In the ML model there are many techniques to convert non-numeric values to numerical representation and in this article, we will discuss 2 of the most common techniques used with the scikit-learn library and then how to deploy and use them as pickle file

1. Label Encoder

In this we give numeric representation to string values for example assume that in our dataset predicting 3 lifestyle diseases like Diabetes, Blood Pressure, Obesity.

We can represent them in numbers by assigning numeric labels to each of these 3 values Diabetes -1 , Blood Pressure -2 and Obesity -3

1. One Hot Encoding

Once data have been loaded in a dataframe like pandas then by using one hot encoding those many columns get added in the dataframe for each of the distinct value and if that value occurs in that row then it becomes 1 else 0

For example consider below dataframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Weight (Kg) | Diabetes | Blood Pressure | Obesity |
| 27 | 90 | 0 | 0 | 1 |
| 39 | 80 | 1 | 1 | 0 |
| 56 | 102 | 1 | 0 | 0 |
| 26 | 70 | 0 | 1 | 0 |

In python we can serialize any of the objects that we create during model creation and then use them during deployment to ensure consistency and same mapping.

Assume we have the model created and now we need all the encoders mapping that have been created during model training to be used during deployment of the model as a REST api using python base Django or Flask framework.

The following python code snippet can help in achieving this task

from sklearn.preprocessing import LabelEncoder

import pandas as pd

import joblib

# Label Encoding on Pandas dataframe df

le\_Disease= LabelEncoder()

le\_Disease.fit(df[‘Disease’])

df['Disease\_encoded'] = le\_Disease.transform(df['Disease'])

#Saving the encoder mapping as a file to be used during prediction on input data.

filehandler = open("le\_Disease.obj","wb")

joblib.dump(le\_Disease,filehandler)

filehandler.close()

Now in the deployment code where encoding saved as a file “le\_Disease.obj” is required to transform input dataframe dfInput we can use below code

file = open("le\_Disease.obj",'rb')

le\_ Disease \_loaded = joblib.load(file)

file.close()

dfInput['Disease\_encoded '] = le\_Disease\_loaded.transform(dfInput [' Disease '])

The below coding snippet can be used to convert dataframe in one hot encoding using pandas get dummies method.

import pandas as pd

import joblib

onehot\_dfData=pd.get\_dummies(dfData)

onehot\_dfData = onehot\_dfData.reindex(columns=onehot\_dfData.columns, fill\_value=0)

onehot\_dfData\_cols= onehot\_dfData.columns

#Saving onehot\_dfData columns from dummies as pkl file to be used in api

joblib.dump(onehot\_dfData\_cols, “onehot\_dfData\_cols.pkl')

The below code will be used during deployment to ensure the consistency of mapping of one hot columns generated and saved in pickle file.

#Convert Input dataframe with the same onehot mapping as used during training

onehot\_dfData\_cols = joblib.load(“onehot\_dfData\_cols.pkl')

dfInput\_onehot=pd.get\_dummies(dfInput)

dfInput\_onehot = dfInput\_onehot.reindex(columns= onehot\_dfData\_cols, fill\_value=0)

One need to ensure that once the encoded columns have been generated there may be need to remove the original non-numeric columns.