Technological Institute of the Philippines	Quezon City - Computer Engineering
Course Code:	CPE 019
Code Title:	Emerging Technologies in CpE 2
2nd Semester	AY 2024 - 2025

ASSIGNMENT 7.1	Classifications Regression
Name	Calvadores, Kelly Joseph
Section	CPE32S3
Date Performed:	March 8, 2024
Date Submitted:	April 11, 2024
Instructor:	Engr. Roman M. Richard

Classifications

Explain your datasets and the problem being addressed.

The Dataset is about Hepatitis C Virus, that contians about information of patients that is infected with the virus. The task is to predict the development of the illness or the probability of the outcomes based on the patients data or information and medical history.

```
!pip install scikeras
```

```
Requirement already satisfied: scikeras in /usr/local/lib/python3.10/dist-packages (0.12.0)
Requirement already satisfied: packaging>=0.21 in /usr/local/lib/python3.10/dist-packages (from scikeras) (24.0)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from scikeras) (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.25.2
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.11.4
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.3.2
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras)
```

```
!pip install np_utils
```

```
Requirement already satisfied: np_utils in /usr/local/lib/python3.10/dist-packages (0.6.0)
Requirement already satisfied: numpy>=1.0 in /usr/local/lib/python3.10/dist-packages (from np_utils) (1.25.2)
```

```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from scikeras.wrappers import KerasClassifier
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
```

Load Data

Link: *https://archive.ics.uci.edu/dataset/571/hcv+data*

```
C_Dataframe = pd.read_csv("hcv+data_Classification.csv")
C_Dataframe
```

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA
0	1	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0
1	2	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0
2	3	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0
3	4	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0
4	5	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0
...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	3=Cirrhosis	62	f	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7
611	612	3=Cirrhosis	64	f	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0

Remarks:

As seen in this table that there are some data in this dataset are missing and also there are also some data types that need to change, in this dataset the target variable is Category.

```
C_Dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  615 non-null    int64  
 1   Category    615 non-null    object 
 2   Age         615 non-null    int64  
 3   Sex         615 non-null    object 
 4   ALB         614 non-null    float64
 5   ALP         597 non-null    float64
 6   ALT         614 non-null    float64
 7   AST         615 non-null    float64
 8   BIL         615 non-null    float64
 9   CHE         615 non-null    float64
 10  CHOL        605 non-null    float64
 11  CREA        615 non-null    float64
 12  GGT         615 non-null    float64
 13  PROT        614 non-null    float64
dtypes: float64(10), int64(2), object(2)
memory usage: 67.4+ KB
```

```
LE = LabelEncoder()
for i in C_Dataframe:
  if C_Dataframe[i].dtypes == 'object':
    C_Dataframe[i] = LE.fit_transform(C_Dataframe[i])
  else:
    pass
C_Dataframe
```

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0	32	1	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0	32	1	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0	32	1	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0	32	1	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0	32	1	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	4	62	0	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7	650.9	68.5
611	612	4	64	0	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0	35.9	71.3
612	613	4	64	0	29.0	87.3	3.5	99.0	48.0	1.66	3.63	66.7	64.2	82.0
613	614	4	46	0	33.0	NaN	39.0	62.0	20.0	3.56	4.20	52.0	50.0	71.0
614	615	4	59	0	36.0	NaN	100.0	80.0	12.0	9.07	5.30	67.0	34.0	68.0

615 rows × 14 columns

```
C_Dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  615 non-null    int64  
 1   Category    615 non-null    int64  
 2   Age         615 non-null    int64  
 3   Sex         615 non-null    int64  
 4   ALB         614 non-null    float64
 5   ALP         597 non-null    float64
 6   ALT         614 non-null    float64
 7   AST         615 non-null    float64
 8   BIL         615 non-null    float64
 9   CHE         615 non-null    float64
 10  CHOL        605 non-null    float64
 11  CREA        615 non-null    float64
 12  GGT         615 non-null    float64
 13  PROT        614 non-null    float64
dtypes: float64(10), int64(4)
memory usage: 67.4 KB
```

```
C_Dataframe = C_Dataframe.fillna(0)
```

```
C_Dataframe
```

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT	PROT
0	1	0	32	1	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1	69.0
1	2	0	32	1	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6	76.5
2	3	0	32	1	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2	79.3
3	4	0	32	1	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8	75.7
4	5	0	32	1	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9	68.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	4	62	0	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7	650.9	68.5
611	612	4	64	0	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0	35.9	71.3
612	613	4	64	0	29.0	87.3	3.5	99.0	48.0	1.66	3.63	66.7	64.2	82.0
613	614	4	46	0	33.0	0.0	39.0	62.0	20.0	3.56	4.20	52.0	50.0	71.0
614	615	4	59	0	36.0	0.0	100.0	80.0	12.0	9.07	5.30	67.0	34.0	68.0

615 rows × 14 columns

Removing Outliers or columns that is not correlated to target variable

```
C_Dataframe.corr()
```

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA
Unnamed: 0	1.000000	0.568466	0.420477	-0.598597	-0.315866	-0.088372	-0.037345	0.332626	0.181459	-0.270549	-0.119960	-0.025987
Category	0.568466	1.000000	0.106341	0.060657	-0.312823	-0.162111	0.103473	0.648341	0.473006	-0.329472	-0.302038	0.182040
Age	0.420477	0.106341	1.000000	-0.024544	-0.208897	0.142408	-0.004309	0.088666	0.032492	-0.075093	0.093556	-0.022296
Sex	-0.598597	0.060657	-0.024544	1.000000	0.131317	-0.025669	0.160177	0.130891	0.111177	0.169111	0.017186	0.159589
ALB	-0.315866	-0.312823	-0.208897	0.131317	1.000000	-0.071028	-0.004611	-0.192259	-0.213765	0.364480	0.209717	0.001046
ALP	-0.088372	-0.162111	0.142408	-0.025669	-0.071028	1.000000	-0.029767	-0.019753	-0.009932	0.053445	0.156822	0.145103
ALT	-0.037345	0.103473	-0.004309	0.160177	-0.004611	-0.029767	1.000000	0.273140	-0.038022	0.145663	0.049990	-0.042857
AST	0.332626	0.648341	0.088666	0.130891	-0.192259	-0.019753	0.273140	1.000000	0.312231	-0.208536	-0.207026	-0.021387
BIL	0.181459	0.473006	0.032492	0.111177	-0.213765	-0.009932	-0.038022	0.312231	1.000000	-0.333172	-0.202870	0.031224
CHE	-0.270549	-0.329472	-0.075093	0.169111	0.364480	0.053445	0.145663	-0.208536	-0.333172	1.000000	0.363859	-0.011157
CHOL	-0.119960	-0.302038	0.093556	0.017186	0.209717	0.156822	0.049990	-0.207026	-0.202870	0.363859	1.000000	-0.025413
CREA	-0.025987	0.182040	-0.022296	0.159589	0.001046	0.145103	-0.042857	-0.021387	0.031224	-0.011157	-0.025413	1.000000
GGT	0.247781	0.471164	0.153087	0.133276	-0.163287	0.337397	0.248520	0.491263	0.217024	-0.110345	-0.029935	0.121003

```
CorrMatr= C_Dataframe.corr()
TargCorr = CorrMatr['Category']
AbsTarCor = TargCorr.abs()
LowCorrFeat = AbsTarCor[AbsTarCor <= 0.1].index.tolist()
print(LowCorrFeat)
```

```
['Sex', 'PROT']
```

```
C_Dataframe = C_Dataframe.drop(columns = LowCorrFeat)
C_Dataframe
```

	Unnamed: 0	Category	Age	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	GGT
0	1	0	32	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	12.1
1	2	0	32	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	15.6
2	3	0	32	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	33.2
3	4	0	32	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	33.8
4	5	0	32	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	29.9
...	...	...	...	...	...	...	...	...	...	...	...	...
610	611	4	62	32.0	416.6	5.9	110.3	50.0	5.57	6.30	55.7	650.9
611	612	4	64	24.0	102.8	2.9	44.4	20.0	1.54	3.02	63.0	35.9
612	613	4	64	29.0	87.3	3.5	99.0	48.0	1.66	3.63	66.7	64.2
613	614	4	46	33.0	0.0	39.0	62.0	20.0	3.56	4.20	52.0	50.0
614	615	4	59	36.0	0.0	100.0	80.0	12.0	9.07	5.30	67.0	34.0

615 rows × 12 columns

Remarks:

In the code above as seen, the code is resposible for finding and removing variables that are not correlated to the target variable.

```
C_Dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  615 non-null    int64  
 1   Category    615 non-null    int64  
 2   Age         615 non-null    int64  
 3   ALB         615 non-null    float64
 4   ALP         615 non-null    float64
 5   ALT         615 non-null    float64
 6   AST         615 non-null    float64
 7   BIL         615 non-null    float64
 8   CHE         615 non-null    float64
 9   CHOL        615 non-null    float64
 10  CREA        615 non-null    float64
 11  GGT         615 non-null    float64
dtypes: float64(9), int64(3)
memory usage: 57.8 KB
```

```
CX = C_Dataframe.drop(columns = 'Category')
CY = C_Dataframe['Category']
```

Standard the Data

```
Standard = StandardScaler()
CXnorm = Standard.fit_transform(CX)
```

Splitting training and test

```
CX_train, CX_test, Cy_train, Cy_test = train_test_split(CXnorm, CY, test_size = 0.001, random_state = 123)
```

```
LE.fit(Cy_train)
LEY = LE.transform(Cy_train)
Cdummy_y = to_categorical(LEY)
```

1.Create a base model

Creating Model

```
def CLarge_Model():
  Model = Sequential()
  Model.add(Dense(11, input_dim = 11, activation = 'relu'))

  Model.add(Dense(5, activation = 'softmax'))

  Model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
  return Model;
```

2.Evaluate the model with k-fold cross validation

Training Model

```
tf.random.set_seed(1241)
Estimator = KerasClassifier(build_fn = CLarge_Model, epochs = 2000, batch_size = 50000, verbose = 0)
FoldK = KFold(n_splits = 10, shuffle = True)
Results = cross_val_score(Estimator, CX_train, Cdummy_y, cv = FoldK)
print("Baseline: %.2f%% (%.2f%%)" % (Results.mean()*100, Results.std()*100))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
Baseline: 93.33% (1.82%)
```

3. Improve the accuracy of your model by applying additional hidden layers

```
def CLarge_Model():
  Model = Sequential()
  Model.add(Dense(11, input_dim = 11, activation = 'relu'))
  Model.add(Dense(9, activation = 'relu'))
  Model.add(Dense(7, activation = 'relu'))

  Model.add(Dense(5, activation = 'softmax'))

  Model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
  return Model;
```

```
tf.random.set_seed(1241)
Estimator = KerasClassifier(build_fn = CLarge_Model, epochs = 2000, batch_size = 50000, verbose = 0)
FoldK = KFold(n_splits = 10, shuffle = True)
Results = cross_val_score(Estimator, CX_train, Cdummy_y, cv = FoldK)
print("Baseline: %.2f%% (%.2f%%)" % (Results.mean()*100, Results.std()*100))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
```

```
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
Baseline: 94.13% (2.67%)
```

Remarks:

As seen in the Results, the percentage of the average is 94.47%, while the standard is 2.07%, this is percentage or values might change, it might be high or low but the lowest it can go is 92% and the highest is 94%. The process is fluctuating, there might still noise coming from the dataset.

Regression

Explain your datasets and the problem being addressed.

The Steel Industry Dataset is a manufactorer that comes from DAEWOO Steel Co. Ltd in Gwangyang, South Korea. This company mainly produces coils, steel plates, and iron plates. The problem being addressed with the Steel Industry is involves with analysis or prediction about consumption of the electricity of the Steel Industry

```
!pip install scikeras
```

```
Collecting scikeras
  Downloading scikeras-0.12.0-py3-none-any.whl (27 kB)
Requirement already satisfied: packaging>=0.21 in /usr/local/lib/python3.10/dist-packages (from scikeras) (24.0)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from scikeras) (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.25.2
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.11.4
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras) (1.3.2
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->scikeras)
Installing collected packages: scikeras
Successfully installed scikeras-0.12.0
```

```
!pip install np_utils
```

```
Collecting np_utils
  Downloading np_utils-0.6.0.tar.gz (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.0/62.0 kB 1.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.0 in /usr/local/lib/python3.10/dist-packages (from np_utils) (1.25.2)
Building wheels for collected packages: np_utils
  Building wheel for np_utils (setup.py) ... done
  Created wheel for np_utils: filename=np_utils-0.6.0-py3-none-any.whl size=56441 sha256=6a6eaf35ff348e6cf61e6740b58a77ef4d6d703a32
  Stored in directory: /root/.cache/pip/wheels/b6/c7/50/2307607f44366dd021209f660045f8d51cb976514d30be7cc7
Successfully built np_utils
Installing collected packages: np_utils
Successfully installed np_utils-0.6.0
```

```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from scikeras.wrappers import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
```

Load the data

Link:https://archive.ics.uci.edu/dataset/851/steel+industry+energy+consumption

```
R_Dataframe = pd.read_csv('Steel_industry_Regression.csv')
R_Dataframe
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_R
0	01/01/2018 00:15	3.17	2.95	
1	01/01/2018 00:30	4.00	4.46	
2	01/01/2018 00:45	3.24	3.28	
3	01/01/2018 01:00	3.31	3.56	
4	01/01/2018 01:15	3.82	4.50	
...	...	...	...	
35035	31/12/2018 23:00	3.85	4.86	
35036	31/12/2018 23:15	3.74	3.74	
35037	31/12/2018 23:30	3.78	3.17	
35038	31/12/2018 23:45	3.78	3.06	
35039	31/12/2018 00:00	3.67	3.02	

35040 rows × 11 columns

Preprocess the dataset

```
R_Dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35040 entries, 0 to 35039
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   date                                  35040 non-null  object
 1   Usage_kWh                             35040 non-null  float64
 2   Lagging_Current_Reactive.Power_kVarh  35040 non-null  float64
 3   Leading_Current_Reactive_Power_kVarh  35040 non-null  float64
 4   CO2(tCO2)                             35040 non-null  float64
 5   Lagging_Current_Power_Factor          35040 non-null  float64
 6   Leading_Current_Power_Factor          35040 non-null  float64
 7   NSM                                   35040 non-null  int64
 8   WeekStatus                            35040 non-null  object
 9   Day_of_week                           35040 non-null  object
 10  Load_Type                             35040 non-null  object
dtypes: float64(6), int64(1), object(4)
memory usage: 2.9+ MB
```

```
R_Dataframe.isnull()
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reacti
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	
...	...	...	...	
35035	False	False	False	
35036	False	False	False	
35037	False	False	False	
35038	False	False	False	
35039	False	False	False	

35040 rows × 11 columns

```
RLE = LabelEncoder()
for i in R_Dataframe:
  if R_Dataframe[i].dtypes == 'object':
    R_Dataframe[i] = RLE.fit_transform(R_Dataframe[i])
  else:
    pass
R_Dataframe
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_React
0	1	3.17	2.95	
1	2	4.00	4.46	
2	3	3.24	3.28	
3	4	3.31	3.56	
4	5	3.82	4.50	
...	...	...	...	
35035	35036	3.85	4.86	
35036	35037	3.74	3.74	
35037	35038	3.78	3.17	
35038	35039	3.78	3.06	
35039	34944	3.67	3.02	

35040 rows × 11 columns

```
R_Dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35040 entries, 0 to 35039
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   date                                  35040 non-null  int64
 1   Usage_kWh                             35040 non-null  float64
 2   Lagging_Current_Reactive.Power_kVarh  35040 non-null  float64
 3   Leading_Current_Reactive_Power_kVarh  35040 non-null  float64
 4   CO2(tCO2)                             35040 non-null  float64
 5   Lagging_Current_Power_Factor          35040 non-null  float64
 6   Leading_Current_Power_Factor          35040 non-null  float64
 7   NSM                                   35040 non-null  int64
 8   WeekStatus                            35040 non-null  int64
 9   Day_of_week                           35040 non-null  int64
 10  Load_Type                             35040 non-null  int64
dtypes: float64(6), int64(5)
memory usage: 2.9 MB
```

```
R_Dataframe.corr()
```

	date	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Power_kVarh
date	1.000000	-0.011093	-0.013170	-0.013660
Usage_kWh	-0.011093	1.000000	0.896150	-0.324922
Lagging_Current_Reactive.Power_kVarh	-0.013170	0.896150	1.000000	-0.405142
Leading_Current_Reactive_Power_kVarh	-0.013660	-0.324922	-0.405142	1.000000
CO2(tCO2)	-0.003346	0.988180	0.886948	-0.332777
Lagging_Current_Power_Factor	0.024473	0.385960	0.144534	0.526770
Leading_Current_Power_Factor	0.005546	0.353566	0.407716	-0.944039
NSM	0.002740	0.234610	0.082662	0.371605
WeekStatus	-0.008929	-0.295475	-0.319870	0.260431
Day_of_week	0.007892	0.039865	0.043780	-0.019785
Load_Type	0.015578	0.444092	0.249674	0.223557

```
RCorrMatr= R_Dataframe.corr()
RTargCorr = RCorrMatr['Load_Type']
RAbsTarCor = RTargCorr.abs()
RLowCorrFeat = RAbsTarCor[RAbsTarCor <= 0.1].index.tolist()
print(RLowCorrFeat)
```

```
['date', 'Day_of_week']
```

```
R_Dataframe = R_Dataframe.drop(columns = RLowCorrFeat)
R_Dataframe
```

	Usage_kWh	Lagging_Current_Reactive.Power_kVarh	Leading_Current_Reactive_Pow
0	3.17	2.95	
1	4.00	4.46	
2	3.24	3.28	
3	3.31	3.56	
4	3.82	4.50	
...	...	...	
35035	3.85	4.86	
35036	3.74	3.74	
35037	3.78	3.17	
35038	3.78	3.06	
35039	3.67	3.02	

35040 rows × 9 columns

Balancing Data

```
R_Dataframe['Load_Type'].value_counts()
```

```
Load_Type
0    18072
2     9696
1     7272
Name: count, dtype: int64
```

Remarks

In this line of code, I balance my data due to the reason of the result is not getting the result that it should be, in this case, the dataset is almoset overfitting, that is why I must implement Undersample.

```
R_TheZero = R_Dataframe[(R_Dataframe.Load_Type == 0)]
R_TheOne = R_Dataframe[(R_Dataframe.Load_Type == 1)]
R_TheTwo = R_Dataframe[(R_Dataframe.Load_Type == 2)]
```

```
R_TheZeroDownside = R_TheZero.sample(len(R_TheOne), random_state = 123)
R_TheTwoDownside = R_TheTwo.sample(len(R_TheOne), random_state = 123)
```

```
R_Dataframe_Balanced = pd.concat([R_TheZeroDownside, R_TheTwoDownside, R_TheOne])
```

```
R_Dataframe_Balanced['Load_Type'].value_counts()
```

```
Load_Type
0    7272
2    7272
1    7272
Name: count, dtype: int64
```

```
R_Dataframe = R_Dataframe_Balanced
```

```
RX = R_Dataframe.drop(columns = 'Load_Type')
RY = R_Dataframe['Load_Type']
```

1.Create a base model

Creating Model

```
def RLarge_Model():
  Model = Sequential()
  Model.add(Dense(8, input_shape = (8, ), kernel_initializer = 'normal', activation = 'relu'))

  Model.add(Dense(1, kernel_initializer = 'normal'))
  Model.compile(loss = 'mean_squared_error', optimizer = 'adam')
  return Model
```

Training Model

```
tf.random.set_seed(1241)
REstimator = KerasRegressor(build_fn = RLarge_Model, epochs = 2000, batch_size = 50000,  verbose = 0)
Dolfk = KFold(n_splits = 10)
```

Evaluating the model

```
Rresult = cross_val_score(REstimator, RX, RY, cv = Dolfk, scoring = 'neg_mean_squared_error')
print("Baseline: %.2f (%.2f) MSE" % (Rresult.mean(), Rresult.std()))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff0c9951ab0> t
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
WARNING:tensorflow:6 out of the last 6 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7ff0c0f5d000> t
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
Baseline: -0.61 (0.52) MSE
```

2.Improve the model by standardizing the dataset

Standard or normalize dataset

```
Normalize = StandardScaler()
RXnorm = Normalize.fit_transform(RX)
```

Spliting training and test

```
RX_train, RX_test, Ry_train, Ry_test = train_test_split(RXnorm, RY, test_size = 0.001, random_state = 1241)
```

```
def RLarge_Model():
  Model = Sequential()
  Model.add(Dense(8, input_shape = (8, ), kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(8, kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(6, kernel_initializer = 'normal', activation = 'relu'))

  Model.add(Dense(1, kernel_initializer = 'normal'))
  Model.compile(loss = 'mean_squared_error', optimizer = 'adam')
  return Model
```

```
tf.random.set_seed(1241)
REstimator = KerasRegressor(build_fn = RLarge_Model, epochs = 2000, batch_size = 50000,  verbose = 0)
Dolfk = KFold(n_splits = 10)
```

```
Rresult = cross_val_score(REstimator, RX_train, Ry_train, cv = Dolfk, scoring = 'neg_mean_squared_error')
print("Baseline: %.2f (%.2f) MSE" % (Rresult.mean(), Rresult.std()))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
Baseline: -0.24 (0.02) MSE
```

3.Show tuning of layers and neurons (see evaluating small and larger networks)

Small Networks

```
def RLarge_Model():
  Model = Sequential()
  Model.add(Dense(6, input_shape = (8, ), kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(6, kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(4, kernel_initializer = 'normal', activation = 'relu'))

  Model.add(Dense(1, kernel_initializer = 'normal'))
  Model.compile(loss = 'mean_squared_error', optimizer = 'adam')
  return Model
```

```
tf.random.set_seed(1241)
REstimator = KerasRegressor(build_fn = RLarge_Model, epochs = 2000, batch_size = 50000,  verbose = 0)
Dolfk = KFold(n_splits = 10)
```

```
Rresult = cross_val_score(REstimator, RX_train, Ry_train, cv = Dolfk, scoring = 'neg_mean_squared_error')
print("Baseline: %.2f (%.2f) MSE" % (Rresult.mean(), Rresult.std()))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
```

```
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
Baseline: -0.30 (0.12) MSE
```

Larger Network

```
def RLarge_Model():
  Model = Sequential()
  Model.add(Dense(13, input_shape = (8, ), kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(10, kernel_initializer = 'normal', activation = 'relu'))
  Model.add(Dense(8, kernel_initializer = 'normal', activation = 'relu'))

  Model.add(Dense(1, kernel_initializer = 'normal'))
  Model.compile(loss = 'mean_squared_error', optimizer = 'adam')
  return Model


tf.random.set_seed(1241)
REstimator = KerasRegressor(build_fn = RLarge_Model, epochs = 2000, batch_size = 50000,  verbose = 0)
Dolfk = KFold(n_splits = 10)


Rresult = cross_val_score(REstimator, RX_train, Ry_train, cv = Dolfk, scoring = 'neg_mean_squared_error')
print("Baseline: %.2f (%.2f) MSE" % (Rresult.mean(), Rresult.std()))
```

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future
```