Package document:

Working: The package works on the input text file , reads it in the transformation method and works on the data row by row.

Column data transformation:

1. ***Engine location transformed to one hot transformation***
2. **Cylinder count from string to integer**
3. **Aspiration to Boolean flag**
4. **Price converted from cent to euro**
5. **Converted horsepower from German representation to float**
6. **Added all the transformed data to a temporary list to have a row data in the list.**

Then the row is bifurcated to get the data column wise. It has been seperated by ‘ ; ‘.

row.split(“;”) has been used to split the data.

The dummy data has been filtered according to the specification. If any column consists of ‘\_’ or similar transformation the whole row has been left out.

Transformation reason:

* Cylinder count: Integer encoding has been used, it represent the count of the cylinders in the vehicle. The mapping is as follows.

cylinders\_count = {

**'one'**: 1,

**'two'**: 2,

**'three'**: 3,

**'four'**: 4,

**'five'**: 5,

**'six'**: 6,

**'seven'**: 7,

**'eight'**: 8

}

* Aspiration: Boolean flag used to represent turbo as 1 and std as 0.

aspiration\_flag = {

**'turbo'**: 1,

**'std'**: 0

}

* Engine location: Since the output has to be int type so, Integer transformed has been used

engine\_loc\_transform = {

**'front'**: 1,**'mid'**: 2,**'back'**: 3}

* Price:

Converted cents to Euro by dividing it by 100.

* Horsepower: Converted to decimal format from German comma format.
* Wheel base measured between rotational centers of wheels.[Float]
* stroke is the distance so measured in float.
* num-of-doors is an integer field.

no\_of\_doors = {

**'one'**:1,

**'two'**:2,

**'three'**:3,

**'four'**:4,

**'five'**:5

}

* Drive Wheel can be represented as integer field.

drive\_wheels = {**'fwd'**: 1,

**'rwd'**: 2,

**'4wd'**: 4,

**'6wd'**: 6,

**'8wd'**: 8,

**'12wd'**: 12}

* Fuel type represented as int.

fuel\_cache = {

**'gas'**: 1,

**'diesel'**: 2,

**'petrol'**: 3

}

The code contains two method :

1. full\_tranformation
2. transform

The method **full\_transformation** does transformation on all the columns.

The method **transform** des transformation on restricted columns.

Return: The transformed data is returned in list of list format where the first index of the final list has the list of columns as per the specification.