Lab 1

Aim : Design and develop an Artificial Neural Network model which can predict the cost of insurance

```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("Insurance Costs.csv")
```

```
df
```

	person_id	age	sex	bmi	children	smoker	region	total_charges
0	1	19.0	female	27.900	0	yes	southwest	16884.9240
1	2	18.0	male	33.770	1	no	southeast	1725.5522
2	3	28.0	male	33.000	3	no	southeast	4449.4620
3	4	33.0	male	22.705	0	no	northwest	21984.4700
4	5	32.0	male	28.880	0	no	northwest	3866.8552
...	...	...	...	...	...	...	...	...
1333	1334	50.0	male	30.970	3	no	northwest	10600.5480
1334	1335	18.0	female	31.920	0	no	northeast	2205.9807
1335	1336	18.0	female	36.850	0	no	southeast	1629.8335
1336	1337	21.0	female	25.800	0	no	southwest	2007.9450
1337	1338	61.0	female	29.070	0	yes	northwest	29141.3600

1338 rows × 8 columns

```
#!pip install pandas-profiling
```

```
#pip install -U ydata-profiling
```

EDA

```
#! pip install https://github.com/pandas-profiling/pandas-profiling/archive/master.zip
```

```
# from pandas_profiling import ProfileReport
'''from ydata_profiling import ProfileReport
ProfileReport(df)'''
```

```
'from ydata_profiling import ProfileReport\nProfileReport(df)'
```

```
df.dropna(subset=["age","bmi","smoker","region"],inplace = True)
df.drop_duplicates(inplace=True)
```

```
print(df)
```

```
      person_id   age     sex     bmi  children smoker     region  \
0             1  19.0  female  27.900         0    yes  southwest
1             2  18.0    male  33.770         1     no  southeast
2             3  28.0    male  33.000         3     no  southeast
3             4  33.0    male  22.705         0     no  northwest
4             5  32.0    male  28.880         0     no  northwest
...         ...   ...     ...     ...       ...    ...        ...
1333       1334  50.0    male  30.970         3     no  northwest
1334       1335  18.0  female  31.920         0     no  northeast
1335       1336  18.0  female  36.850         0     no  southeast
1336       1337  21.0  female  25.800         0     no  southwest
1337       1338  61.0  female  29.070         0    yes  northwest

      total_charges
0        16884.9240
1         1725.5522
2         4449.4620
3        21984.4700
4         3866.8552
...             ...
1333     10600.5480
1334      2205.9807
1335      1629.8335
1336      2007.9450
1337     29141.3600

[1328 rows x 8 columns]
```

Encoding

```
df['sex'].unique()
```

```
array(['female', 'male'], dtype=object)
```

```
df['sex'] = df['sex'].replace('male',0)
df['sex'] = df['sex'].replace('female',1)
```

```
df['smoker'].unique()
```

```
array(['yes', 'no'], dtype=object)
```

```
df['smoker'] = df['smoker'].replace('no',0)
df['smoker'] = df['smoker'].replace('yes',1)
```

```
df['region'].unique()
```

```
array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

```
df['region'] = df['region'].replace('southwest',0)
df['region'] = df['region'].replace('southeast',1)
df['region'] = df['region'].replace('northwest',2)
df['region'] = df['region'].replace('northeast',3)
```

Train Test data

```
# Selecting only 'age' and 'smoker' columns from the DataFrame
x=df.drop(columns='total_charges')
y=df.total_charges

# Split the data into train and test
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=0)
```

```
#Standardise data
from sklearn.preprocessing import StandardScaler
st=StandardScaler()
st.fit(x_train)
st.fit(x_test)

x_train_std=st.fit_transform(x_train)
x_test_std=st.fit_transform(x_test)
```

```
x_train_std.shape
```

```
(996, 7)
```

```
!pip install livelossplot
```

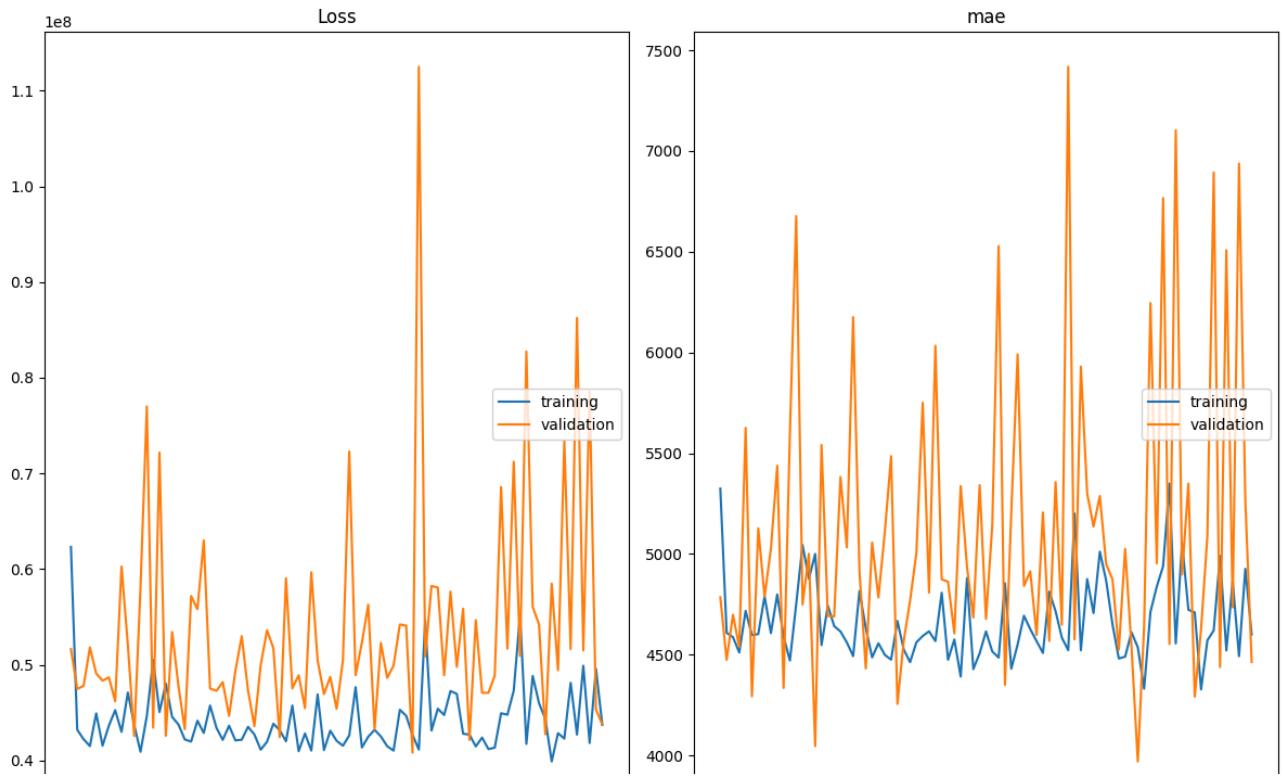
```
Collecting livelossplot
  Downloading livelossplot-0.5.5-py3-none-any.whl (22 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: bokeh in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: numpy>=1.16 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.10/d
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/pytho
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/
```

```
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-pac
Installing collected packages: livelossplot
Successfully installed livelossplot-0.5.5
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense
from livelossplot import PlotLossesKerasTF
```

Design Perceptron

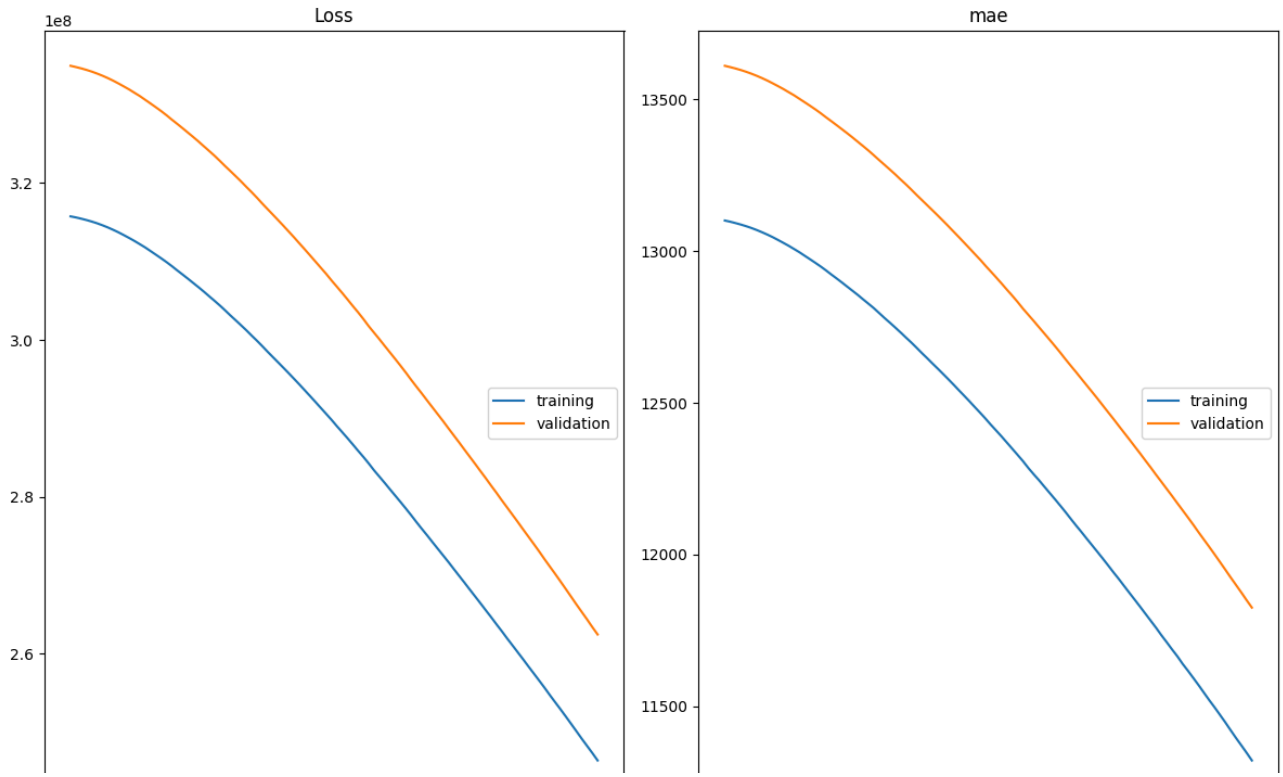
```
# Perceptron
model=Sequential()

# Add input and hidden Layer
model.add(Input(shape=(7,),name="Input Layer"))
model.add(Dense(1,activation='relu',name='OutputLayer'))

# compile the model
opt = tf.keras.optimizers.SGD(learning_rate=0.1,momentum=0.9,nesterov=True)  # momentum=
model.compile(loss='mse',optimizer=opt,metrics=['mae'])

# train the model
model.fit(x_train_std,y_train,epochs=85,validation_data=(x_test_std,y_test),callbacks=[P
```



Design Deep Neural Network

```
'''model1 = Sequential()

model1.add(Input(shape=(7,),name="Input Layer"))
model1.add(Dense(14, activation='relu', name="Hidden1"))
model1.add(Dense(21, activation='relu', name="Hidden2"))

model1.add(Dense(1, activation='relu', name='OutputLayer'))

# Compile the model
opt = tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9)
model1.compile(loss='mse', optimizer=opt, metrics=['mae'])


# Train the model
model1.fit(x_train_std,y_train,epochs=85,validation_data=(x_test_std,y_test),callbacks=



model1 = Sequential()

model1.add(Input(shape=(7,),name="Input Layer"))
model1.add(Dense(14, activation='relu', name="Hidden1"))
model1.add(Dense(21, activation='relu', name="Hidden2"))

model1.add(Dense(1, activation='relu', name='OutputLayer'))

# Compile the model
opt = tf.keras.optimizers.Adagrad(learning_rate=0.01)
model1.compile(loss='mse', optimizer=opt, metrics=['mae'])


# Train the model
```

```
model1.fit(x_train_std,y_train,epochs=100,validation_data=(x_test_std,y_test),callbacks
```




```
y_test
```

```
1296     1708.9258
455     21797.0000
468     23288.9280
261     17085.2680
1260     4544.2350
           ...
1327     9377.9040
1285     8534.6720
589      5976.8310
1025     2020.1770
141      3490.5490
Name: total_charges, Length: 332, dtype: float64
```

```
# save model
model1.save("insurance.h5")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: Us
  saving_api.save_model(
```