DataSet : 50\_StartUps

Description for mUlti linear regression :::

50 Companies :::Total

Columns :::

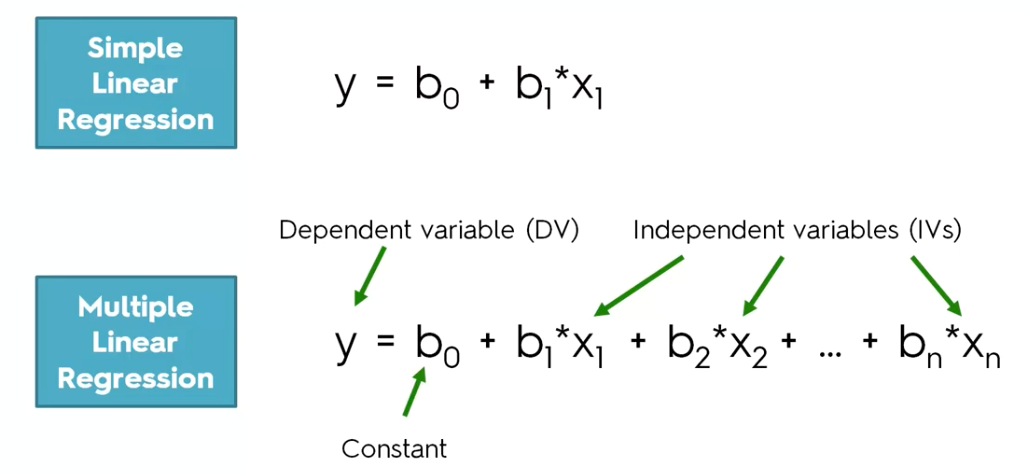
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R&D Spend | Administration | Marketing Spend | State | Profit |

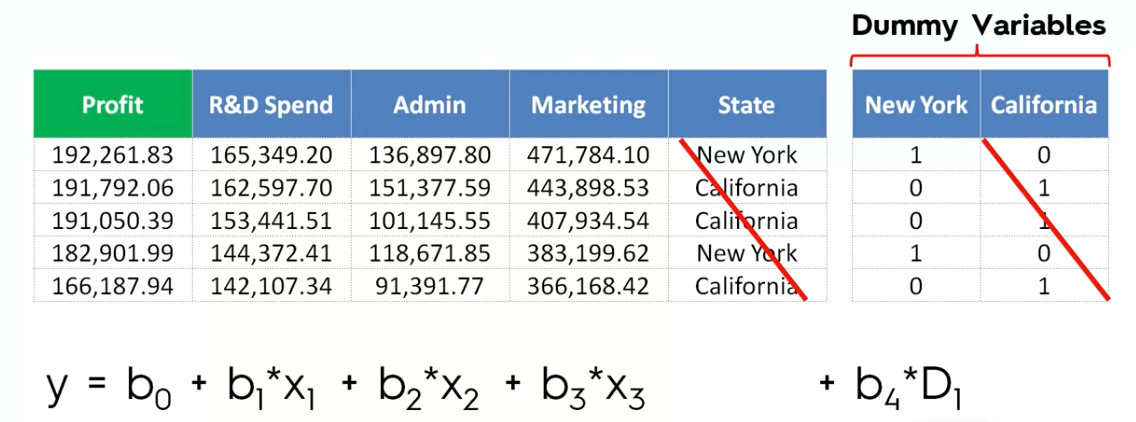
Profit Is Dependend Variable:::

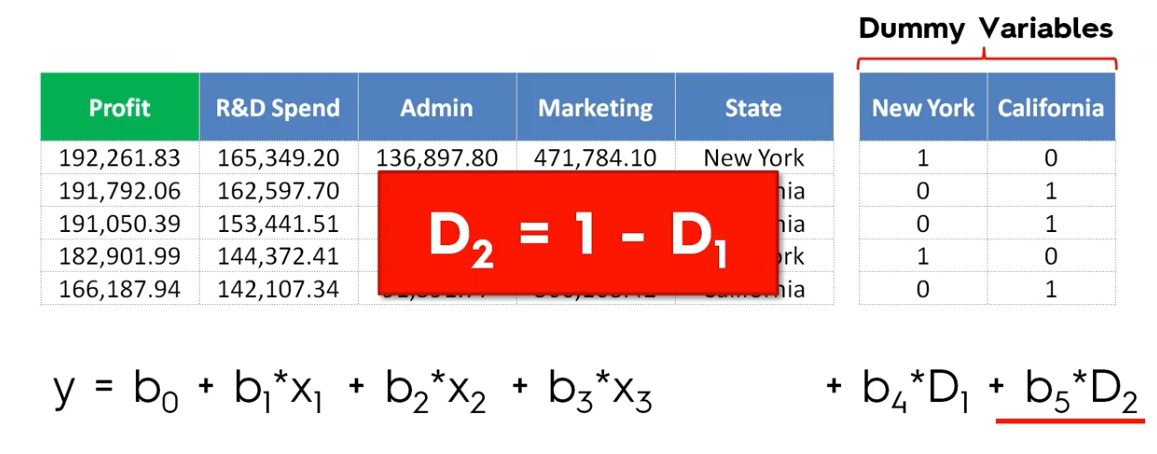
All other are independent variables :::

Challenge ::: Which is best company ::: Prepare a model that should say best To Invest :::

In Take R&D Spend :::Marketing spend ::: for analysis



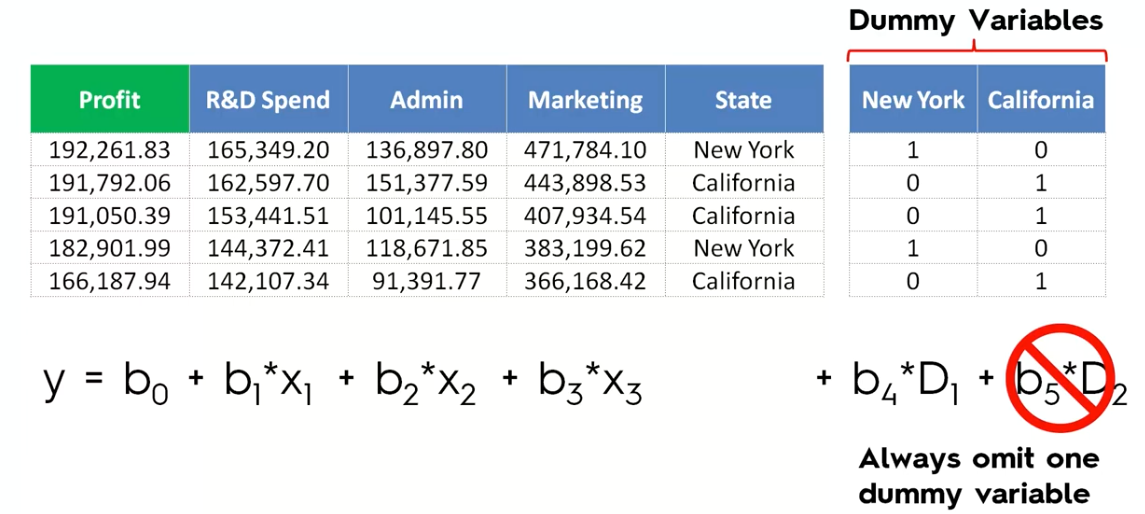


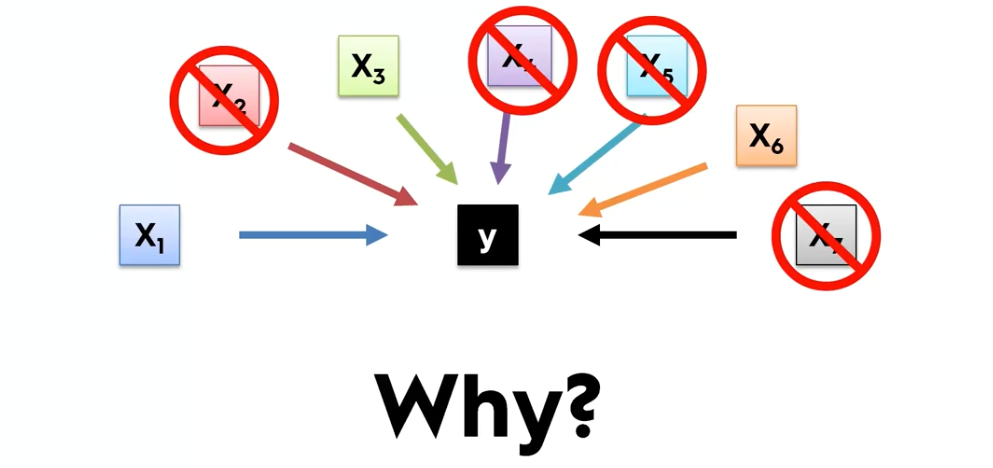


If you have 100 Dummy variables---Declare 99 –only

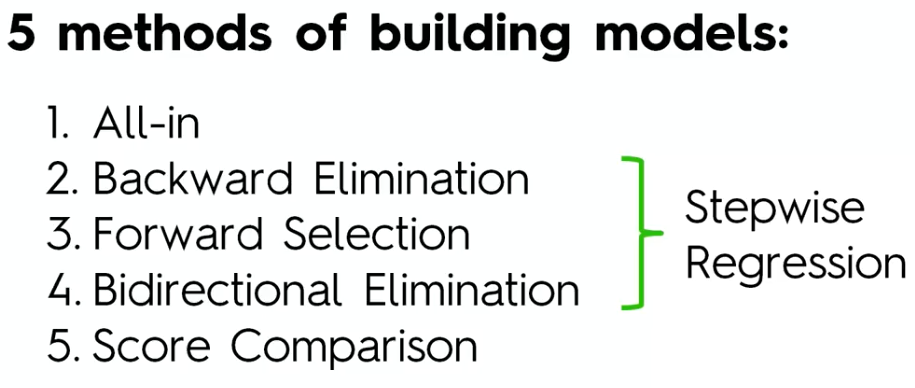
If you have 50 Dummy variables --- Declare 49—only

If you have 2 Dummy variables – Declare 1—Only

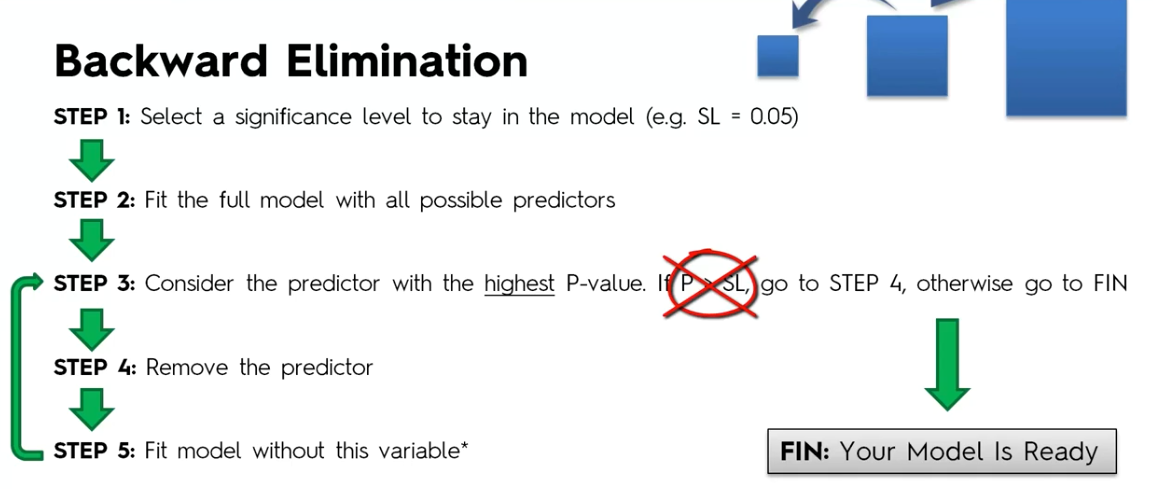


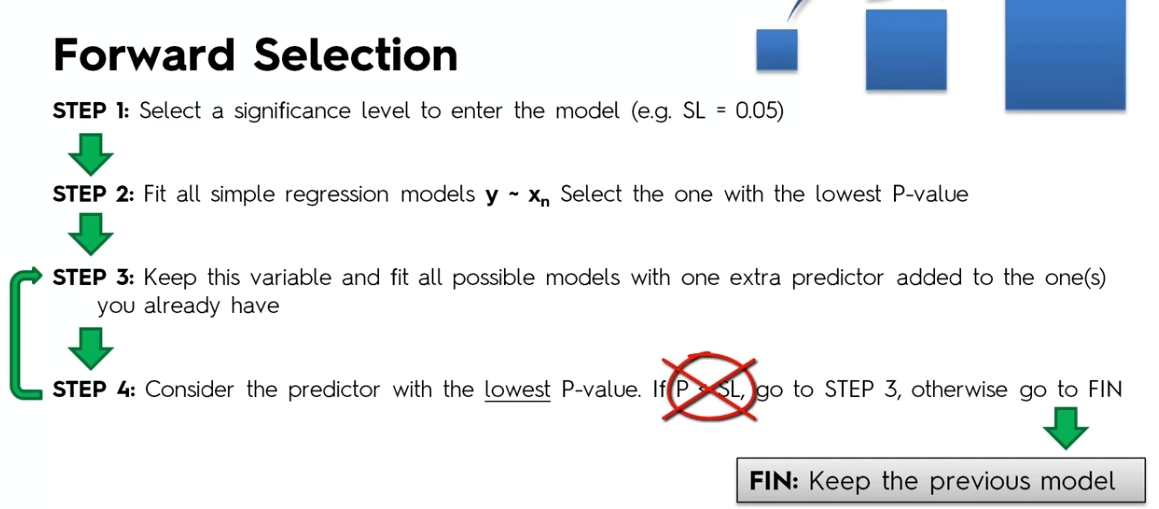


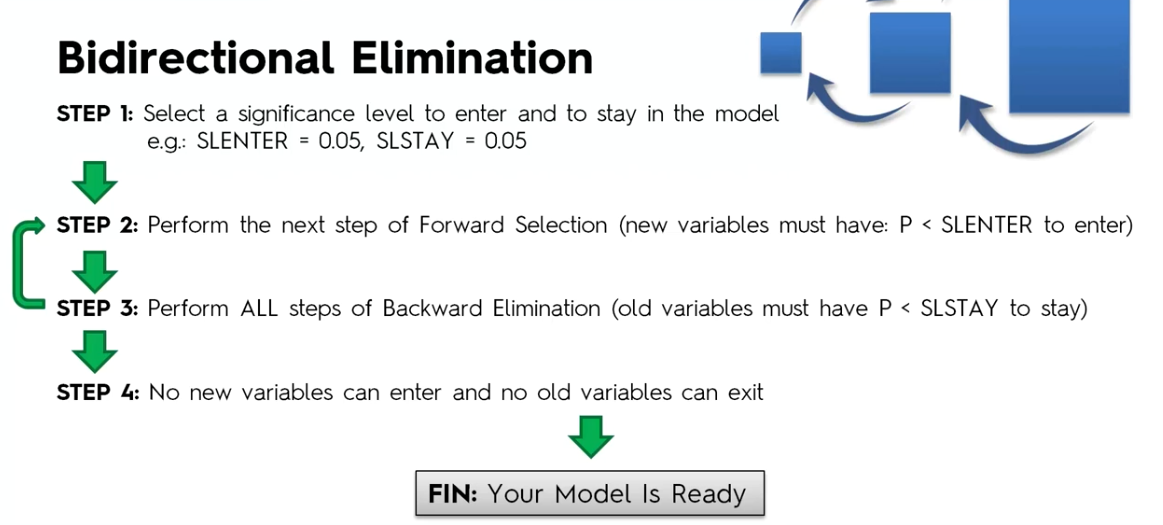
================================================================

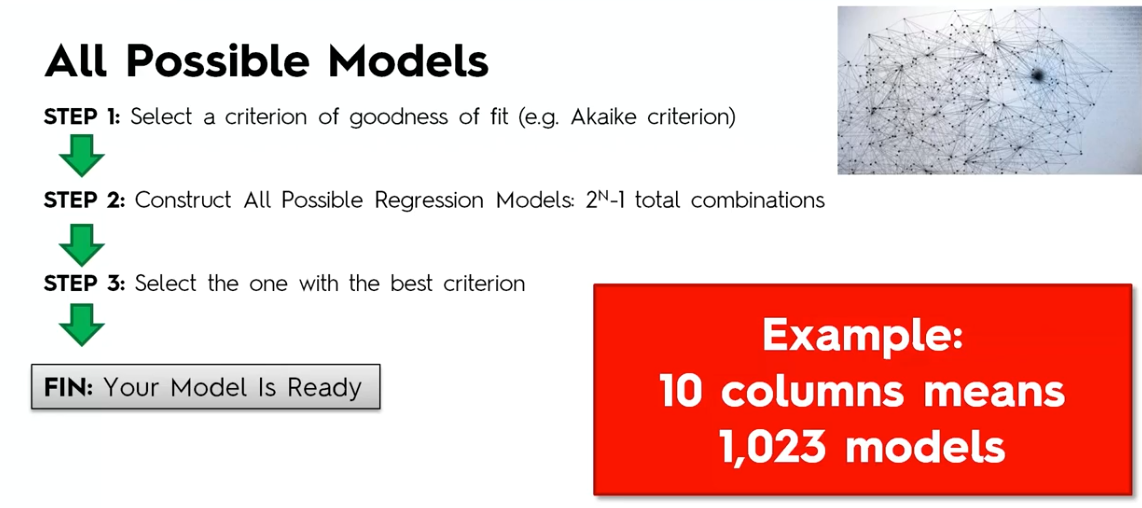


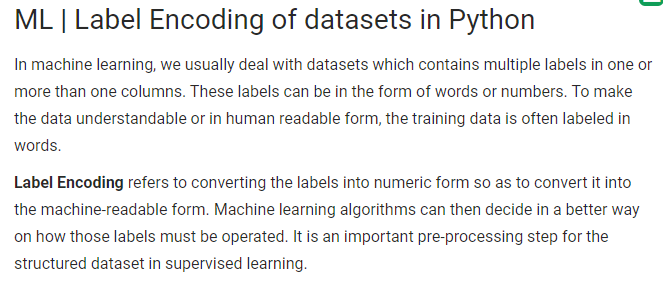


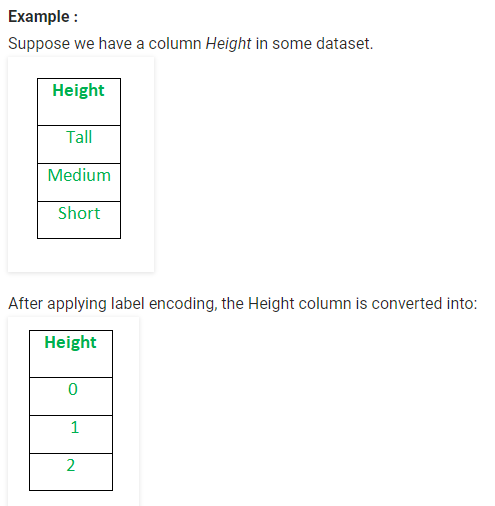




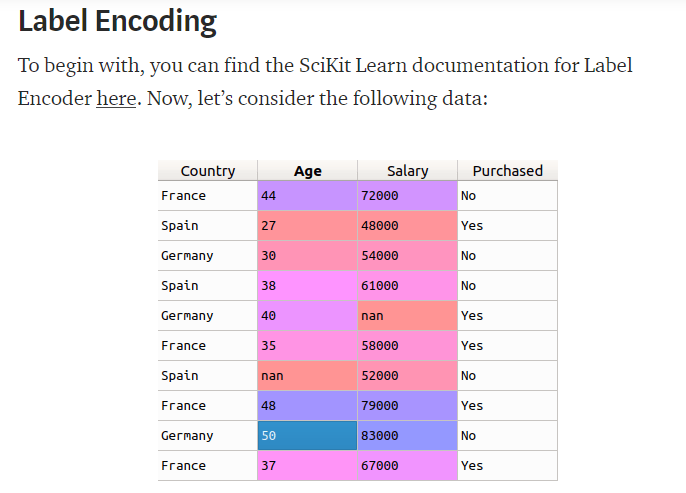


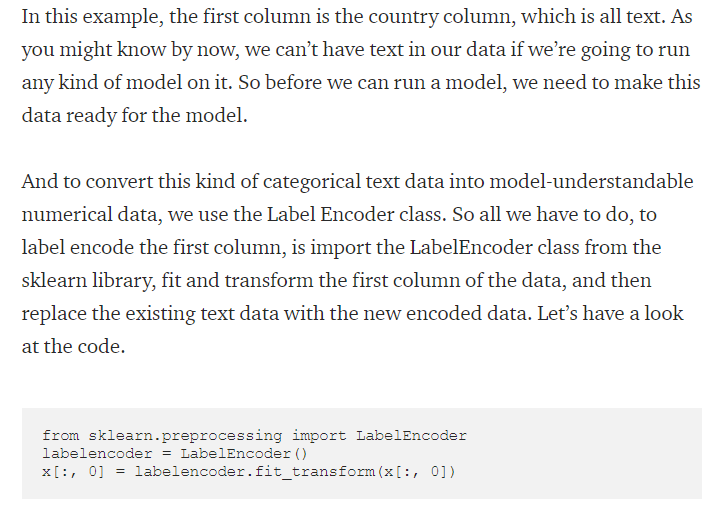


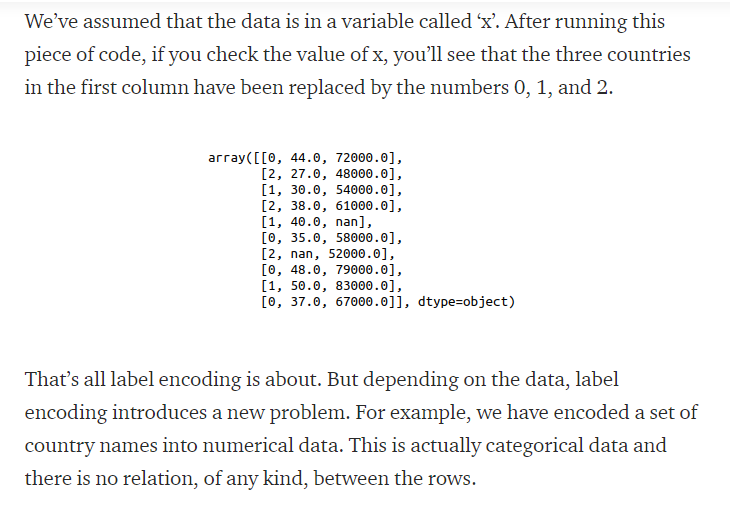


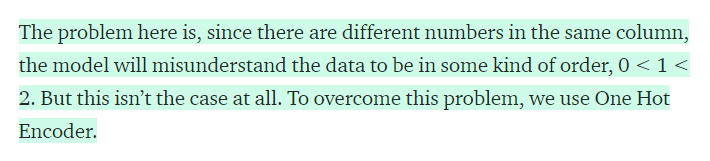


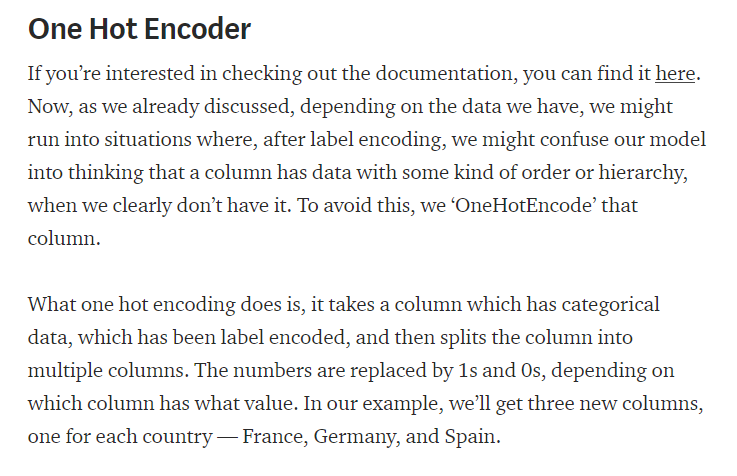
Label Encoder Example 2:::

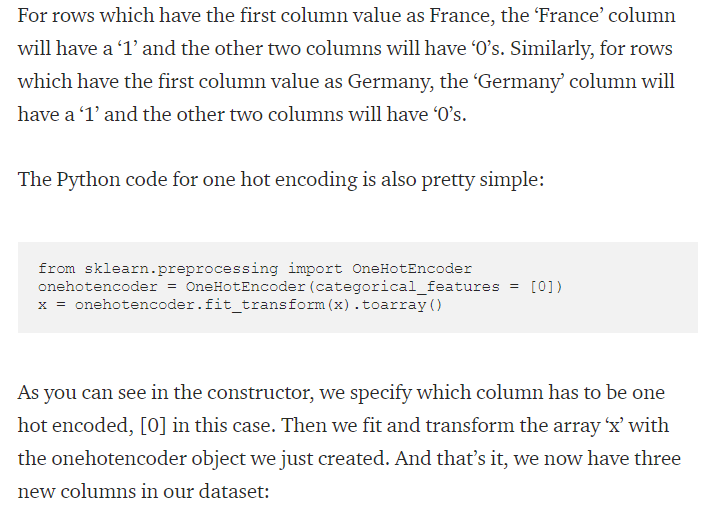


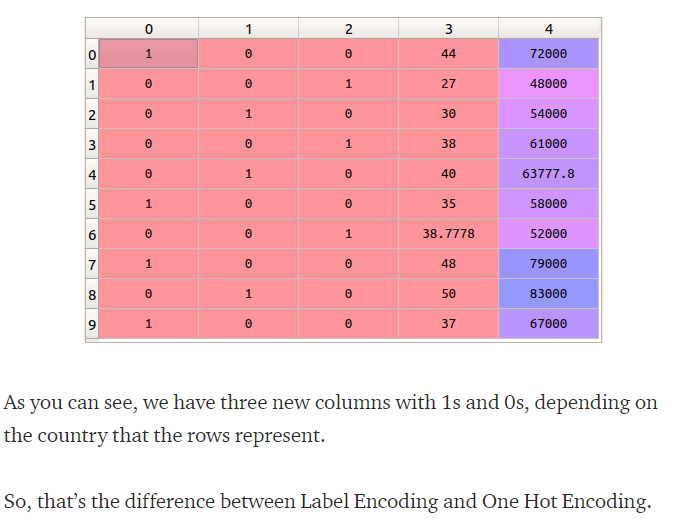












Avoiding

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('50\_Startups.csv')

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

y = dataset.iloc[:, 4].values

X

# Encoding categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder = LabelEncoder()

X[:, 3] = labelencoder.fit\_transform(X[:, 3])

onehotencoder = OneHotEncoder(categorical\_features = [3])

X = onehotencoder.fit\_transform(X).toarray()

# Avoiding the Dummy Variable Trap

X = X[:, 1:]

X

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

"""from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

sc\_y = StandardScaler()

y\_train = sc\_y.fit\_transform(y\_train)"""

# Fitting Multiple Linear Regression to the Training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = regressor.predict(X\_test)