BUT Informatique 2ème année Parcours A

R3.A.15: Machine Learning

2023-2024

Sébastien Lefèvre
sebastien.lefevre@univ-ubs.fr

Vannes iut : Université Bretagne Sud

iutvannes.fr

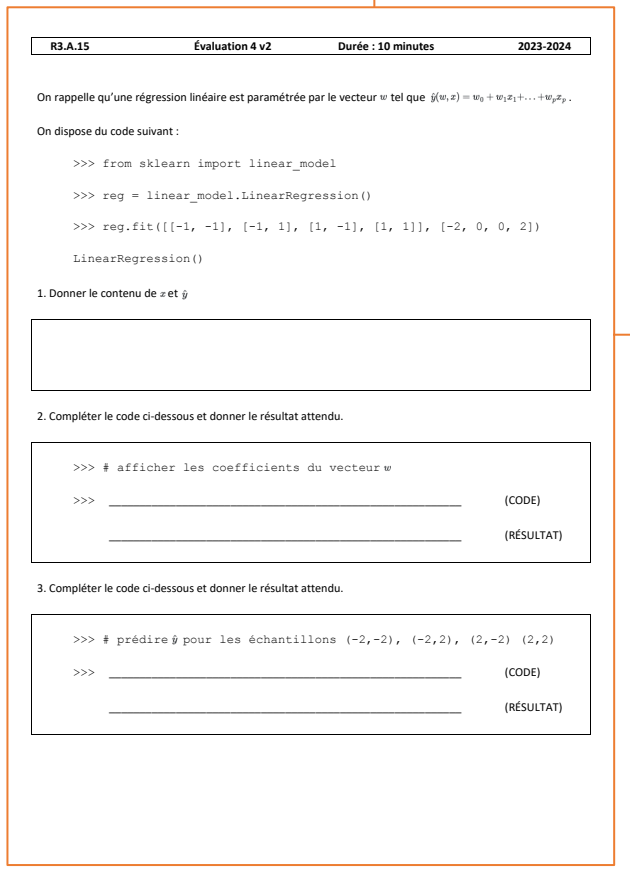
Evaluation

- Correction en séance
- Discussion : modalité d'évaluation

R3.A.15	Évaluation 4 v1	Durée : 10 minutes	2023-2024

Expliquer le principe d'un des algorithmes de régression vus lors de la séance précédente.

Nom de l'algorithme : *(Name in english :)*



R3.A.15	Évaluation 4 v2	Durée : 10 minutes	2023-2024

On rappelle qu'une régression linéaire est paramétrée par le vecteur w tel que $\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$.

On dispose du code suivant :

```
>>> from sklearn import linear_model
>>> reg = linear_model.LinearRegression()
>>> reg.fit([[-1, -1], [-1, 1], [1, -1], [1, 1]], [-2, 0, 0, 2])
LinearRegression()
```

1. Donner le contenu de x et $\hat{y}$

2. Compléter le code ci-dessous et donner le résultat attendu.

```
>>> # afficher les coefficients du vecteur w
>>> _____ (CODE)
    _____ (RÉSULTAT)
```

3. Compléter le code ci-dessous et donner le résultat attendu.

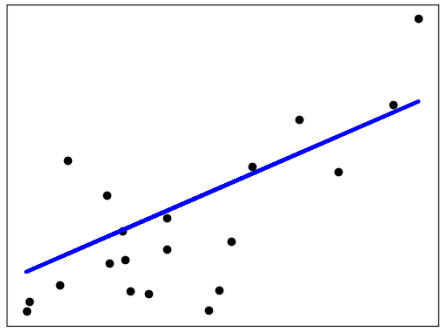
```
>>> # prédire ŷ pour les échantillons (-2,-2), (-2,2), (2,-2) (2,2)
>>> _____ (CODE)
    _____ (RÉSULTAT)
```

Vannes iut : Université Bretagne Sud

Bilan séance 4

1. Découverte des principaux algorithmes de régression

2. Mise en œuvre sous scikit-learn



```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
# Use only one feature
diabetes_X = diabetes_X[:, np.newaxis, 2]
# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
# Split the targets into training/testing sets
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
# Create linear regression object
regr = linear_model.LinearRegression()
# Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)
# Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)
# The coefficients
print("Coefficients: \n", regr.coef_)
# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test, diabetes_y_pred))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination: %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))
# Plot outputs
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)
plt.xticks(())
plt.yticks(())
plt.show()
```

Séance 5 : TP en 2h15 (tiers-temps : 3h)

Un jeu de données publiques vous est fourni en début de séance.

Réaliser une analyse de ce jeu de données, en exploitant notamment les méthodes de classification (supervisée ou non), régression ou autre qui sont disponibles sous scikit-learn.

Le rendu (à la fin de la séance) prendra la forme d'un Notebook Jupyter qui sera évalué selon différents critères : qualité et logique du raisonnement, clarté des explications et des illustrations, qualité du code Python, interprétation et discussion des résultats...

Vannes
iut :
Université Bretagne Sud

Bilan du module

Mise en œuvre pédagogique

- CM
- TP

Evaluation

- (rendus hebdomadaires)
- CC : évaluations CM + TP
- CT : sujet à discuter

Contenu

- Notions qu'il aurait fallu aborder
- Notions qu'il n'aurait pas fallu aborder

Différence avec enseignement classique

- Investissement
- Apprentissage

Vannes
iut
Université Bretagne Sud