MACHINE LEARNING

# Comitês de Modelos

cognitiza

Your Data & AI school

Prof. Altino Dantas

# Ensemble

- Comitês baseados em média;
  - Vários modelos opinam sobre uma amostra e escolhe-se uma resposta;
  - Diferentes estratégias: votação, construção de um novo classificador, média das predições, etc;
  - No geral, funcionam porque ajudam a reduzir a variância dos modelos.
- Comitês baseado em impulsionamento (boosting);
  - A ideia é que a combinação de vários modelos fracos pode gerar um conjunto forte;
- A biblioteca sklearn possui várias implementações de comitês;

# Random Forest

- sklearn.ensemble.RandomForestClassifier
- Criam um conjunto de árvores de decisão, cada uma gerada por certas perturbações no processo de construção;
- A predição do conjunto é dada pela média (ou moda) das predições de cada árvore isolada;

```
from sklearn.datasets  import load_iris
from sklearn.ensemble  import RandomForestClassifier

X, y = load_iris(return_X_y= True)

clf = RandomForestClassifier(n_estimators= 10,
random_state= 42)
from sklearn.model_selection  import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
     X, y, stratify=y, random_state= 42
 )
clf.fit(X_train, y_train).score(X_test, y_test)
```

# Stack - pilha de modelos

- sklearn.ensemble.StackingClassifier
- Reúne um conjunto de modelos e adiciona um classificador no final para combinar as saídas do modelos;
- Verifique a documentação para checar parâmetros do método.

```
from sklearn.datasets import load iris
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make pipeline
from sklearn.ensemble import StackingClassifier
X, y = load iris(return_X_y= True)
estimators = [
     ('rf', RandomForestClassifier(n_estimators= 10,
random state= 42)),
     ('svr', make_pipeline(StandardScaler(),

LinearSVC(random_state= 42)))
 ]
clf = StackingClassifier(
     estimators=estimators,
final_estimator=LogisticRegression()
 )
from sklearn.model selection import train test split
X train, X test, y train, y test = train_test_split(
     X, y, stratify=y, random_state= 42
 )
clf.fit(X_train, y_train).score(X_test, y_test)
```

# Votação

- sklearn.ensemble.VotingClassifier
- Faz um eleição entre os modelos;
- Pode ser por regra majoritária simples ou suavizada;

```
import numpy as np
from sklearn.linear model  import LogisticRegression
from sklearn.naive bayes  import GaussianNB
from sklearn.ensemble  import RandomForestClassifier,
VotingClassifier

clf1 = LogisticRegression(multi_class= 'multinomial',
random state=1)
clf2 = RandomForestClassifier(n_estimators= 50,
random state=1)
clf3 = GaussianNB()

X = np.array([[ -1, -1], [-2, -1], [-3, -2], [1, 1],
[2, 1], [3, 2]])
y = np.array([ 1, 1, 1, 2, 2, 2])

eclf1 = VotingClassifier(estimators=[
         ('lr', clf1), ('rf', clf2), ('gnb', clf3)],
voting='hard')
eclf1 = eclf1.fit(X, y)

print(eclf1.predict(X))
```

# Ada Boost

- sklearn.ensemble.AdaBoostClassifier
- Ajusta um classificador nos dados originais depois cria diferentes versões do classificador e faz ajustes nas versões considerando as classificações incorretas nos dados originais;

```
from sklearn.datasets  import load_iris
from sklearn.ensemble  import AdaBoostClassifier
from sklearn.model_selection  import
train_test_split

X, y = load_iris(return_X_y= True)

clf = AdaBoostClassifier(n_estimators= 1000,
random_state= 0)

X_train, X_test, y_train, y_test =
train_test_split(
     X, y, stratify=y, random_state= 42
 )
clf.fit(X_train, y_train).score(X_test, y_test)
```

# Outros...

- A lista exaustiva de métodos de *ensemble* do *sklearn* pode ser encontrada aqui:

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

# Exercício

Com o dataset diabetes, utilize alguma estratégia de *ensemble* que possa melhorar a acurácia e/ou a matriz de confusão final das predições, quando comparadas com um único algoritmo.

Por exemplo, mostre que o resultado produzido por um *ensemble,* composto por SMV, MLP e Random Forest é melhor do que uma MLP isolado.

# Obrigado.