In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

import tensorflow as tf
```

In [2]:
```python
df=pd.read_csv("/content/drive/MyDrive/DL/diamonds.csv")
df.head()
```

Out[2]:

	Unnamed: 0	carat	cut	color	clarity	depth	table	price	x	y	z
0	1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [3]:
```python
# Dropping the column "Unnamed: 0", which is unncessary
df.drop(columns = 'Unnamed: 0', axis = 1, inplace = True )
df.head()
```

Out[3]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [4]:
```python
df.shape
```

Out[4]:
```
(53940, 10)
```

Exploratory Data Analysis (EDA) and Preprocessing

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    53940 non-null  float64
 1   cut      53940 non-null  object
 2   color    53940 non-null  object
 3   clarity  53940 non-null  object
 4   depth    53940 non-null  float64
 5   table    53940 non-null  float64
 6   price    53940 non-null  int64
 7   x        53940 non-null  float64
 8   y        53940 non-null  float64
 9   z        53940 non-null  float64
dtypes: float64(6), int64(1), object(3)
memory usage: 4.1+ MB
```

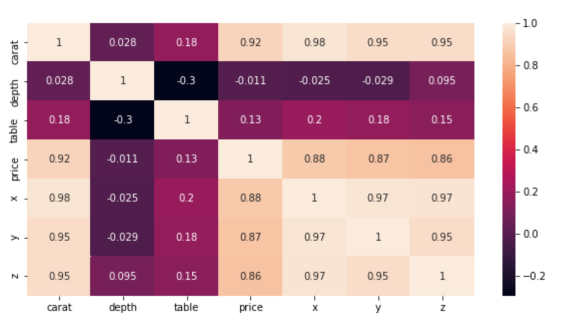
In [6]:
```python
df.describe()
```

Out[6]:

	carat	depth	table	price	x	y	z
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

In [7]:
```python
import seaborn as sns
plt.figure(figsize = (10,5))
sns.heatmap(df.corr(),annot = True)
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37a7c90090>
```



We can conclude that carat,x,y & z features has strong correlation w.r.t price variable and depth has very weak relation w.r.t price variable. Hence, we can drop depth feature from the final list of input features to the model.

In [8]:
```python
df_cat = df.select_dtypes(object)
df_num = df.select_dtypes(["float64","int64"])
#  Label encoding

from sklearn.preprocessing import LabelEncoder

for col in df_cat:
    le = LabelEncoder()
    df_cat[col] = le.fit_transform(df_cat[col])
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guid
e/indexing.html#returning-a-view-versus-a-copy
  if __name__ == '__main__':
```

In [9]:
```python
df_new = pd.concat([df_num,df_cat],axis=1)
```

In [10]:
```python
x = df_new.drop(["price","depth"],axis=1)
y = df_new["price"]
```

Dividing into training and testing data

In [11]:
```python
xtrain,xtest,ytrain,ytest=train_test_split(x,y,random_state=1,test_size=0.3)
```

In [12]:
```python
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
xtrain=ss.fit_transform(xtrain)
xtest=ss.transform(xtest)
```

In [13]:
```python
xtrain
```

Out[13]:
```
array([[-1.0011366 , -0.65177807, -1.16032405, ..., -0.54330807,
         1.41577642,  1.83637126],
       [ 1.5529171 ,  0.24415295,  1.43589605, ...,  0.43235188,
        -0.34781759, -1.06660994],
       [-0.18467681, -0.65177807, -0.03499998, ..., -0.54330807,
        -0.34781759,  0.09458254],
       ...,
       [ 0.23402051,  2.48398049,  0.28398952, ...,  1.40801183,
         0.24004708,  0.67517878],
       [ 0.79926191, -0.65177807,  0.93969016, ..., -0.54330807,
         2.00364109,  0.09458254],
       [-1.0011366 , -0.65177807, -1.16918487, ..., -0.54330807,
        -0.93568226,  1.25577502]])
```

In [14]:
```python
model=tf.keras.Sequential([
  tf.keras.layers.Dense(1,input_dim=8)
])
```

In [15]:
```python
model.compile(loss='mse',optimizer='sgd')
```

In [16]:
```python
trained_model=model.fit(xtrain,ytrain,epochs=100)
```

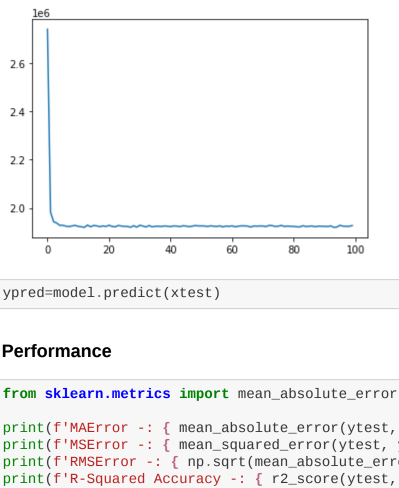
```
Epoch 1/100
1180/1180 [==============================] - 2s 961us/step - loss: 2741445.0000
Epoch 2/100
1180/1180 [==============================] - 1s 968us/step - loss: 1982490.5000
Epoch 3/100
1180/1180 [==============================] - 1s 908us/step - loss: 1942846.1250
Epoch 4/100
1180/1180 [==============================] - 1s 913us/step - loss: 1937429.0000
Epoch 5/100
1180/1180 [==============================] - 1s 916us/step - loss: 1927888.3750
Epoch 6/100
1180/1180 [==============================] - 1s 921us/step - loss: 1927895.3750
Epoch 7/100
1180/1180 [==============================] - 1s 923us/step - loss: 1924653.5000
Epoch 8/100
1180/1180 [==============================] - 1s 921us/step - loss: 1922621.7500
Epoch 9/100
1180/1180 [==============================] - 1s 940us/step - loss: 1925480.0000
Epoch 10/100
1180/1180 [==============================] - 1s 915us/step - loss: 1927612.6250
Epoch 11/100
1180/1180 [==============================] - 1s 936us/step - loss: 1923710.0000
Epoch 12/100
1180/1180 [==============================] - 1s 924us/step - loss: 1921981.2500
Epoch 13/100
1180/1180 [==============================] - 1s 890us/step - loss: 1919801.3750
Epoch 14/100
1180/1180 [==============================] - 1s 913us/step - loss: 1928792.5000
Epoch 15/100
1180/1180 [==============================] - 1s 929us/step - loss: 1921982.3750
Epoch 16/100
1180/1180 [==============================] - 1s 918us/step - loss: 1927272.8750
Epoch 17/100
1180/1180 [==============================] - 1s 934us/step - loss: 1925729.2500
Epoch 18/100
1180/1180 [==============================] - 1s 926us/step - loss: 1922341.6250
Epoch 19/100
1180/1180 [==============================] - 1s 910us/step - loss: 1925630.5000
Epoch 20/100
1180/1180 [==============================] - 1s 919us/step - loss: 1923387.7500
Epoch 21/100
1180/1180 [==============================] - 1s 933us/step - loss: 1928147.5000
Epoch 22/100
1180/1180 [==============================] - 1s 955us/step - loss: 1923526.8750
Epoch 23/100
1180/1180 [==============================] - 1s 914us/step - loss: 1921646.5000
Epoch 24/100
1180/1180 [==============================] - 1s 938us/step - loss: 1927287.8750
Epoch 25/100
1180/1180 [==============================] - 1s 963us/step - loss: 1925015.8750
Epoch 26/100
1180/1180 [==============================] - 1s 945us/step - loss: 1923740.1250
Epoch 27/100
1180/1180 [==============================] - 1s 892us/step - loss: 1923347.0000
Epoch 28/100
1180/1180 [==============================] - 1s 934us/step - loss: 1920090.3750
Epoch 29/100
1180/1180 [==============================] - 1s 952us/step - loss: 1926417.0000
Epoch 30/100
1180/1180 [==============================] - 1s 913us/step - loss: 1920492.6250
Epoch 31/100
1180/1180 [==============================] - 1s 960us/step - loss: 1928095.3750
Epoch 32/100
1180/1180 [==============================] - 1s 925us/step - loss: 1925033.8750
Epoch 33/100
1180/1180 [==============================] - 1s 932us/step - loss: 1920964.3750
Epoch 34/100
1180/1180 [==============================] - 1s 933us/step - loss: 1927076.3750
Epoch 35/100
1180/1180 [==============================] - 1s 955us/step - loss: 1921737.3750
Epoch 36/100
1180/1180 [==============================] - 1s 923us/step - loss: 1923590.2500
Epoch 37/100
1180/1180 [==============================] - 1s 912us/step - loss: 1924489.7500
Epoch 38/100
1180/1180 [==============================] - 1s 935us/step - loss: 1923182.7500
Epoch 39/100
1180/1180 [==============================] - 1s 942us/step - loss: 1925260.3750
Epoch 40/100
1180/1180 [==============================] - 1s 979us/step - loss: 1924350.0000
Epoch 41/100
1180/1180 [==============================] - 1s 952us/step - loss: 1922542.5000
Epoch 42/100
1180/1180 [==============================] - 1s 978us/step - loss: 1925469.2500
Epoch 43/100
1180/1180 [==============================] - 1s 933us/step - loss: 1924687.3750
Epoch 44/100
1180/1180 [==============================] - 1s 960us/step - loss: 1922643.8750
Epoch 45/100
1180/1180 [==============================] - 1s 927us/step - loss: 1926015.8750
Epoch 46/100
1180/1180 [==============================] - 1s 919us/step - loss: 1925090.5000
Epoch 47/100
1180/1180 [==============================] - 1s 929us/step - loss: 1922346.3750
Epoch 48/100
1180/1180 [==============================] - 1s 982us/step - loss: 1924585.0000
Epoch 49/100
1180/1180 [==============================] - 1s 937us/step - loss: 1927035.7500
Epoch 50/100
1180/1180 [==============================] - 1s 983us/step - loss: 1925520.7500
Epoch 51/100
1180/1180 [==============================] - 1s 968us/step - loss: 1925492.2500
Epoch 52/100
1180/1180 [==============================] - 1s 948us/step - loss: 1925255.1250
Epoch 53/100
1180/1180 [==============================] - 1s 938us/step - loss: 1923289.8750
Epoch 54/100
1180/1180 [==============================] - 1s 923us/step - loss: 1925877.0000
Epoch 55/100
1180/1180 [==============================] - 1s 1ms/step - loss: 1924191.2500
Epoch 56/100
1180/1180 [==============================] - 1s 931us/step - loss: 1923412.7500
Epoch 57/100
1180/1180 [==============================] - 1s 927us/step - loss: 1925751.7500
Epoch 58/100
1180/1180 [==============================] - 1s 953us/step - loss: 1921361.6250
Epoch 59/100
1180/1180 [==============================] - 1s 949us/step - loss: 1924899.3750
Epoch 60/100
1180/1180 [==============================] - 1s 936us/step - loss: 1923913.5000
Epoch 61/100
1180/1180 [==============================] - 1s 935us/step - loss: 1925739.3750
Epoch 62/100
1180/1180 [==============================] - 1s 924us/step - loss: 1921685.1250
Epoch 63/100
1180/1180 [==============================] - 1s 934us/step - loss: 1924477.1250
Epoch 64/100
1180/1180 [==============================] - 1s 935us/step - loss: 1925847.5000
Epoch 65/100
1180/1180 [==============================] - 1s 913us/step - loss: 1925512.5000
Epoch 66/100
1180/1180 [==============================] - 1s 930us/step - loss: 1924979.3750
Epoch 67/100
1180/1180 [==============================] - 1s 908us/step - loss: 1920786.0000
Epoch 68/100
1180/1180 [==============================] - 1s 924us/step - loss: 1925218.5000
Epoch 69/100
1180/1180 [==============================] - 1s 926us/step - loss: 1924601.8750
Epoch 70/100
1180/1180 [==============================] - 1s 955us/step - loss: 1925206.6250
Epoch 71/100
1180/1180 [==============================] - 1s 945us/step - loss: 1925125.3750
Epoch 72/100
1180/1180 [==============================] - 1s 940us/step - loss: 1922845.0000
Epoch 73/100
1180/1180 [==============================] - 1s 939us/step - loss: 1927752.1250
Epoch 74/100
1180/1180 [==============================] - 1s 956us/step - loss: 1926666.7500
Epoch 75/100
1180/1180 [==============================] - 1s 916us/step - loss: 1923157.1250
Epoch 76/100
1180/1180 [==============================] - 1s 920us/step - loss: 1924550.2500
Epoch 77/100
1180/1180 [==============================] - 1s 913us/step - loss: 1927469.6250
Epoch 78/100
1180/1180 [==============================] - 1s 932us/step - loss: 1923193.6250
Epoch 79/100
1180/1180 [==============================] - 1s 961us/step - loss: 1924993.1250
Epoch 80/100
1180/1180 [==============================] - 1s 928us/step - loss: 1923683.2500
Epoch 81/100
1180/1180 [==============================] - 1s 934us/step - loss: 1923585.2500
Epoch 82/100
1180/1180 [==============================] - 1s 918us/step - loss: 1921709.0000
Epoch 83/100
1180/1180 [==============================] - 1s 935us/step - loss: 1921624.5000
Epoch 84/100
1180/1180 [==============================] - 1s 915us/step - loss: 1926320.5000
Epoch 85/100
1180/1180 [==============================] - 1s 928us/step - loss: 1922851.8750
Epoch 86/100
1180/1180 [==============================] - 1s 957us/step - loss: 1924267.3750
Epoch 87/100
1180/1180 [==============================] - 1s 923us/step - loss: 1925245.1250
Epoch 88/100
1180/1180 [==============================] - 1s 939us/step - loss: 1922070.0000
Epoch 89/100
1180/1180 [==============================] - 1s 972us/step - loss: 1924640.2500
Epoch 90/100
1180/1180 [==============================] - 1s 934us/step - loss: 1923996.7500
Epoch 91/100
1180/1180 [==============================] - 1s 931us/step - loss: 1923918.5000
Epoch 92/100
1180/1180 [==============================] - 1s 960us/step - loss: 1922978.3750
Epoch 93/100
1180/1180 [==============================] - 1s 961us/step - loss: 1925788.5000
Epoch 94/100
1180/1180 [==============================] - 1s 993us/step - loss: 1919612.2500
Epoch 95/100
1180/1180 [==============================] - 1s 943us/step - loss: 1920914.6250
Epoch 96/100
1180/1180 [==============================] - 1s 925us/step - loss: 1928031.6250
Epoch 97/100
1180/1180 [==============================] - 1s 912us/step - loss: 1924485.1250
Epoch 98/100
1180/1180 [==============================] - 1s 953us/step - loss: 1923840.7500
Epoch 99/100
1180/1180 [==============================] - 1s 961us/step - loss: 1924001.6250
Epoch 100/100
1180/1180 [==============================] - 1s 942us/step - loss: 1926962.2500
```

In [17]:
```python
trained_model.history['loss']
```

Out[17]:
```
[2741445.0,
 1982490.5,
 1942846.125,
 1937429.0,
 1927888.375,
 1927895.375,
 1924653.5,
 1922621.75,
 1925480.0,
 1927612.625,
 1923710.0,
 1921981.25,
 1919801.375,
 1928792.5,
 1921982.375,
 1927272.875,
 1925729.25,
 1922341.625,
 1925630.5,
 1923387.75,
 1928147.5,
 1923526.875,
 1921646.5,
 1927287.875,
 1925015.875,
 1923740.125,
 1923347.0,
 1920090.375,
 1926417.0,
 1920492.625,
 1928095.375,
 1925033.875,
 1920964.375,
 1927076.375,
 1921737.375,
 1923590.25,
 1924489.75,
 1923182.75,
 1925260.375,
 1924350.0,
 1922542.5,
 1925469.25,
 1924687.375,
 1922643.875,
 1926015.875,
 1925090.5,
 1922346.375,
 1924585.0,
 1927035.75,
 1925520.75,
 1925492.25,
 1925255.125,
 1923289.875,
 1925877.0,
 1924191.25,
 1923412.75,
 1925751.75,
 1921361.625,
 1924899.375,
 1923913.5,
 1925739.375,
 1921685.125,
 1924477.125,
 1925847.5,
 1925512.5,
 1924979.375,
 1920786.0,
 1925218.5,
 1924601.875,
 1925206.625,
 1925125.375,
 1922845.0,
 1927752.125,
 1926666.75,
 1923157.125,
 1924550.25,
 1927469.625,
 1923193.625,
 1924993.125,
 1923683.25,
 1923585.25,
 1921709.0,
 1921624.5,
 1926320.5,
 1922851.875,
 1924267.375,
 1925245.125,
 1922070.0,
 1924640.25,
 1923996.75,
 1923918.5,
 1922978.375,
 1925788.5,
 1919612.25,
 1920914.625,
 1928031.625,
 1924485.125,
 1923840.75,
 1924001.625,
 1926962.25]
```

In [18]:
```python
plt.plot(trained_model.history['loss'])
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x7f375891f310>]
```



In [19]:
```python
ypred=model.predict(xtest)
```

Performance

In [20]:
```python
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

print(f'MAError -: { mean_absolute_error(ytest, ypred) }')
print(f'MSError -: { mean_squared_error(ytest, ypred) }')
print(f'RMSError -: { np.sqrt(mean_absolute_error(ytest, ypred)) }')
print(f'R-Squared Accuracy -: { r2_score(ytest, ypred) }')
```

```
MAError -: 849.3594103039836
MSError -: 1773884.7403221824
RMSError -: 29.143771380931184
R-Squared Accuracy -: 0.8843607509734402
```