Project Name : Concrete Compressive Strength Prediction using DL Auto Keras(Auto ML)

- To predict and analysis concrete compressive strength using Machine Learning techniques and auto ML

Abstract

Concrete is the most important material in civil engineering. Theconcrete compressive strength is a highly nonlinear function of age andingredients. These ingredients include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate

Data Characteristics

The actual concrete compressive strength (MPa) for a given mixture under a specific age (days) was determined from laboratory.

Time Line o fthe Project:

- Data Analysis
- Data Preprocessing
- Feature Engineering
- Model Building using DL
- Model Building using Auto Keras

Importing Libraries

```
import pandas as pd
import numpy as np
```

```
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
```

```
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv("/content/drive/MyDrive/concrete_data.csv")
df.head()
```

```
Mounted at /content/drive
```

	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age	concrete_compressive_st
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)
```

```
df.shape
```

```
(1030, 9)
```

Data Analysis

```
df.describe()
```

	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age
count	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000	1030.000000
mean	281.167864	73.895825	54.188350	181.567282	6.204660	972.918932	773.580485	45.662136
std	104.506364	86.279342	63.997004	21.354219	5.973841	77.753954	80.175980	63.169912
min	102.000000	0.000000	0.000000	121.800000	0.000000	801.000000	594.000000	1.000000
25%	192.375000	0.000000	0.000000	164.900000	0.000000	932.000000	730.950000	7.000000
50%	272.900000	22.000000	0.000000	185.000000	6.400000	968.000000	779.500000	28.000000
75%	350.000000	142.950000	118.300000	192.000000	10.200000	1029.400000	824.000000	56.000000
max	540.000000	359.400000	200.100000	247.000000	32.200000	1145.000000	992.600000	365.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   cement                        1030 non-null   float64
 1   blast_furnace_slag            1030 non-null   float64
 2   fly_ash                       1030 non-null   float64
 3   water                         1030 non-null   float64
```

```
 4   superplasticizer               1030 non-null   float64
 5   coarse_aggregate               1030 non-null   float64
 6   fine_aggregate                 1030 non-null   float64
 7   age                            1030 non-null   int64
 8   concrete_compressive_strength  1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
```

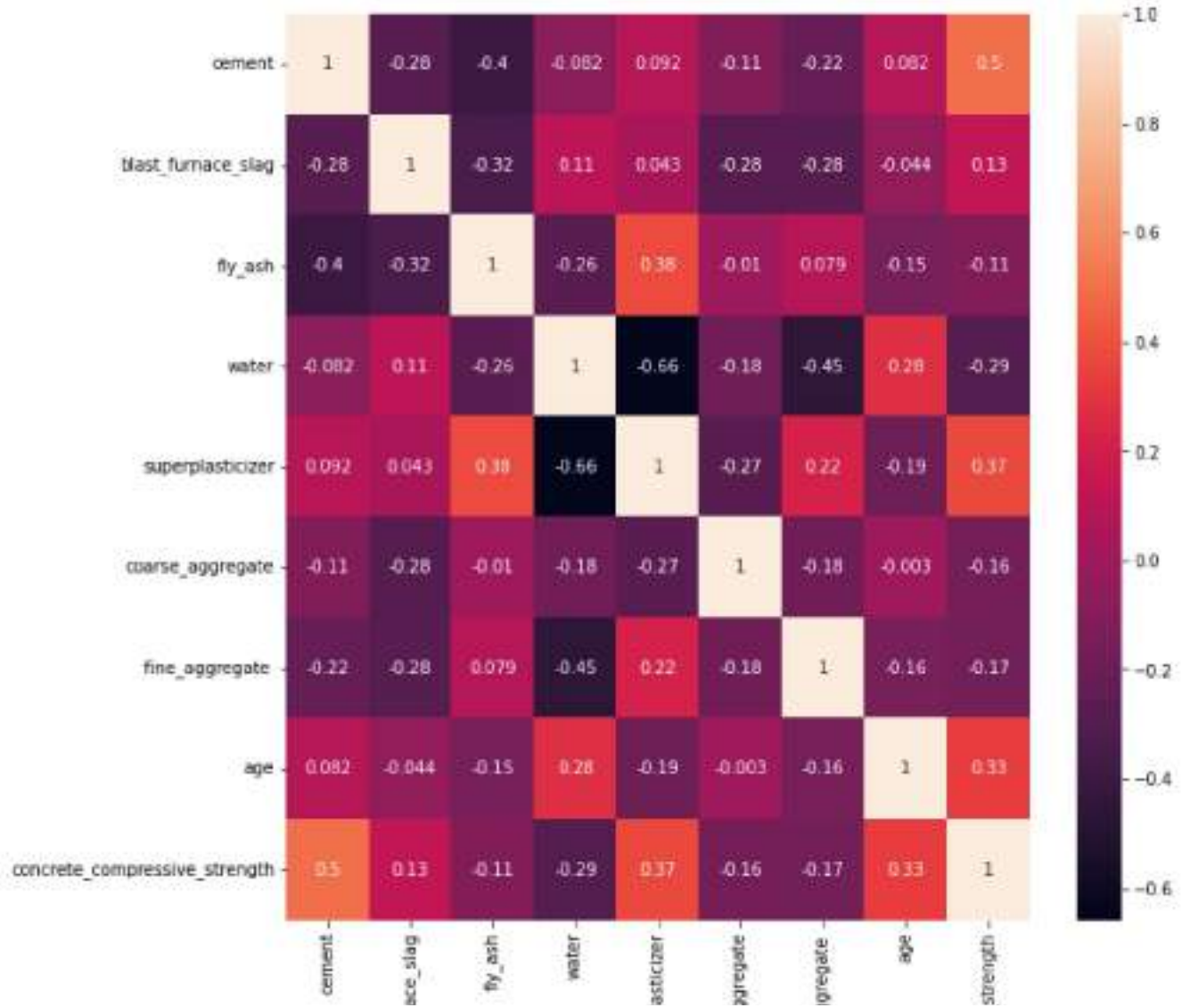
```
df.isna().sum()
```

```
cement                         0
blast_furnace_slag             0
fly_ash                        0
water                          0
superplasticizer               0
coarse_aggregate               0
fine_aggregate                 0
age                            0
concrete_compressive_strength  0
dtype: int64
```

Heatmap

```
import seaborn as sns
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5831b87510>
```

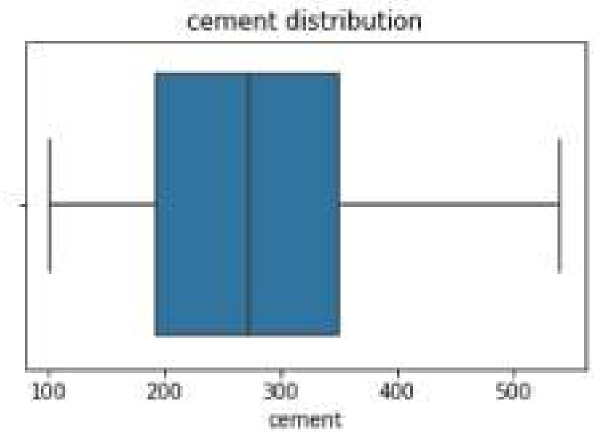


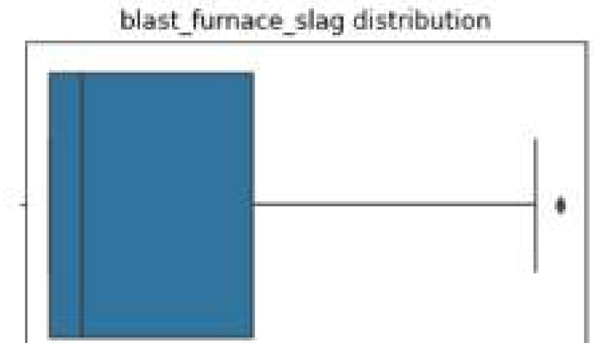

```
for i in df.columns:
    for j in df.columns:
        plt.figure(figsize=(9,7))
        sns.scatterplot(x=i,y=j,hue="concrete_compressive_strength",data=df)
        plt.show()
```

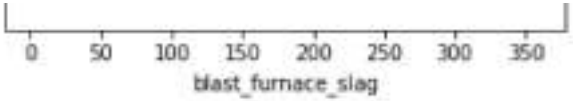
Outlier Analysis

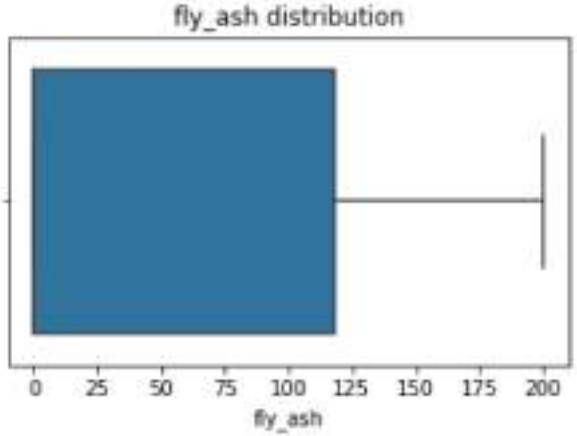
```
def outlier(data,column):
    plt.figure(figsize=(5,3))
    sns.boxplot(data[column])
    plt.title("{} distribution".format(column))


for i in df.columns:
    outlier(df,i)
```

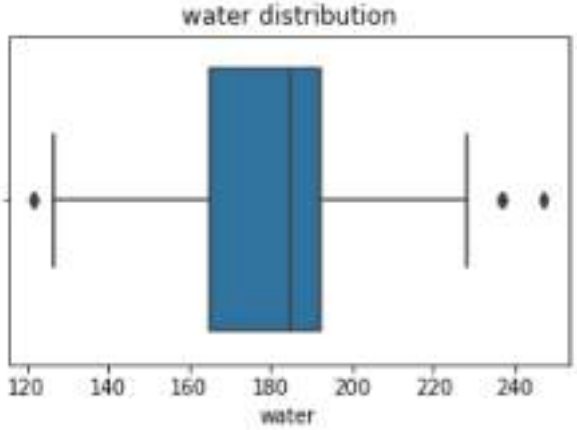
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg
  FutureWarning
```

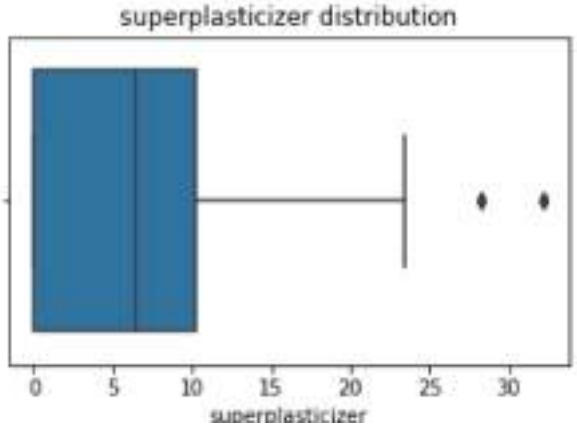


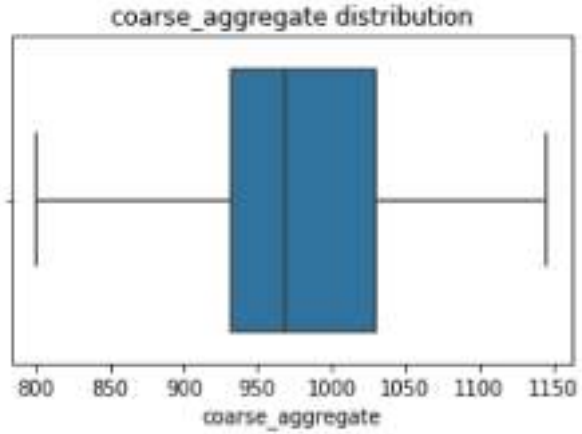


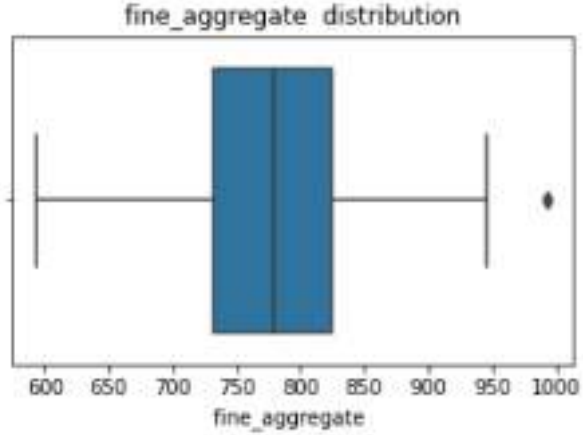

0
50
100
150
200
250
300
350
blast_furnace_slag

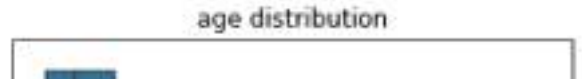

fly_ash distribution
0
25
50
75
100
125
150
175
200
fly_ash


water distribution
120
140
160
180
200
220
240
water


superplasticizer distribution
0
5
10
15
20
25
30
superplasticizer







Findind the min and max value for every feature

```
def end_value_show(data,column):
    print("min value of {} is {} \nmax value of {} is {}".format(column,data[column].min(),column,data[column].max()))
```

```
for i in df.columns:
    end_value_show(df,i)
```

```
min value of cement is 102.0
max value of cement is 540.0
min value of blast_furnace_slag is 0.0
max value of blast_furnace_slag is 359.4
min value of fly_ash is 0.0
max value of fly_ash is 200.1
min value of water is 121.8
max value of water is 247.0
min value of superplasticizer is 0.0
max value of superplasticizer is 32.2
min value of coarse_aggregate is 801.0
max value of coarse_aggregate is 1145.0
min value of fine_aggregate  is 594.0
max value of fine_aggregate  is 992.6
min value of age is 1
max value of age is 365
min value of concrete_compressive_strength is 2.33
max value of concrete_compressive_strength is 82.6
```

Replacing the Outliers

```
df=df[df["blast_furnace_slag"]<350]
df=df[(df["water"]<246) & (df["water"]>122)]
df=df[df["superplasticizer"]<25]
df=df[df["age"]<150]
```

Feature Engineering

```
df.columns
```

```
Index(['cement', 'blast_furnace_slag', 'fly_ash', 'water', 'superplasticizer',
       'coarse_aggregate', 'fine_aggregate ', 'age',
       'concrete_compressive_strength'],
      dtype='object')
```

```
df.drop(["blast_furnace_slag"],axis=1,inplace=True)
df.drop(["coarse_aggregate"],axis=1,inplace=True)
df.drop(["fine_aggregate "],axis=1,inplace=True)


df.columns

     Index(['cement', 'fly_ash', 'water', 'superplasticizer', 'age',
            'concrete_compressive_strength'],
           dtype='object')


plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f58211ba5d0>
```

Splitting the Data

```
x=df.drop(["concrete_compressive_strength"],axis=1)
y=df["concrete_compressive_strength"]
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

```
x_train.shape
```

```
(666, 5)
```

Model Building using DL

We will be using Keras Sequential Model for this project

```
from tensorflow.keras import models,layers


model=models.Sequential()
model.add(layers.Dropout(0.1))
model.add(layers.Dense(100,activation='relu',input_shape=(x_train.iloc[1].shape)))
model.add(layers.Dropout(0.7))
model.add(layers.Dense(5,activation='tanh'))
model.add(layers.Dropout(0.2))

model.add(layers.Dense(1))
model.compile(optimizer='rmsprop',loss='mse',metrics=['mae'])


model.fit(x_train,y_train,epochs=100,batch_size=1,validation_data=(x_test,y_test))
```

```
Epoch 1/100
666/666 [==============================] - 2s 2ms/step - loss: 1485.7920 - mae: 34.6102 - val_loss: 1331.3364 - val_mae: 32.3
Epoch 2/100
666/666 [==============================] - 1s 2ms/step - loss: 1305.1753 - mae: 31.8833 - val_loss: 1128.5198 - val_mae: 29.2
Epoch 3/100
666/666 [==============================] - 1s 2ms/step - loss: 1133.1284 - mae: 29.0827 - val_loss: 975.5533 - val_mae: 26.50
Epoch 4/100
666/666 [==============================] - 1s 2ms/step - loss: 972.3732 - mae: 26.3195 - val_loss: 836.2960 - val_mae: 23.829
Epoch 5/100
666/666 [==============================] - 1s 2ms/step - loss: 831.9205 - mae: 23.7815 - val_loss: 713.6224 - val_mae: 21.440
Epoch 6/100
666/666 [==============================] - 1s 2ms/step - loss: 711.1116 - mae: 21.5480 - val_loss: 611.5874 - val_mae: 19.549
Epoch 7/100
666/666 [==============================] - 1s 2ms/step - loss: 614.9681 - mae: 19.6823 - val_loss: 527.9816 - val_mae: 17.924
Epoch 8/100
666/666 [==============================] - 1s 2ms/step - loss: 542.0759 - mae: 18.4474 - val_loss: 461.3580 - val_mae: 16.525
Epoch 9/100
666/666 [==============================] - 1s 2ms/step - loss: 482.2609 - mae: 17.2462 - val_loss: 411.5935 - val_mae: 15.457
Epoch 10/100
```

```
666/666 [==============================] - 1s 2ms/step - loss: 440.0196 - mae: 16.5716 - val_loss: 373.9684 - val_mae: 14.715
Epoch 11/100
666/666 [==============================] - 1s 2ms/step - loss: 410.7732 - mae: 16.0131 - val_loss: 347.4101 - val_mae: 14.233
Epoch 12/100
666/666 [==============================] - 1s 2ms/step - loss: 386.4586 - mae: 15.3980 - val_loss: 327.3560 - val_mae: 13.892
Epoch 13/100
666/666 [==============================] - 1s 2ms/step - loss: 368.7299 - mae: 15.1724 - val_loss: 313.0540 - val_mae: 13.653
Epoch 14/100
666/666 [==============================] - 1s 2ms/step - loss: 341.2177 - mae: 14.7428 - val_loss: 302.0838 - val_mae: 13.457
Epoch 15/100
666/666 [==============================] - 1s 2ms/step - loss: 363.3705 - mae: 15.0023 - val_loss: 295.2697 - val_mae: 13.341
Epoch 16/100
666/666 [==============================] - 1s 2ms/step - loss: 353.3517 - mae: 14.7393 - val_loss: 290.4298 - val_mae: 13.254
Epoch 17/100
666/666 [==============================] - 2s 2ms/step - loss: 335.1943 - mae: 14.4431 - val_loss: 286.6155 - val_mae: 13.183
Epoch 18/100
666/666 [==============================] - 2s 3ms/step - loss: 313.7349 - mae: 14.1529 - val_loss: 283.6222 - val_mae: 13.135
Epoch 19/100
666/666 [==============================] - 1s 2ms/step - loss: 333.9993 - mae: 14.5991 - val_loss: 282.0964 - val_mae: 13.117
Epoch 20/100
666/666 [==============================] - 1s 2ms/step - loss: 328.4701 - mae: 14.4783 - val_loss: 280.7684 - val_mae: 13.099
Epoch 21/100
666/666 [==============================] - 1s 2ms/step - loss: 340.3510 - mae: 14.7521 - val_loss: 279.9156 - val_mae: 13.090
Epoch 22/100
666/666 [==============================] - 1s 2ms/step - loss: 336.9561 - mae: 14.5285 - val_loss: 279.3905 - val_mae: 13.086
Epoch 23/100
666/666 [==============================] - 1s 2ms/step - loss: 330.3562 - mae: 14.6115 - val_loss: 278.7209 - val_mae: 13.085
Epoch 24/100
666/666 [==============================] - 1s 2ms/step - loss: 333.9696 - mae: 14.5908 - val_loss: 278.2091 - val_mae: 13.087
Epoch 25/100
666/666 [==============================] - 1s 2ms/step - loss: 319.9666 - mae: 14.4245 - val_loss: 277.9817 - val_mae: 13.090
Epoch 26/100
666/666 [==============================] - 1s 2ms/step - loss: 320.1514 - mae: 14.3281 - val_loss: 277.5012 - val_mae: 13.102
Epoch 27/100
666/666 [==============================] - 1s 2ms/step - loss: 340.0629 - mae: 14.8509 - val_loss: 277.4897 - val_mae: 13.102
Epoch 28/100
666/666 [==============================] - 1s 2ms/step - loss: 329.7220 - mae: 14.6775 - val_loss: 277.4768 - val_mae: 13.103
```

```
model.evaluate(x_test,y_test)
```

```
9/9 [==============================] - 0s 3ms/step - loss: 276.4749 - mae: 13.1592
[276.4749450683594, 13.159246444702148]
```

```
pred=model.predict(x_test)
pred[4]
```

```
array([34.20076], dtype=float32)
```

Using Auto Keras

AutoKeras: An AutoML system based on Keras. It is developed by DATA Lab at Texas A&M University

Installing Auto Keras

```
!pip install git+https://github.com/keras-team/keras-tuner.git@1.0.2rc1
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting git+https://github.com/keras-team/keras-tuner.git@1.0.2rc1
```

```
  Cloning https://github.com/keras-team/keras-tuner.git (to revision 1.0.2rc1) to /tmp/pip-req-build-c5lmgiak
  Running command git clone -q https://github.com/keras-team/keras-tuner.git /tmp/pip-req-build-c5lmgiak
  Running command git checkout -q 0fb69434a132093518e0e53d40020145ae192629
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (21.3)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (0.16.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (1.21.6)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (0.8.10)
Collecting terminaltables
  Downloading terminaltables-3.1.10-py2.py3-none-any.whl (15 kB)
Collecting colorama
  Downloading colorama-0.4.5-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (4.64.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (2.23.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (1.7.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from keras-tuner==1.0.2rc1) (1.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->keras-tuner=
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner==1.0.2rc1) (2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner==1.0.2r
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner==1.0.2rc
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->keras-tuner==
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->keras-tuner==1.0.2rc1
Building wheels for collected packages: keras-tuner
  Building wheel for keras-tuner (setup.py) ... done
  Created wheel for keras-tuner: filename=keras_tuner-1.0.2rc1-py3-none-any.whl size=85445 sha256=24dfc93fbf97a8e12ccf43e7a4f6e
  Stored in directory: /tmp/pip-ephem-wheel-cache-7jiogi8s/wheels/44/e5/92/e83049ca00432aec622a4fa0200e254d88aefae9d74aa86941
Successfully built keras-tuner
Installing collected packages: terminaltables, colorama, keras-tuner
Successfully installed colorama-0.4.5 keras-tuner-1.0.2rc1 terminaltables-3.1.10
```

```
!pip install autokeras
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting autokeras
  Downloading autokeras-1.0.19-py3-none-any.whl (162 kB)
     |████████████████████████████████| 162 kB 5.0 MB/s
Requirement already satisfied: tensorflow>=2.8.0 in /usr/local/lib/python3.7/dist-packages (from autokeras) (2.8.2+zzzcolab20
Collecting keras-tuner>=1.1.0
  Downloading keras_tuner-1.1.3-py3-none-any.whl (135 kB)
```

```
     |████████████████████████████████| 135 kB 52.1 MB/s
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from autokeras) (1.3.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from autokeras) (21.3)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (from keras-tuner>=1.1.0->autokeras) (7.9.0)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.7/dist-packages (from keras-tuner>=1.1.0->autokeras) (2
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from keras-tuner>=1.1.0->autokeras) (2.23
Collecting kt-legacy
  Downloading kt_legacy-1.0.4-py3-none-any.whl (9.6 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from keras-tuner>=1.1.0->autokeras) (1.21.6)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autoker
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autoker
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (57.4
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autok
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (1.1
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (3.1
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autoke
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (
Requirement already satisfied: tensorflow-estimator<2.9,>=2.8 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflo
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->au
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (0.5
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow>=2.8.0->autokeras) (1
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0->tensorfl
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py>=2.9.0->tensorflow>=2.8.0
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner>=1.1
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard->
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorbo
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard->ker
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages (from tensorboard->keras-tuner>=1
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->t
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboa
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensc
```

```
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->go
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner>=1.1.0
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner>=1.1.0->aut
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from reques
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->keras-tuner>=1.1
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->goog
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->keras-tuner>=1.1.0->au
Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages (from ipython->keras-tuner>=1.1.0->autokera
```

```
!pip show autokeras
```

```
Name: autokeras
Version: 1.0.19
Summary: AutoML for deep learning
Home-page: http://autokeras.com
Author: DATA Lab, Keras Team
Author-email: jhfjhfj1@gmail.com
License: Apache License 2.0
Location: /usr/local/lib/python3.7/dist-packages
Requires: pandas, keras-tuner, tensorflow, packaging
Required-by:
```

```
import numpy as np
import pandas as pd
import tensorflow as tf
```

```
import autokeras as ak
```

```
reg = ak.StructuredDataRegressor(
    overwrite=True, max_trials=3
)
```

```
reg.fit(x=x_train, y=y_train, verbose=0)
```

```
Epoch 1/187
21/21 [==============================] - 1s 3ms/step - loss: 1473.0736 - mean_squared_error: 1473.0737
Epoch 2/187
21/21 [==============================] - 0s 3ms/step - loss: 1297.2740 - mean_squared_error: 1297.2740
Epoch 3/187
21/21 [==============================] - 0s 3ms/step - loss: 1003.8372 - mean_squared_error: 1003.8372
Epoch 4/187
21/21 [==============================] - 0s 2ms/step - loss: 609.1078 - mean_squared_error: 609.1078
Epoch 5/187
21/21 [==============================] - 0s 2ms/step - loss: 297.9042 - mean_squared_error: 297.9042
Epoch 6/187
21/21 [==============================] - 0s 3ms/step - loss: 209.0874 - mean_squared_error: 209.0874
Epoch 7/187
21/21 [==============================] - 0s 2ms/step - loss: 194.4424 - mean_squared_error: 194.4424
Epoch 8/187
21/21 [==============================] - 0s 3ms/step - loss: 185.9703 - mean_squared_error: 185.9703
Epoch 9/187
21/21 [==============================] - 0s 2ms/step - loss: 180.3415 - mean_squared_error: 180.3415
Epoch 10/187
21/21 [==============================] - 0s 3ms/step - loss: 175.4085 - mean_squared_error: 175.4085
Epoch 11/187
21/21 [==============================] - 0s 3ms/step - loss: 171.2221 - mean_squared_error: 171.2221
Epoch 12/187
21/21 [==============================] - 0s 2ms/step - loss: 167.6964 - mean_squared_error: 167.6964
Epoch 13/187
21/21 [==============================] - 0s 3ms/step - loss: 164.6630 - mean_squared_error: 164.6630
Epoch 14/187
21/21 [==============================] - 0s 3ms/step - loss: 161.9766 - mean_squared_error: 161.9766
Epoch 15/187
21/21 [==============================] - 0s 3ms/step - loss: 159.6227 - mean_squared_error: 159.6227
Epoch 16/187
21/21 [==============================] - 0s 2ms/step - loss: 157.5123 - mean_squared_error: 157.5123
Epoch 17/187
21/21 [==============================] - 0s 3ms/step - loss: 155.5729 - mean_squared_error: 155.5729
Epoch 18/187
21/21 [==============================] - 0s 2ms/step - loss: 153.7965 - mean_squared_error: 153.7965
Epoch 19/187
21/21 [==============================] - 0s 3ms/step - loss: 152.1306 - mean_squared_error: 152.1306
Epoch 20/187
21/21 [==============================] - 0s 3ms/step - loss: 150.5817 - mean_squared_error: 150.5817
Epoch 21/187
```

```
21/21 [==============================] - 0s 3ms/step - loss: 149.1191 - mean_squared_error: 149.1191
Epoch 22/187
21/21 [==============================] - 0s 3ms/step - loss: 147.7410 - mean_squared_error: 147.7410
Epoch 23/187
21/21 [==============================] - 0s 3ms/step - loss: 146.4451 - mean_squared_error: 146.4451
Epoch 24/187
21/21 [==============================] - 0s 2ms/step - loss: 145.2242 - mean_squared_error: 145.2242
Epoch 25/187
21/21 [==============================] - 0s 2ms/step - loss: 144.0528 - mean_squared_error: 144.0528
Epoch 26/187
21/21 [==============================] - 0s 3ms/step - loss: 142.9184 - mean_squared_error: 142.9184
Epoch 27/187
21/21 [==============================] - 0s 2ms/step - loss: 141.8251 - mean_squared_error: 141.8251
Epoch 28/187
21/21 [==============================] - 0s 2ms/step - loss: 140.7735 - mean_squared_error: 140.7735
Epoch 29/187
21/21 [==============================] - 0s 2ms/step - loss: 139.7614 - mean_squared_error: 139.7614
```

```
# evaluate the model
mae, _  = reg.evaluate(x_test, y_test, verbose=0)
#print('MAE: %.3f' % mae)
# use the model to make a prediction
yhat_test = reg.predict(x_test)

# get the best performing model
model = reg.export_model()
```

```
9/9 [==============================] - 0s 2ms/step
```

```
# summarize the loaded model
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 5)]               0
```

```
 multi_category_encoding (Mu  (None, 5)                0
 ltiCategoryEncoding)

 normalization (Normalizatio  (None, 5)                11
 n)

 dense (Dense)               (None, 256)               1536

 re_lu (ReLU)                (None, 256)               0

 dense_1 (Dense)             (None, 32)                8224

 re_lu_1 (ReLU)              (None, 32)                0

 regression_head_1 (Dense)   (None, 1)                 33

=================================================================
Total params: 9,804
Trainable params: 9,793
Non-trainable params: 11
_________________________________________________________________
```

yhat_test

```
array([[36.693623 ],
       [35.897278 ],
       [22.037338 ],
       [24.728083 ],
       [14.3319435],
       [12.296518 ],
       [51.63543  ],
       [16.158482 ],
       [35.16346  ],
       [40.318798 ],
       [14.769261 ],
       [42.425854 ],
       [49.26095  ],
       [25.563984 ],
       [33.576683 ],
       [47.16456  ],
```

```
[50.2703   ],
[37.072987 ],
[67.02445  ],
[30.300417 ],
[62.089054 ],
[16.838814 ],
[53.939297 ],
[43.691357 ],
[13.726553 ],
[30.756012 ],
[20.511211 ],
[19.667633 ],
[34.71724  ],
[ 5.436697 ],
[30.021776 ],
[22.970392 ],
[43.788544 ],
[54.67426  ],
[47.2357   ],
[70.50105  ],
[23.994823 ],
[61.55199  ],
[21.370832 ],
[36.94697  ],
[35.894295 ],
[52.36794  ],
[25.759142 ],
[56.134426 ],
[55.75851  ],
[34.935055 ],
[43.783604 ],
[19.09735  ],
[47.477283 ],
[10.179867 ],
[52.427822 ],
[52.11327  ],
[47.32485  ],
[34.412193 ],
[37.74918  ],
[43.691357 ],
[47.072247 ],
```

```
       [41.228855 ],
```

```
y_test
```

```
248    44.30
469    44.28
757    18.13
826    24.39
557    17.24
       ...
862    35.23
513    40.29
939    32.72
454    39.64
277    36.97
Name: concrete_compressive_strength, Length: 286, dtype: float64
```