***Machine Learning Library Sunbird--Categorical Encoding***

# *Sunbird:*

**Sunbird** is a library which can be used for Feature Engineering purposes. This library provides various techniques to handle missing values, outliers, categorical encoding, normalization and standardization (scaling) for the data. Here in this article, we look after the Categorical Encoding techniques that are provided by Sunbird. To install this library, we need to use the following command in Anaconda Prompt in Administration mode:

*pip install sunbird*

# *Categorical Encoding:*

Most of the Machine Learning Algorithms cannot handle categorical variables to compute, since they perform some mathematical operations, for which categorical text data need to be converted to numerical format.

So, converting Categorical text data to Numerical format is called *Categorical Encoding.*

The variables having fixed categories or variables that can be categorized into categories is called *Categorical Variables.* These Categorical variables are of two types: *Nominal(*that cannot be ordered), *Ordinal(*that can be ordered).

Example for Nominal variable:

* Red, Black, White.
* Apple, Papaya, Grapes.

Example for Ordinal variable:

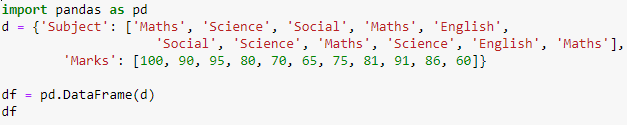
* Bad, Average, Good.
* Low, Medium, High.

Since Sunbird is a new library, let’s see what types of Encoding techniques it is providing and how are they useful.

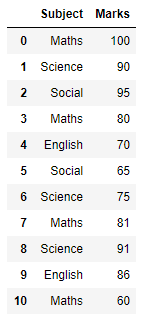
Categorical Encoding techniques available in Sunbird library are:

* Frequency Encoding
* Target Guided Encoding
* Probability Ratio Encoding
* Mean Encoding
* One Hot Encoding
* One Hot Encoding with Multiple Categories

First, we create a Data frame using Pandas DataFrame and perform all above techniques on them.



**Data frame**:



## Frequency Encoding:

In Frequency encoding, the categorical features are replaced with their respective frequency (no. of terms) of each category, which is nothing but *value\_counts().*  This technique is useful when you have large categories with unique frequencies, so that, dimension of data will not get increased.

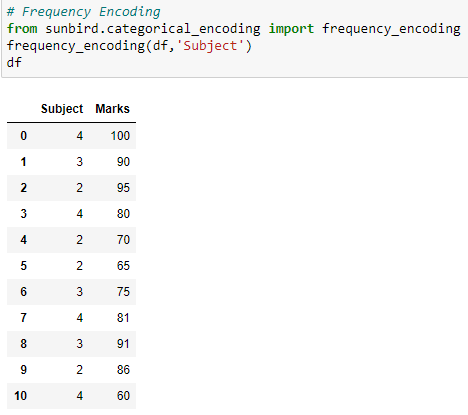
Here, in our data frame, both Social and English have their frequency of 2, which means both are encoded with the same value.

So, this is a disadvantage in this technique, because, categories with same frequency will be having same encoded value. This may confuse our model.

Syntax:

from sunbird.categorical\_encoding import frequency\_encoding

frequency\_encoding(dataframe, ‘categorical\_column\_name’)



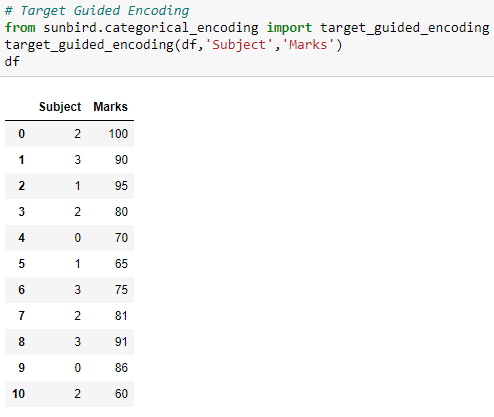
## Target Guided Encoding:

In Target Guided Encoding, it calculates the average of the targets with respective to categories. They are sorted in ascending order and assign values to their positions. The resultant order is encoded to each of the category.

Syntax:

from sunbird.categorical\_encoding import target\_guided\_encoding

target\_guided\_encoding(dataframe, ‘categorical\_column\_name’, ‘target\_column\_name’)



## Probability Ratio Encoding:

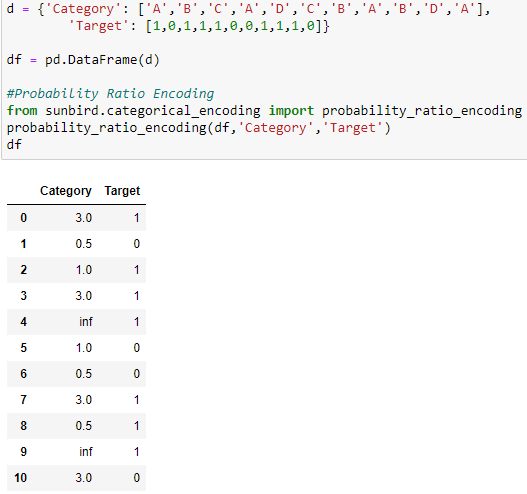
In Probability Ratio Encoding, ratio of good and bad probability between the feature variable and target variable is used. For each category, it calculates the mean of target being 1, that is probability of being 1 (p(1)), and also probability of target being 0 (p(0)). Now, it calculates the ratio of p(1)/p(0) and replace the category by that ratio.

Here, it has a drawback. There is a chance where for any particular category, there is no target being 0, which results the category to get encoded with infinity.

Syntax:

from sunbird.categorical\_encoding import probability\_ratio\_encoding

probability\_ratio\_encoding(dataframe, ‘categorical\_column\_name’, ‘target\_column\_name’)



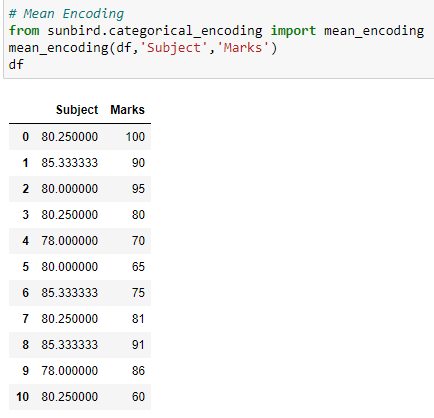
## Mean Encoding:

In Mean Encoding, the categorical variables are encoded with the mean of target with respective to each category. Mean Encoding retains the actual useful information about the categories and does not change the dimensionality of the data. In this case, there is chance of over fitting in the model.

Syntax:

from sunbird.categorical\_encoding import mean\_encoding

mean\_encoding(dataframe, ‘categorical\_column\_name’, ‘target\_column\_name’)



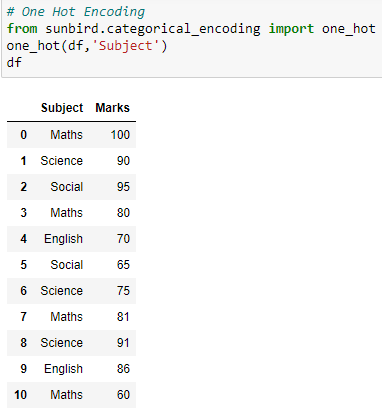
## One Hot Encoding:

In One Hot Encoding, new dummy columns are added with 0’s and 1’s denoting the absence and presence of the category. The number of dummy columns depends on the number of categories.

Syntax:

from sunbird.categorical\_encoding import one\_hot

one\_hot(dataframe, ‘categorical\_column\_name’)



Here, we can observe that, though we applied One Hot Encoding using Sunbird, there is no change in categorical columns. So, this might be due to some issues in the library.

## One Hot Encoding with Multiple Categories:

This is same as One hot encoding with dummy variable trap. But in dummy variable trap, only first column can be dropped. In One Hot Encoding with Multiple Categories, ‘k’ value decides how many columns to reside. By default, k=10. Here, in our example, we gave k=3 and observe that 4th category column is skipped.

Syntax:

from sunbird.categorical\_encoding import kdd\_cup

kdd\_cup(dataframe, ‘categorical\_column\_name’,k=10)

