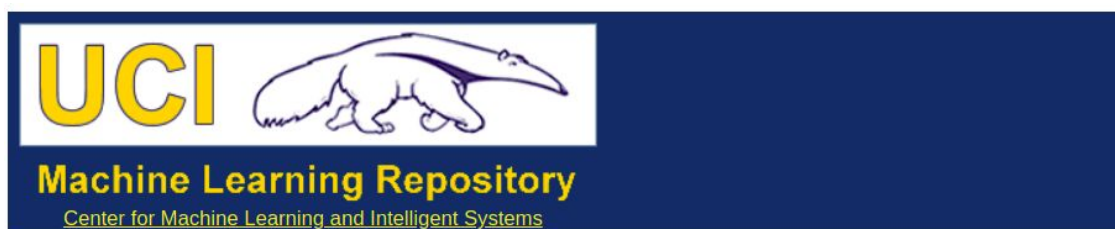


REGRESSÃO

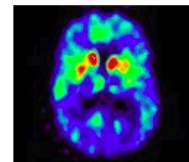
Cada problema tem o seu domínio próprio e suas características. Estas características se refletem no conjunto de dados deste domínio. Para exemplificar, vamos supor que queiramos criar um sistema de regressão para predizer o valor de UPDRS baseado nos dados fornecidos. (*“Unified Parkinson’s Disease 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/>).



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:



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

```
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:

```

1  #REGRESSAO
2  import pandas as pd
3  from sklearn import linear_model
4  from sklearn.ensemble import RandomForestRegressor
5  from sklearn.metrics import mean_squared_error, r2_score
6  from sklearn.model_selection import train_test_split, KFold
7  from sklearn.neural_network import MLPClassifier, MLPRegressor
8  from sklearn.svm import SVR
9  from sklearn import ensemble
10 from sklearn.decomposition import PCA
11 from sklearn.preprocessing import scale
12
13 df = pd.read_csv('parkinsons_updrs.csv')
14 cols = df.columns.tolist()
15 motor_UPDRS = cols.pop(4)
16 total_UPDRS = cols.pop(4)
17 cols.append(motor_UPDRS)
18 cols.append(total_UPDRS)
19 df = df[cols]
20 df = df.dropna()
21 y1 = df['motor_UPDRS'].values
22 X1 = df.loc[:, 'Jitter(%)': 'PPE'].values
23
24 df = pd.read_csv('CASP.csv')
25 df = df.dropna()
26 y = df['RMSD']
27 X = df.loc[:, 'F1': 'F9']
28
29
30 X = scale(X)
31 pca = PCA(n_components=6)
32 Xpca = pca.fit_transform(X)
33 X = Xpca
34
35
36 nfolds = 10
37 kf = KFold(n_splits=nfolds, shuffle=True)
38 params = {'n_estimators': 500, 'max_depth': 4, 'min_samples_split': 2,
39           'learning_rate': 0.0001, 'loss': 'ls'}
40
41 regress = {"Boost": ensemble.GradientBoostingRegressor(**params),
42            "RF": RandomForestRegressor(n_estimators=1000, n_jobs=-1),
43            "LINR": linear_model.SGDRegressor(max_iter=500, tol=1e-6),
44            "MLP": MLPRegressor(hidden_layer_sizes=(100), max_iter=500),
45            "svr_rbf": SVR(kernel='rbf', C=1e3, gamma=0.1),
46            "svr_li": SVR(kernel='linear', C=1e3),
47            "svr_poly": SVR(kernel='poly', C=1e3, degree=2)}
48
49
50
51
52 for train_index, test_index in kf.split(X):
53     X_train, X_test = X[train_index], X[test_index]
54     y_train, y_test = y[train_index], y[test_index]
55
56     for name, regr in regress.items():
57         # Train the model using the training sets
58         regr.fit(X_train, y_train)
59         # Make predictions using the testing set
60         y_pred = regr.predict(X_test)
61         print(name)
62         print("MSE: %.2f" % mean_squared_error(y_test, y_pred))
63         print("R2 score: %.2f, 1 is the best! " % r2_score(y_test, y_pred))
64

```

Um fragmento da saída da máquina de regressão é:

```
Boost
MSE: 36.39
R2 score: 0.03, 1 is the best!
RF
MSE: 14.89
R2 score: 0.60, 1 is the best!
LINR
MSE: 28.06
R2 score: 0.25, 1 is the best!
MLP
MSE: 21.39
R2 score: 0.43, 1 is the best!
svr_rbf
MSE: 23.21
R2 score: 0.38, 1 is the best!
```

Pode-se observar que para cada algoritmo as métricas são diferentes.

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>