# Simple Housing Dataset

Create a regression model that predicts the price of boston house

```
!wget https://storage.googleapis.com/nicksdemobucket/housing-data.csv
```

```
--2019-03-30 11:55:25--  https://storage.googleapis.com/nicksdemobucket/housing-data.
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.141.128, 2607:f8b
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.141.128|:443...
HTTP request sent, awaiting response... 200 OK
Length: 816 [application/octet-stream]
Saving to: ‘housing-data.csv’

housing-data.csv    100%[===================>]     816  --.-KB/s    in 0s

2019-03-30 11:55:25 (15.1 MB/s) - ‘housing-data.csv’ saved [816/816]
```

```
import pandas as pd
df= pd.read_csv('housing-data.csv')
```

```
df.head()
```

| | sqft | bdrms | age | price |
|---|---|---|---|---|
| **0** | 2104 | 3 | 70 | 399900 |
| **1** | 1600 | 3 | 28 | 329900 |
| **2** | 2400 | 3 | 44 | 369000 |
| **3** | 1416 | 2 | 49 | 232000 |
| **4** | 3000 | 4 | 75 | 539900 |

```
df.describe()
```

| | sqft | bdrms | age | price |
|---|---|---|---|---|
| **count** | 47.000000 | 47.000000 | 47.000000 | 47.000000 |
| mean | 2000.680851 | 3.170213 | 42.744681 | 340412.659574 |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47 entries, 0 to 46
Data columns (total 4 columns):
sqft     47 non-null int64
bdrms    47 non-null int64
age      47 non-null int64
price    47 non-null int64
dtypes: int64(4)
memory usage: 1.5 KB
```

```
from sklearn.preprocessing import OneHotEncoder
X = df.iloc[:,:3] #X
X.shape
```

```
(47, 3)
```

```
y = df.iloc [:,3:] #y
y.shape
```

```
(47, 1)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.train import AdamOptimizer
```

```
model = Sequential()

model.add(Dense(60, input_shape=(3,), activation='relu'))
model.add(Dense(60, activation='relu'))
model.add(Dense (1))

model.compile(optimizer=AdamOptimizer(0.01),
              loss ='mean_squared_error',
              metrics=['mean_absolute_error'])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/kera
Instructions for updating:
Use tf.cast instead.
```

```
h = model.fit(X, y, epochs=100,batch_size=16, validation_split=0.2)
```

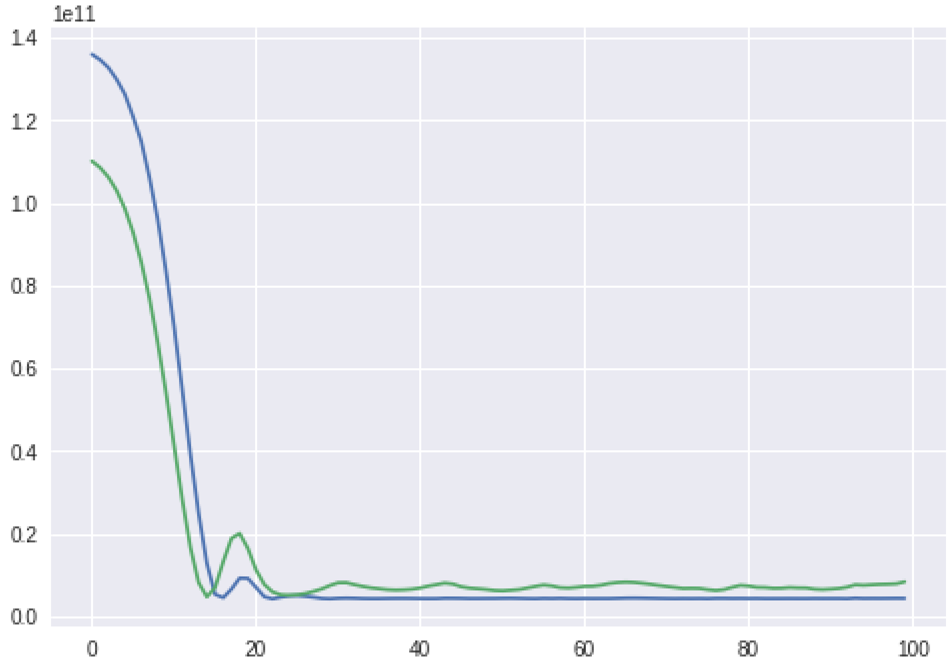
```
Train on 37 samples, validate on 10 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/
Instructions for updating:
Use tf.cast instead.
Epoch 1/100
37/37 [==============================] - 0s 9ms/sample - loss: 135910616313.0811 - me
Epoch 2/100
37/37 [==============================] - 0s 364us/sample - loss: 134570994494.2703 -
Epoch 3/100
37/37 [==============================] - 0s 618us/sample - loss: 132700036455.7838 -
Epoch 4/100
37/37 [==============================] - 0s 377us/sample - loss: 129851645509.1892 -
Epoch 5/100
37/37 [==============================] - 0s 313us/sample - loss: 126279999709.4054 -
Epoch 6/100
37/37 [==============================] - 0s 405us/sample - loss: 120869419727.5676 -
Epoch 7/100
37/37 [==============================] - 0s 408us/sample - loss: 114874362132.7568 -
Epoch 8/100
37/37 [==============================] - 0s 359us/sample - loss: 106137831368.6487 -
Epoch 9/100
37/37 [==============================] - 0s 327us/sample - loss: 96033242028.9730 - m
Epoch 10/100
37/37 [==============================] - 0s 366us/sample - loss: 83675766894.7027 - m
Epoch 11/100
37/37 [==============================] - 0s 328us/sample - loss: 70092994615.3513 - m
Epoch 12/100
37/37 [==============================] - 0s 396us/sample - loss: 54862042084.3243 - m
Epoch 13/100
37/37 [==============================] - 0s 336us/sample - loss: 39121233975.3514 - m
Epoch 14/100
37/37 [==============================] - 0s 298us/sample - loss: 24855652490.3784 - m
Epoch 15/100
37/37 [==============================] - 0s 270us/sample - loss: 12803896873.5135 - m
Epoch 16/100
37/37 [==============================] - 0s 475us/sample - loss: 5283406612.7568 - me
Epoch 17/100
37/37 [==============================] - 0s 382us/sample - loss: 4480017484.1081 - me
Epoch 18/100
37/37 [==============================] - 0s 372us/sample - loss: 6552781270.4865 - me
Epoch 19/100
37/37 [==============================] - 0s 403us/sample - loss: 9220114653.4054 - me
Epoch 20/100
37/37 [==============================] - 0s 465us/sample - loss: 9153664249.0811 - me
Epoch 21/100
37/37 [==============================] - 0s 512us/sample - loss: 6952015325.4054 - me
Epoch 22/100
37/37 [==============================] - 0s 357us/sample - loss: 4707606458.8108 - me
Epoch 23/100
37/37 [==============================] - 0s 518us/sample - loss: 4183094161.2973 - me
Epoch 24/100
37/37 [==============================] - 0s 450us/sample - loss: 4604909014.4865 - me
Epoch 25/100
37/37 [==============================] - 0s 493us/sample - loss: 4897991908.3243 - me
Epoch 26/100
37/37 [==============================] - 0s 400us/sample - loss: 4975622974.2703 - me
Epoch 27/100
```

```
h.history.keys()
```

```
dict_keys(['loss', 'mean_absolute_error', 'val_loss', 'val_mean_absolute_error'])
```

```
import matplotlib.pyplot as plt
plt.plot(h.history['loss'])
plt.plot(h.history['val_loss
                  '])
```

```
[<matplotlib.lines.Line2D at 0x7f7caea8eba8>]
```



## Binary Classification

Create a binary classifier for the titanic dataset, will person x survive?

```
!wget https://storage.googleapis.com/nicksdemobucket/titanic-train.csv
```

```
--2019-03-31 11:38:38--  https://storage.googleapis.com/nicksdemobucket/titanic-trair
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.195.128, 2607:f8t
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.195.128|:443...
HTTP request sent, awaiting response... 200 OK
Length: 61194 (60K) [application/octet-stream]
Saving to: ‘titanic-train.csv’

titanic-train.csv   100%[===================>]  59.76K  --.-KB/s    in 0.001s

2019-03-31 11:38:38 (93.3 MB/s) - ‘titanic-train.csv’ saved [61194/61194]
```

```
import pandas as pd
df= pd.read_csv('titanic-train.csv')
```

```
df.head()
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7. |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71. |
| | | | | Heikkinen, | | | | | STON/O2. | |

```
df.describe()
```

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | F |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329 |

```
data = df.drop(['Name','Ticket','Fare','Cabin', 'Embarked','Age'],axis =1)
data.head()
```

| | PassengerId | Survived | Pclass | Sex | SibSp | Parch |
|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | male | 1 | 0 |
| **1** | 2 | 1 | 1 | female | 1 | 0 |
| **2** | 3 | 1 | 3 | female | 0 | 0 |
| **3** | 4 | 1 | 1 | female | 1 | 0 |
| **4** | 5 | 0 | 3 | male | 0 | 0 |

```
from sklearn.preprocessing import LabelEncoder
label_encoder_sex = LabelEncoder()


data.iloc[: , 3] = label_encoder_sex.fit_transform(data.iloc[:,3]) #data.iloc[:,3]


data_ordered = data[['PassengerId','Sex','SibSp','Parch','Pclass','Survived']]
data_ordered.head()
```

| | PassengerId | Sex | SibSp | Parch | Pclass | Survived |
|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 0 | 3 | 0 |
| **1** | 2 | 0 | 1 | 0 | 1 | 1 |
| **2** | 3 | 0 | 0 | 0 | 3 | 1 |
| **3** | 4 | 0 | 1 | 0 | 1 | 1 |
| **4** | 5 | 1 | 0 | 0 | 3 | 0 |

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.train import AdamOptimizer

from sklearn.preprocessing import OneHotEncoder


X = data_ordered.iloc[:,:5] #X
X.shape
```

```
(891, 5)
```

```
y = data_ordered.iloc[:,5:] #y
y.shape
```

```
(891, 1)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/data.py:645: DataConvers
  return self.partial_fit(X, y)
/usr/local/lib/python3.6/dist-packages/sklearn/base.py:464: DataConversionWarning: Da
  return self.fit(X, **fit_params).transform(X)
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense


model = Sequential()

model.add(Dense(output_dim = 10, init = 'uniform', activation='relu',input_dim
                = 5))
model.add(Dense(output_dim = 10, init = 'uniform', activation='relu'))
model.add(Dense(output_dim = 1, init= 'uniform', activation = 'sigmoid'))

model.compile(optimizer = 'adam', loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:4: UserWarning: Update y
  after removing the cwd from sys.path.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: UserWarning: Update y
  """
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:6: UserWarning: Update y
```

```
h = model.fit(X, y, epochs=100, batch_size=16, validation_split=0.2)
```

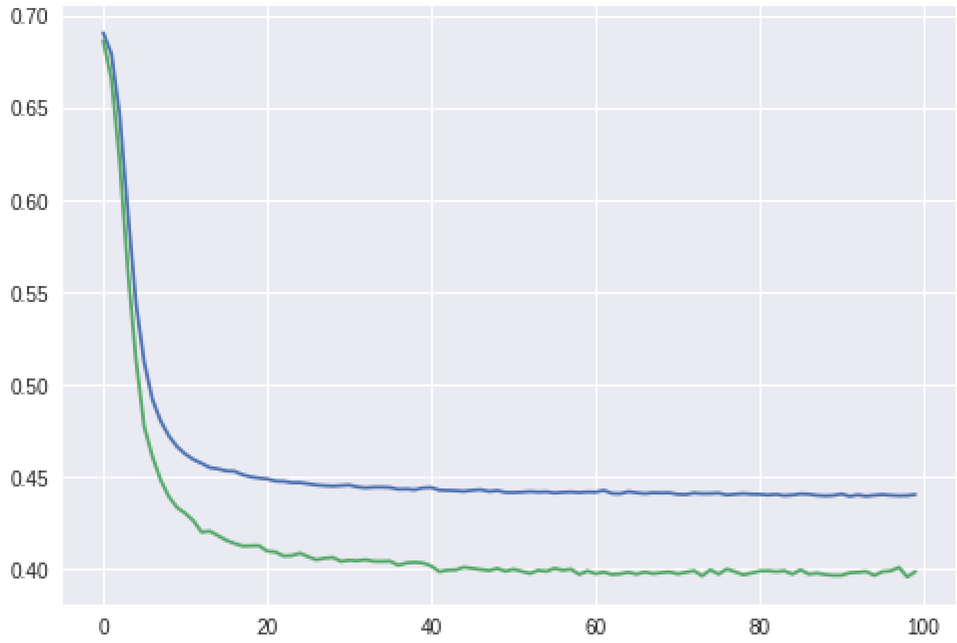
```
Train on 712 samples, validate on 179 samples
Epoch 1/100
712/712 [==============================] - 0s 582us/step - loss: 0.6910 - acc: 0.6138
Epoch 2/100
712/712 [==============================] - 0s 95us/step - loss: 0.6795 - acc: 0.6096
Epoch 3/100
712/712 [==============================] - 0s 91us/step - loss: 0.6465 - acc: 0.6629
Epoch 4/100
712/712 [==============================] - 0s 101us/step - loss: 0.5935 - acc: 0.7837
Epoch 5/100
712/712 [==============================] - 0s 93us/step - loss: 0.5437 - acc: 0.7879
Epoch 6/100
712/712 [==============================] - 0s 93us/step - loss: 0.5122 - acc: 0.7963
Epoch 7/100
712/712 [==============================] - 0s 75us/step - loss: 0.4921 - acc: 0.7921
Epoch 8/100
712/712 [==============================] - 0s 73us/step - loss: 0.4804 - acc: 0.7949
Epoch 9/100
712/712 [==============================] - 0s 74us/step - loss: 0.4723 - acc: 0.7978
```

```
h.history.keys()
```

```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
Epoch 12/100
```

```
import matplotlib.pyplot as plt
plt.plot(h.history['loss'])
plt.plot(h.history['val_loss'])
```

```
[<matplotlib.lines.Line2D at 0x7ff9c3273fd0>]
```



```
Epoch 25/100
712/712 [==============================] - 0s 90us/step - loss: 0.4463 - acc: 0.8062
```

```
Epoch 29/100
```

# MNIST

See how well a fully connected Neural Network performs on MNSIT

```
from tensorflow.keras import datasets
(x_train, y_train), (x_test, y_test) = datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mni
11493376/11490434 [==============================] - 0s 0us/step
```

```
import pandas as pd
```

```
x_train.shape
```

```
(60000, 28, 28)
```

```
x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape (10000, 784)
y_train = pd.get_dummies(y_train)
y_test = pd.get_dummies(y_test)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
model = Sequential()
```

```
model.add(Dense(20,input_shape=(784,), activation='relu'))
model.add(Dense(20,input_shape=(784,), activation='relu'))
model.add(Dense(10, activation='softmax'))
```

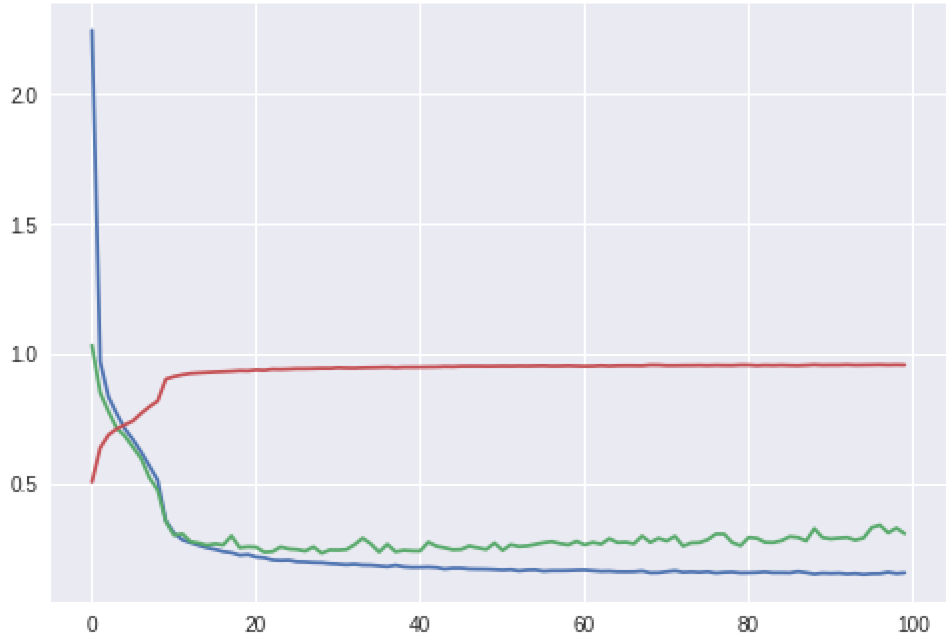
```
model.compile (optimizer='adam',
              loss= 'categorical_crossentropy',
              metrics=['accuracy'])
```

```
h = model.fit(x_train, y_train, epochs=100, batch_size=16,validation_split=0.2)
```

```
Train on 48000 samples, validate on 12000 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/
Instructions for updating:
Use tf.cast instead.
Epoch 1/100
48000/48000 [==============================] - 6s 120us/sample - loss: 2.2455 - acc:
Epoch 2/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.9709 - acc:
Epoch 3/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.8360 - acc:
Epoch 4/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.7711 - acc:
Epoch 5/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.7123 - acc:
Epoch 6/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.6707 - acc:
Epoch 7/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.6225 - acc:
Epoch 8/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.5688 - acc:
Epoch 9/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.5145 - acc:
Epoch 10/100
48000/48000 [==============================] - 6s 115us/sample - loss: 0.3581 - acc:
Epoch 11/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.3103 - acc:
Epoch 12/100
48000/48000 [==============================] - 5s 113us/sample - loss: 0.2847 - acc:
Epoch 13/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.2735 - acc:
Epoch 14/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.2634 - acc:
Epoch 15/100
48000/48000 [==============================] - 5s 113us/sample - loss: 0.2541 - acc:
Epoch 16/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.2478 - acc:
Epoch 17/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.2393 - acc:
Epoch 18/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.2351 - acc:
Epoch 19/100
48000/48000 [==============================] - 5s 113us/sample - loss: 0.2253 - acc:
Epoch 20/100
48000/48000 [==============================] - 5s 111us/sample - loss: 0.2278 - acc:
Epoch 21/100
48000/48000 [==============================] - 5s 114us/sample - loss: 0.2185 - acc:
Epoch 22/100
48000/48000 [==============================] - 5s 113us/sample - loss: 0.2154 - acc:
Epoch 23/100
48000/48000 [==============================] - 5s 112us/sample - loss: 0.2069 - acc:
```

```
import matplotlib.pyplot as plt
plt.plot(h.history['loss'])
plt.plot(h.history['val_loss'])
plt.plot(h.history['acc'])
```

```
[<matplotlib.lines.Line2D at 0x7f44b219a048>]
```


2.0
1.5
1.0
0.5
0
20
40
60
80
100