpca


nfolds = 10
kf = KFold(n_splits=nfolds,shuffle=True)
params = {'n_estimators': 500, 'max_depth': 4, 'min_samples_split': 2,
          'learning_rate': 0.0001, 'loss': 'ls'}

regress = {"Boost": ensemble.GradientBoostingRegressor(**params),
             "RF": RandomForestRegressor(n_estimators=1000,n_jobs=-1),
           "LINR": linear_model.SGDRegressor(max_iter=500, tol=1e-6),
           "MLP": MLPRegressor(hidden_layer_sizes=(100 ),max_iter=500),
             "svr_rbf": SVR(kernel='rbf', C=1e3, gamma=0.1),
             'svr_li':SVR(kernel='linear', C=1e3),
             'svr_poly':SVR(kernel='poly', C=1e3, degree=2)

           }


for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    for name, regr in regress.items():
        # Train the model using the training sets
        regr.fit(X_train, y_train)
        # Make predictions using the testing set
        y_pred = regr.predict(X_test)
        print(name)
        print("MSE: %.2f" % mean_squared_error(y_test, y_pred))
        print('R2 score: %.2f, 1 is the best! ' % r2_score(y_test, y_pred))
```

Resultados do Modelo criado:

Resultados Pasta 1:
BOOST
MSE: 18.79
R2 score: 0.49, 1 is the best!
RF
MSE: 14.25
R2 score: 0.62, 1 is the best!
LINR
MSE: 26.00
R2 score: 0.30, 1 is the best!
MLP
MSE: 19.34
R2 score: 0.48, 1 is the best!

Resultados Pasta 2:
BOOST
MSE: 18.61
R2 score: 0.50, 1 is the best!
RF
MSE: 13.93
R2 score: 0.62, 1 is the best!
LINR
MSE: 26.24
R2 score: 0.29, 1 is the best!
MLP
MSE: 19.18
R2 score: 0.48, 1 is the best!

Resultados Pasta 3:
BOOST
MSE: 18.91
R2 score: 0.50, 1 is the best!
RF
MSE: 14.07
R2 score: 0.63, 1 is the best!
LINR
MSE: 27.17
R2 score: 0.28, 1 is the best!
MLP
MSE: 19.62
R2 score: 0.48, 1 is the best!

Resultados Pasta 4:
BOOST
MSE: 19.09
R2 score: 0.50, 1 is the best!
RF
MSE: 14.13
R2 score: 0.63, 1 is the best!
LINR
MSE: 27.59
R2 score: 0.27, 1 is the best!

Resultados Pasta 5:
BOOST
MSE: 19.42
R2 score: 0.49, 1 is the best!
RF
MSE: 14.34
R2 score: 0.62, 1 is the best!
LINR
MSE: 27.33
R2 score: 0.28, 1 is the best!
MLP
MSE: 19.78
R2 score: 0.48, 1 is the best!

Resultados Pasta 6:
BOOST
MSE: 19.31
R2 score: 0.49, 1 is the best!
RF
MSE: 14.29
R2 score: 0.62, 1 is the best!
LINR
MSE: 27.24
R2 score: 0.28, 1 is the best!
MLP
MSE: 19.84
R2 score: 0.47, 1 is the best!

Resultados Pasta 7:
BOOST
MSE: 19.43

R2 score: 0.48, 1 is the best!
RF
MSE: 14.75
R2 score: 0.61, 1 is the best!
LINR
MSE: 27.14
R2 score: 0.27, 1 is the best!
MLP
MSE: 20.18
R2 score: 0.46, 1 is the best!

Resultados Pasta 8:
BOOST
MSE: 18.86
R2 score: 0.48, 1 is the best!
RF
MSE: 13.76
R2 score: 0.62, 1 is the best!
LINR
MSE: 27.13
R2 score: 0.26, 1 is the best!
MLP
MSE: 20.18
R2 score: 0.46, 1 is the best!

Resultados Pasta 9:
BOOST
MSE: 18.86
R2 score: 0.48, 1 is the best!
RF
MSE: 13.76
R2 score: 0.62, 1 is the best!
LINR
MSE: 27.13
R2 score: 0.26, 1 is the best!
MLP
MSE: 19.56
R2 score: 0.47, 1 is the best!

Resultados Pasta 10:
BOOST
MSE: 19.34
R2 score: 0.47, 1 is the best!

RF
MSE: 14.88
R2 score: 0.60, 1 is the best!
LINR
MSE: 27.32
R2 score: 0.26, 1 is the best!
MLP
MSE: 20.06
R2 score: 0.45, 1 is the best!

Pode-se observar que para cada algoritmo as métricas são diferentes mas RF apresenta melhor desempenho para esta caso específico. Portanto, para utilizarmos em produção, criaríamos um modelo utilizando RF.
Para realizarmos o Deploy, fazemos exatamente como foi feito no processo de classificação.

Exercício - Criar um objeto de predição utilizando os mesmos datasets porém, com redução de dimensionalidade utilizando PCA e MDS. Compare os resultados.
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html