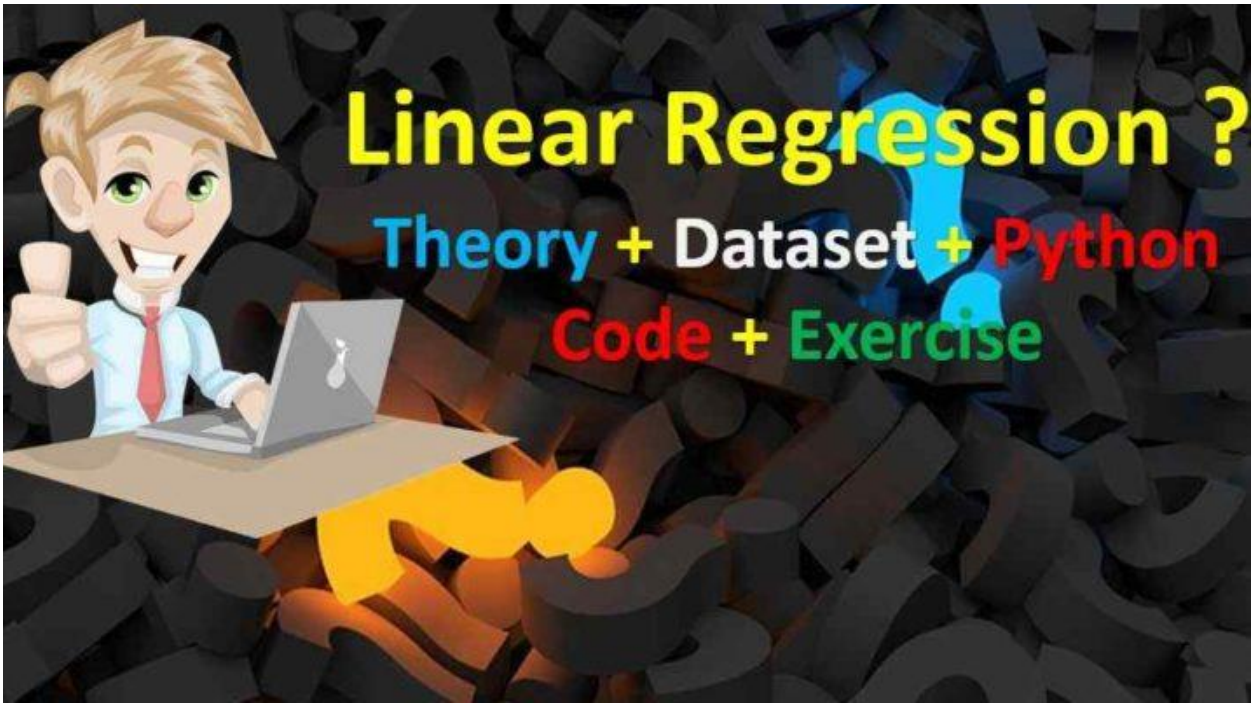
Prediction du salaire

Un Exemple de régression Multiple



Objectifs pédagogiques

Objectif 1 : Préparer un jeu de donnée pour l'analyse.

Objectif 2 : Construire un modèle de prédiction de profit selon différentes variables quantitatives et qualitatives.

Objectif 3 : interpréter la qualité du modèle de prédiction.

Objectif 4 : Prédire les nouvelles entrées

Prérequis

Installation de librairie ***sklearn***

Introduction

La régression linéaire est une partie de l'apprentissage automatique supervisé. La régression linéaire est la meilleure ligne d'ajustement pour le point de données donné. Elle fait référence à une relation linéaire (ligne droite) entre les variables indépendantes et les variables dépendantes.

Plan de travail

1. Importez la bibliothèque Python nécessaire.
2. Stocker les données (csv) dans une variable.
3. Fractionner les données en données d'apprentissage et de test.
4. Ajuster le modèle de régression aux données d'apprentissage et prévoir les données de test.
5. Visualisez le tracé entre les données d'entraînement / test par rapport à la ligne de régression.
6. Calculer l'erreur quadratique moyenne et sa racine carrée (MSE et RMSE).

Chargement de corpus

```
# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the datasetdataset =
pd.read_csv('50_Startups.csv')

X = dataset.iloc[:, :-1]

y = dataset.iloc[:, 4]

states = X.State.values.reshape(-1, 1)
```

```
onehotencoder = OneHotEncoder(handle_unknown='ignore')
dummies = onehotencoder.fit_transform(states).toarray()
X.drop('State', axis=1, inplace=True)
X = X.join(pd.DataFrame(dummies,
columns=onehotencoder.categories_[0]))
```

RQ : essayer de visualiser vos données avec préparés

Fractionner les données

Avant d'entamer cette partie, nous allons tout d'abord répartir notre corpus en données d'apprentissage et en données de test.

1. La fonction ***train_test_split*** de librairie ***sklearn.model_selection*** fait l'affaire.

```
# Importing the dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size = 1/3, random_state = 0)
```

- RQ : Essayer de visualiser vos données X_train, y_train

Ajuster le modèle de régression aux données d'apprentissage et prévoir les données de test.

Maintenant que nous avons nos variables et nos données d'apprentissage, nous pouvons former un modèle pour essayer de prédire le salaire.

Les instructions ci-dessus sert à créer un modèle de Régression Linéaire.

Scikit-learn, notre Framework de prédilection, fournit d'ailleurs une aide au choix de l'algorithme :

```
# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

- Tester le modèle et afficher y_pred,y_test

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

Visualiser le résultat

Prédition des nouveaux salaires

Une fois le modèle est créé, on passe à la prédiction :

```
fl = onehotencoder.transform([['Florida']]).toarray()[0]
test = [50000, 190000, 38000]
test += list(fl)
print(regressor.predict([test]))
```

```
# Pour calculer la performance
from sklearn.metrics import r2_score
coefficient_of_dermination = mean_squared_error(y_test,
y_pred)
r2 = r2_score(y_test, y_pred)
print(r2)
```