```
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.models import Sequential


main_df = pd.read_csv('CarData.csv')
main_df
```

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	driv
0	1	3	alfa-romero giulia	gas	std	two	convertible	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	
3	4	2	audi 100 ls	gas	std	four	sedan	
4	5	2	audi 100ls	gas	std	four	sedan	
...	...	...	...	...	...	...	...	
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	
201	202	-1	volvo 144ea	gas	turbo	four	sedan	
202	203	-1	volvo 244dl	gas	std	four	sedan	
203	204	-1	volvo 246	diesel	turbo	four	sedan	
204	205	-1	volvo 264gl	gas	turbo	four	sedan	

205 rows × 26 columns

```
main_df.drop(columns=['CarName'], inplace=True)
main_df
```

	car_ID	symboling	fueltype	aspiration	doornumber	carbody	drivewheel	engi
0	1	3	gas	std	two	convertible	rwd	
1	2	3	gas	std	two	convertible	rwd	
2	3	1	gas	std	two	hatchback	rwd	
3	4	2	gas	std	four	sedan	fwd	
4	5	2	gas	std	four	sedan	4wd	
...	...	...	...	...	...	...	...	
200	201	-1	gas	std	four	sedan	rwd	
201	202	-1	gas	turbo	four	sedan	rwd	
202	203	-1	gas	std	four	sedan	rwd	

```
#convert columns using one-hot encoding
encoding_columns = ['symboling', 'fueltype', 'aspiration', 'doornumber', 'carbody', 'drive
main_array = np.array(main_df.car_ID).reshape(-1, 1)
for column in main_df.columns:
    if column in encoding_columns:
        temp = np.array(pd.get_dummies(main_df[column]))  # If column is a categorical, pe
    else:
        temp = np.array(main_df[column]).reshape(-1, 1)
    main_array = np.hstack((main_array, temp))  # concantate the columns
main_array = main_array[:, 2:]  # Remove car_ID column
pd.DataFrame(main_array)  # Display array
```

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53
0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.47	2.68	9.0	111.0
1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.47	2.68	9.0	111.0
2	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	2.68	3.47	9.0	154.0
3	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.19	3.40	10.0	102.0
4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.19	3.40	8.0	115.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.78	3.15	9.5	114.0
201	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	3.78	3.15	8.7	160.0
202	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	3.58	2.87	8.8	134.0
203	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	...	0.0	0.0	3.01	3.40	23.0	106.0
204	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	3.78	3.15	9.5	114.0

205 rows × 58 columns

```
X_data = main_array[:, :-1]
y_data = main_array[:, -1].reshape(-1, 1)
x_scaler = StandardScaler()
```

```
y_scaler = StandardScaler()
X_data_scaled = x_scaler.fit_transform(X_data)
y_data_scaled = y_scaler.fit_transform(y_data)
print("Shape of X_data: {}".format(X_data.shape))
print("Shape of y_data: {}".format(y_data.shape))
print("==========X_data after rescaling===============")
print(pd.DataFrame(X_data_scaled).head())
print("==========y_data after rescaling===============")
print(y_data_scaled.ravel())
```

```
Shape of X_data: (205, 57)
Shape of y_data: (205, 1)
==========X_data after rescaling===============
          0         1         2         3         4         5         6   \
0 -0.121867 -0.346726 -0.696784 -0.598010 -0.430083  2.567604 -0.328798
1 -0.121867 -0.346726 -0.696784 -0.598010 -0.430083  2.567604 -0.328798
2 -0.121867 -0.346726 -0.696784  1.672213 -0.430083 -0.389468 -0.328798
3 -0.121867 -0.346726 -0.696784 -0.598010  2.325134 -0.389468 -0.328798
4 -0.121867 -0.346726 -0.696784 -0.598010  2.325134 -0.389468 -0.328798

          7         8         9   ...        47        48        49        50  \
0  0.328798  0.469295 -0.469295  ...  1.08667 -0.214286 -0.070014  0.519071
1  0.328798  0.469295 -0.469295  ...  1.08667 -0.214286 -0.070014  0.519071
2  0.328798  0.469295 -0.469295  ...  1.08667 -0.214286 -0.070014 -2.404880
3  0.328798  0.469295 -0.469295  ...  1.08667 -0.214286 -0.070014 -0.517266
4  0.328798  0.469295 -0.469295  ...  1.08667 -0.214286 -0.070014 -0.517266

          51        52        53        54        55        56
0 -1.839377 -0.288349  0.174483 -0.262960 -0.646553 -0.546059
1 -1.839377 -0.288349  0.174483 -0.262960 -0.646553 -0.546059
2  0.685946 -0.288349  1.264536 -0.262960 -0.953012 -0.691627
3  0.462183 -0.035973 -0.053668  0.787855 -0.186865 -0.109354
4  0.462183 -0.540725  0.275883  0.787855 -1.106241 -1.273900

[5 rows x 57 columns]
==========y_data after rescaling===============
[ 2.73911432e-02  4.04461099e-01  4.04461099e-01  8.44849301e-02
  5.23667906e-01  2.47610036e-01  5.56292928e-01  7.08124756e-01
  1.32988237e+00  5.75010530e-01  3.95677439e-01  4.57790460e-01
  9.65360500e-01  9.82300415e-01  1.41646416e+00  2.19381802e+00
  3.51826840e+00  2.96176083e+00 -1.01962107e+00 -8.76070979e-01
 -8.40936341e-01 -9.66793634e-01 -8.65781550e-01 -6.67521806e-01
 -8.84352716e-01 -8.26255082e-01 -7.11189142e-01 -5.92107815e-01
 -5.46558266e-01 -3.92391883e-02 -8.52982503e-01 -8.05801703e-01
 -9.88501821e-01 -8.46708460e-01 -7.71419950e-01 -7.50590129e-01
 -7.50590129e-01 -6.75301619e-01 -5.24724598e-01 -5.56094811e-01
 -3.74147578e-01 -4.16233245e-02 -3.67873535e-01 -8.14585363e-01
 -5.47122930e-01 -5.47122930e-01 -2.79660498e-01  2.38078449e+00
  2.79487130e+00  2.85133768e+00 -1.01409991e+00 -9.01167150e-01
 -8.13330554e-01 -8.25878639e-01 -7.38042044e-01 -2.92585025e-01
 -1.79652260e-01  4.62132708e-02  2.97174972e-01 -5.56094811e-01
 -6.00013109e-01 -3.36503323e-01 -3.80421620e-01 -3.11407153e-01
 -2.54940770e-01  6.27817012e-01  6.35847787e-01  1.54031376e+00
  1.87861013e+00  1.86957551e+00  2.29922194e+00  2.62346446e+00
  2.73288376e+00  3.47372270e+00  4.03085767e+00  4.04837541e-01
 -9.89756630e-01 -8.89371950e-01 -8.29141141e-01 -7.01150674e-01
 -4.16309144e-01 -5.99511185e-01 -8.12752732e-02  1.99801832e-01
  1.52119108e-01 -7.88987269e-01 -6.38410249e-01 -5.01636122e-01
 -5.01636122e-01 -9.75953736e-01 -7.75184376e-01 -8.31650758e-01
```

```
  -8.06554588e-01 -7.43814163e-01 -7.50088206e-01 -6.87347781e-01
  -7.24992036e-01 -6.62251610e-01 -6.30881398e-01 -5.43044803e-01
  -4.67756292e-01  2.78930666e-02  1.40825832e-01  2.78930666e-02
   4.92172213e-01  8.05874339e-01  6.42749233e-01 -1.72750813e-01
  -9.62570765e-03 -1.04991154e-01  7.31916536e-02  2.89018716e-01
   4.54653439e-01  4.28929864e-01  4.76612588e-01  4.20773609e-01
   5.86408332e-01  6.11504502e-01 -9.66793634e-01 -6.67521806e-01
  -8.84352716e-01 -8.26255082e-01 -7.11189142e-01 -5.46558266e-01
  -6.43353584e-02  1.09686443e+00  2.41566817e+00  2.60388944e+00
```

```
X_train, X_test, y_train, y_test = train_test_split(X_data_scaled, y_data_scaled, test_siz
print("Shape of X_train: {}".format(X_train.shape))
print("Shape of X_test: {}".format(X_test.shape))
print("Shape of y_train: {}".format(y_train.shape))
print("Shape of y_test: {}".format(y_test.shape))
```

```
Shape of X_train: (184, 57)
Shape of X_test: (21, 57)
Shape of y_train: (184, 1)
Shape of y_test: (21, 1)
```

```
model = Sequential()
model.add(Dense(8, activation='relu', input_shape=(None, 57)))
model.add(Dense(4, activation='relu'))
model.add(Dense(1, activation='linear'))
model.compile(optimizer='adam', loss='mse', metrics=['mae'])
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, None, 8)           464

 dense_2 (Dense)             (None, None, 4)           36

 dense_3 (Dense)             (None, None, 1)           5

=================================================================
Total params: 505
Trainable params: 505
Non-trainable params: 0
_________________________________________________________________
```