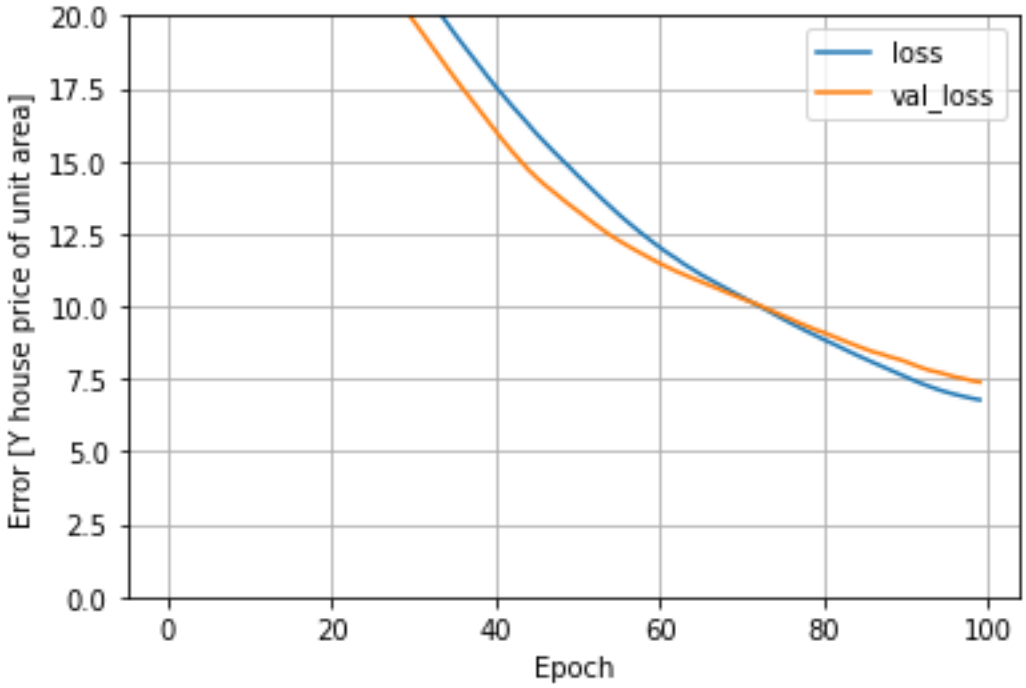
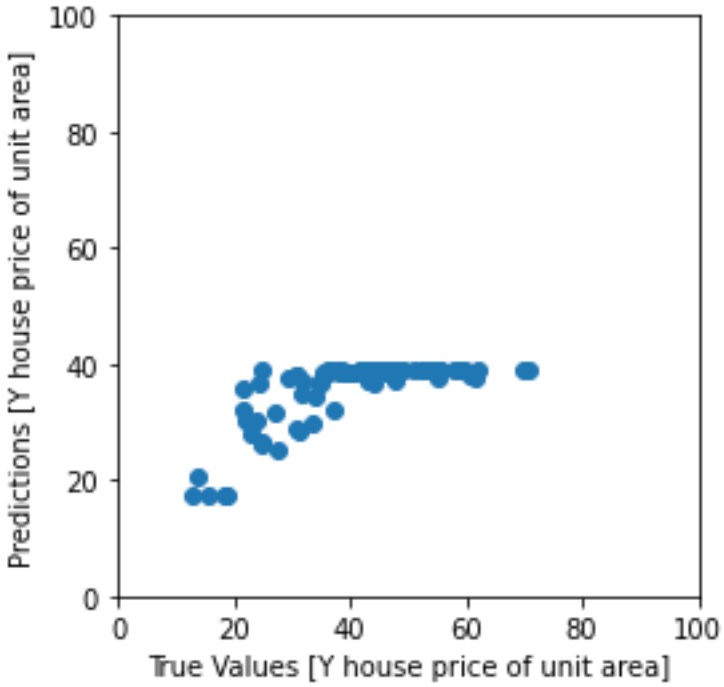
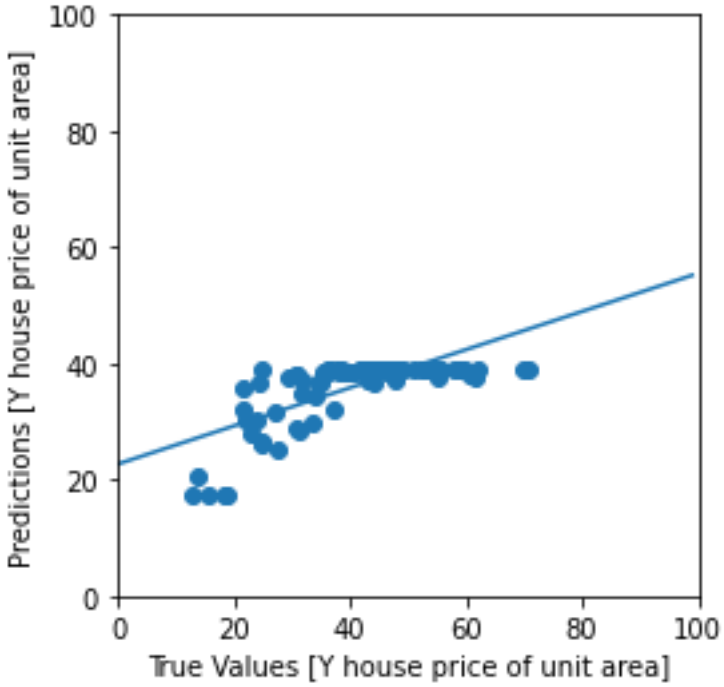
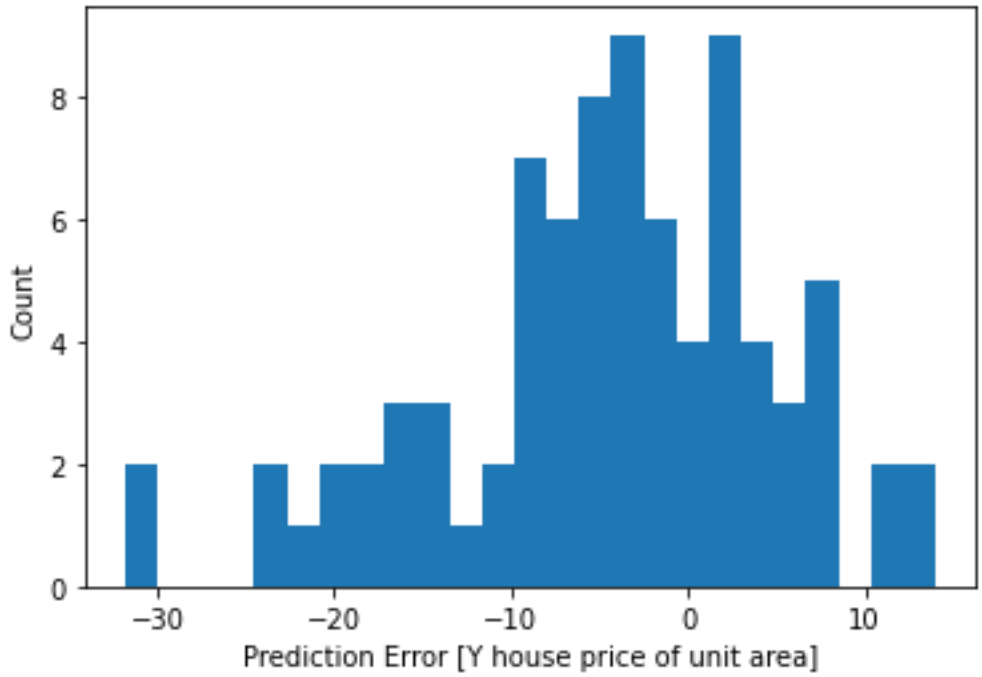
```python
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='sigmoid'),
      layers.Dense(64, activation='sigmoid'),
      layers.Dense(1)
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

RESULTS



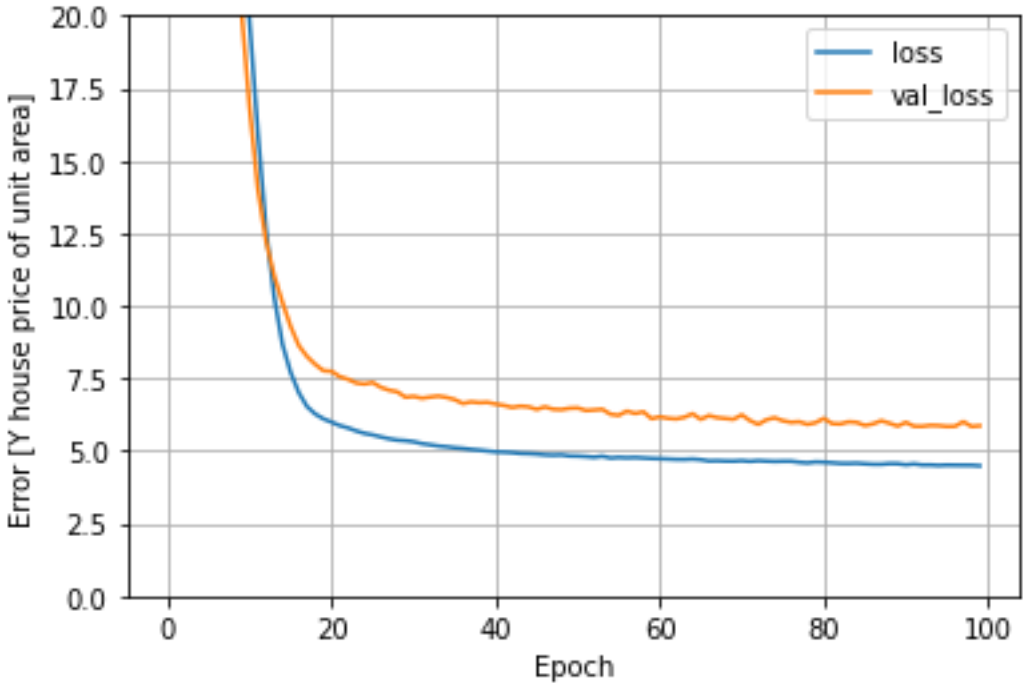

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

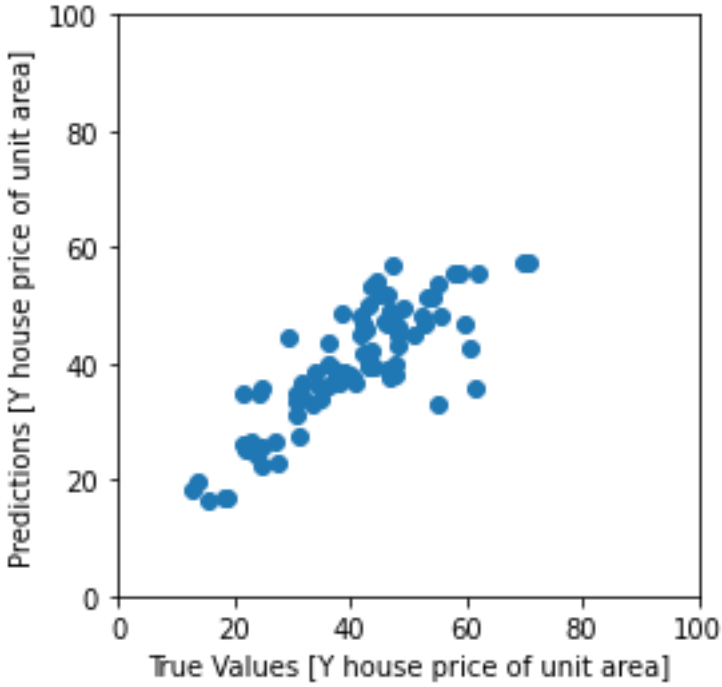

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

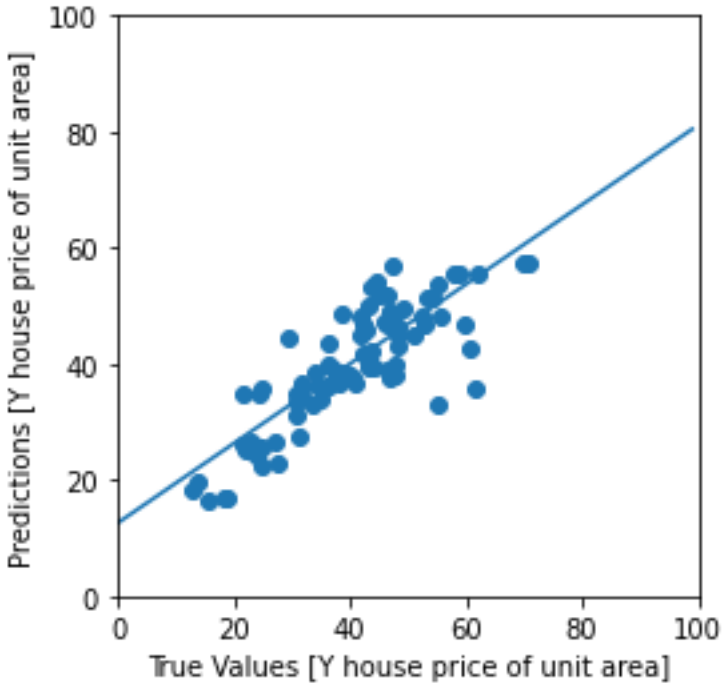


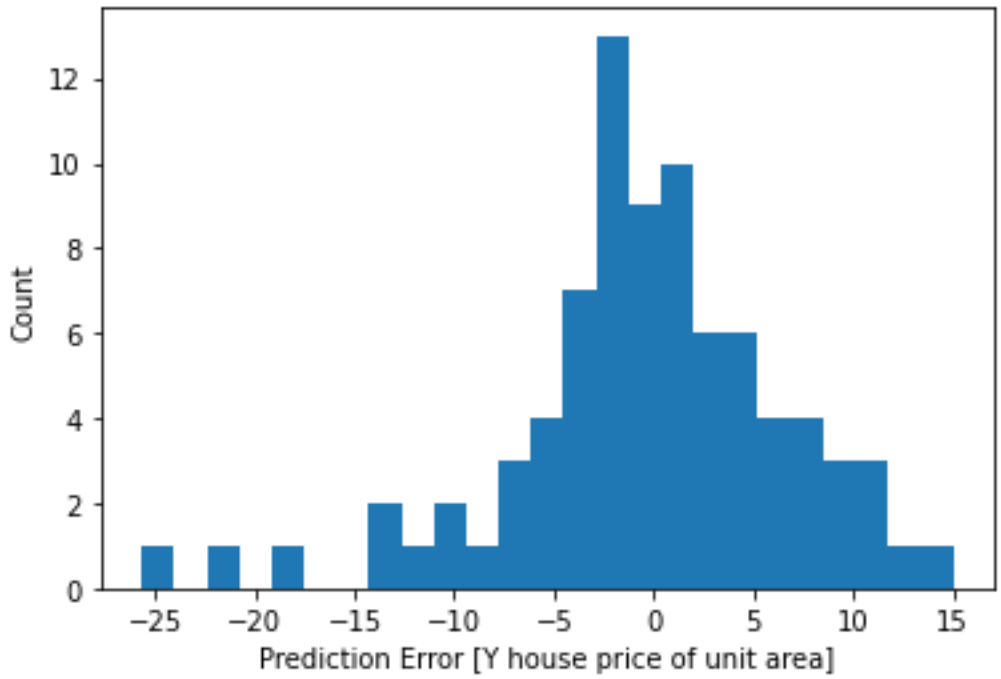

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

RESULTS


loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100


Predictions [Y house price of unit area]
True Values [Y house price of unit area]
100
80
60
40
20
0
0
20
40
60
80
100

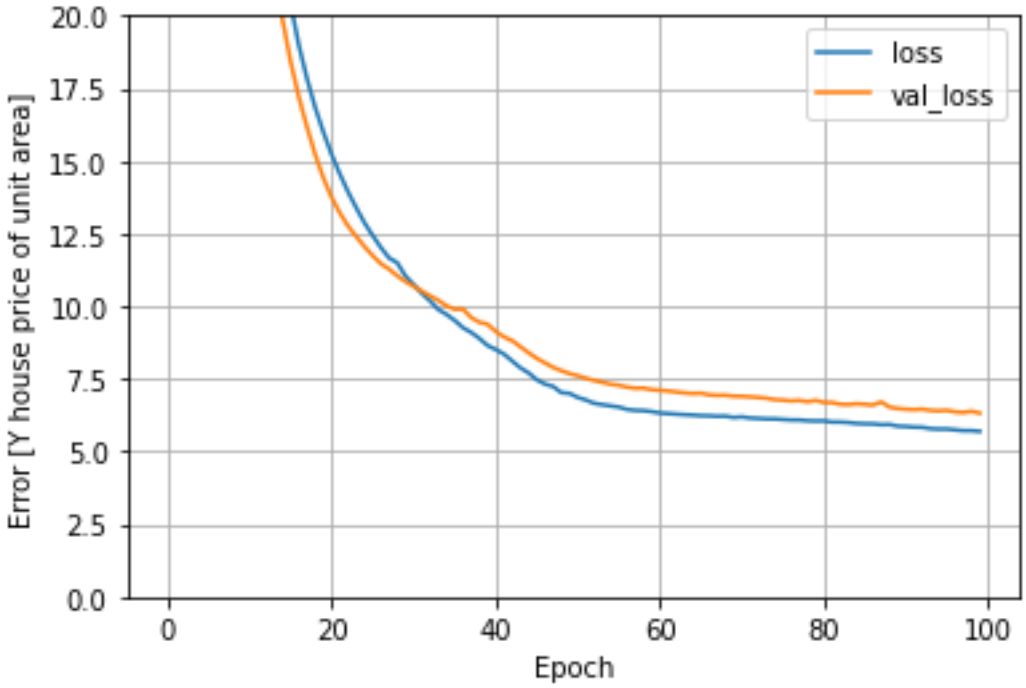


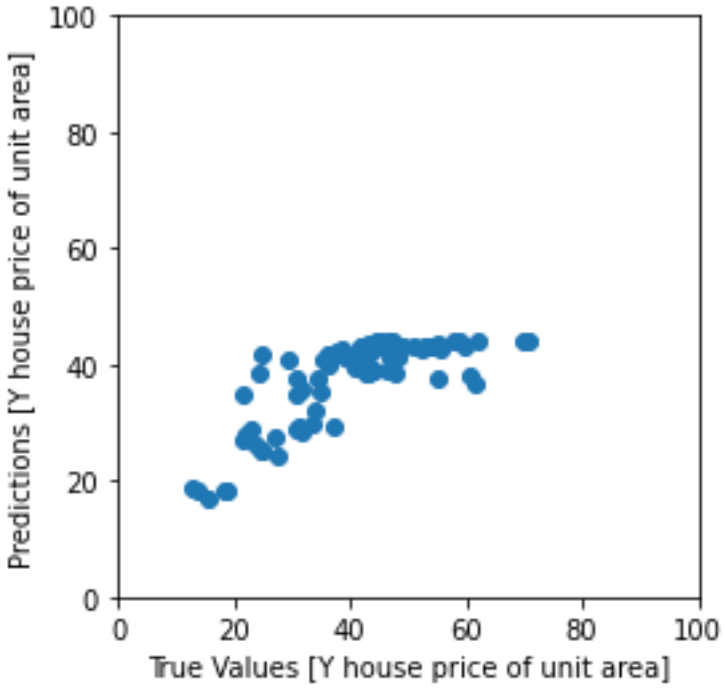


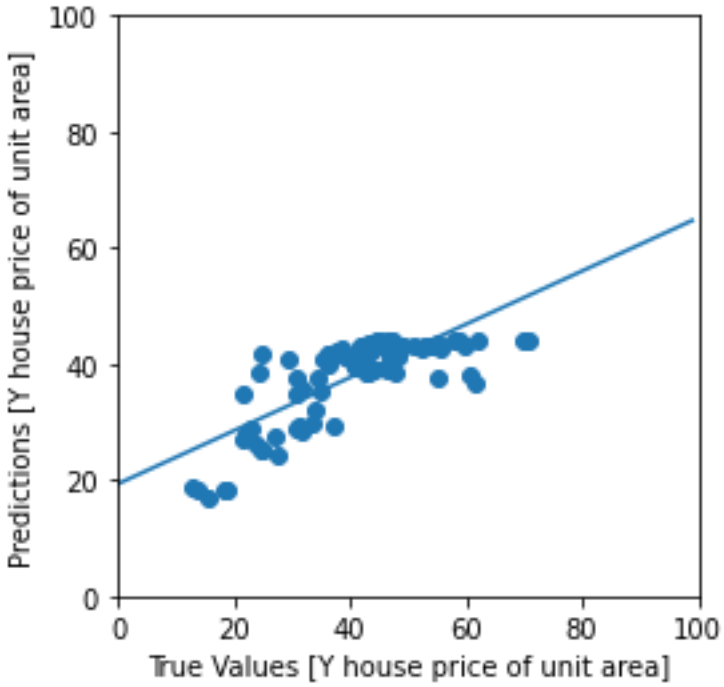

```python
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='tanh'),
      layers.Dense(64, activation='tanh'),
      layers.Dense(1)
  ])
```

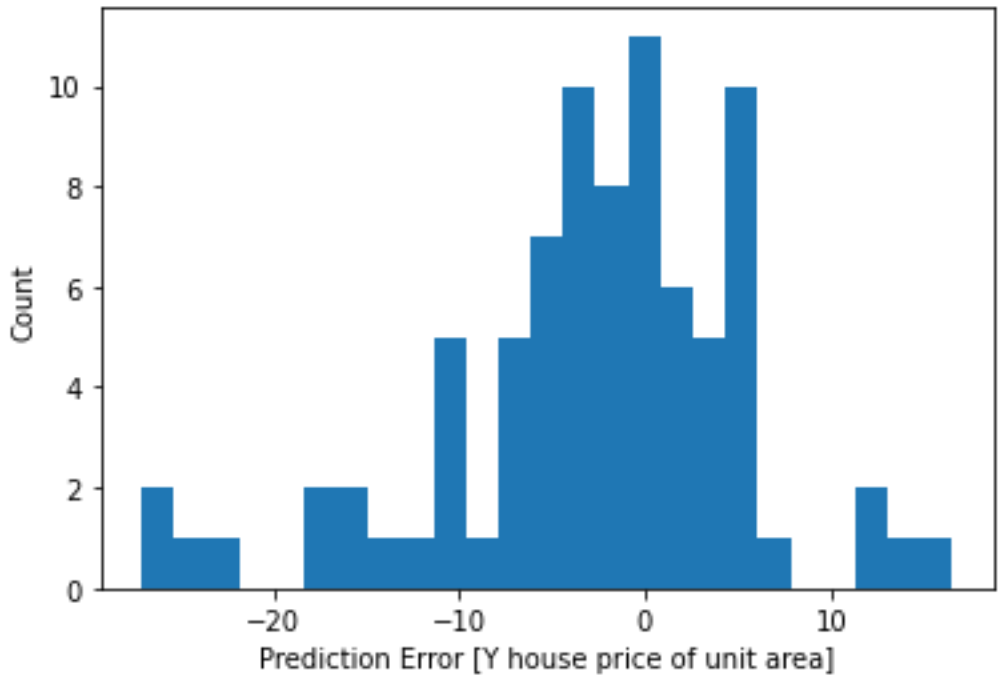
```
model.compile(loss='mean_absolute_error',
              optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
return model
```

RESULTS





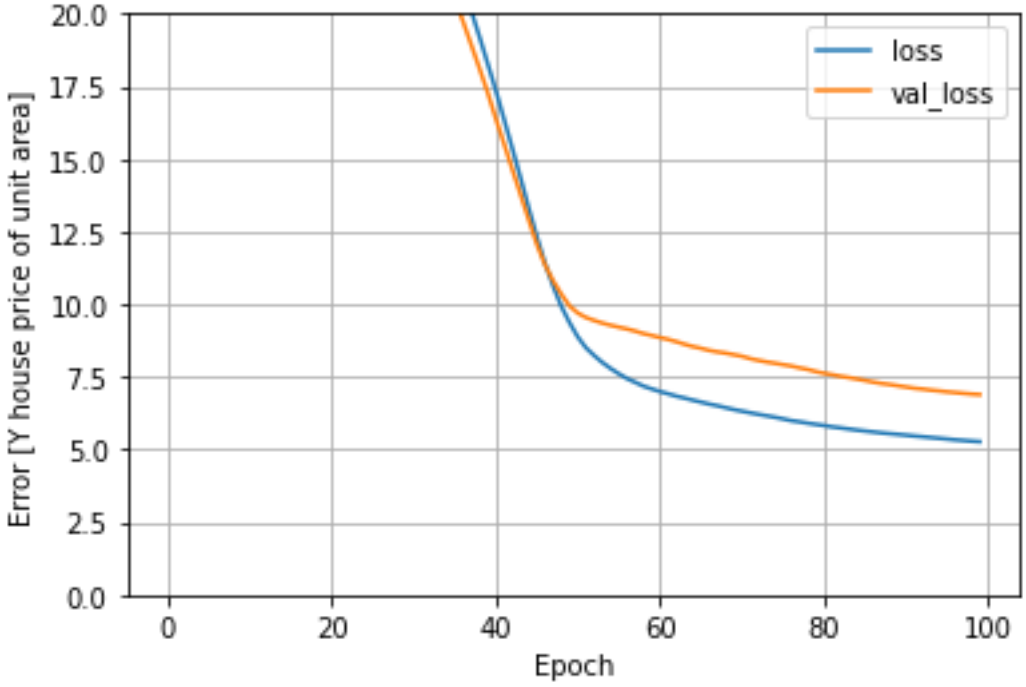


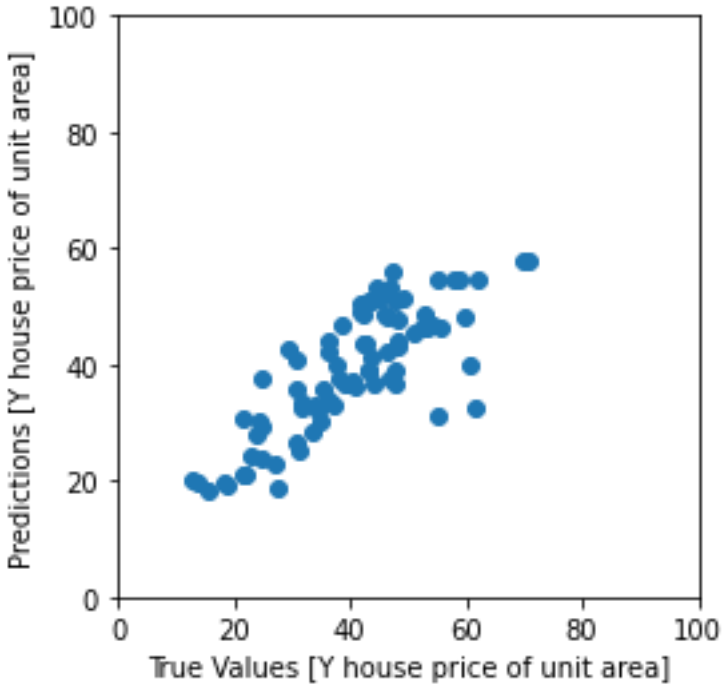


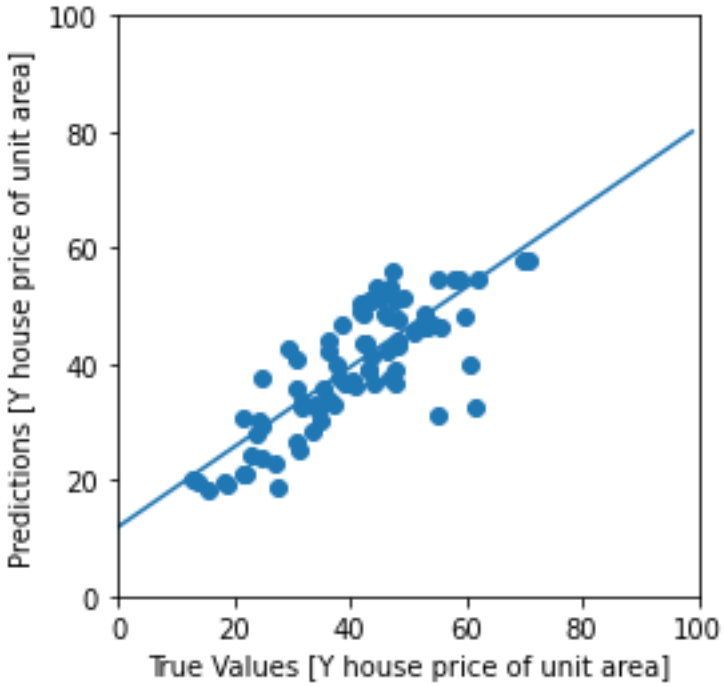

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
  ])
```

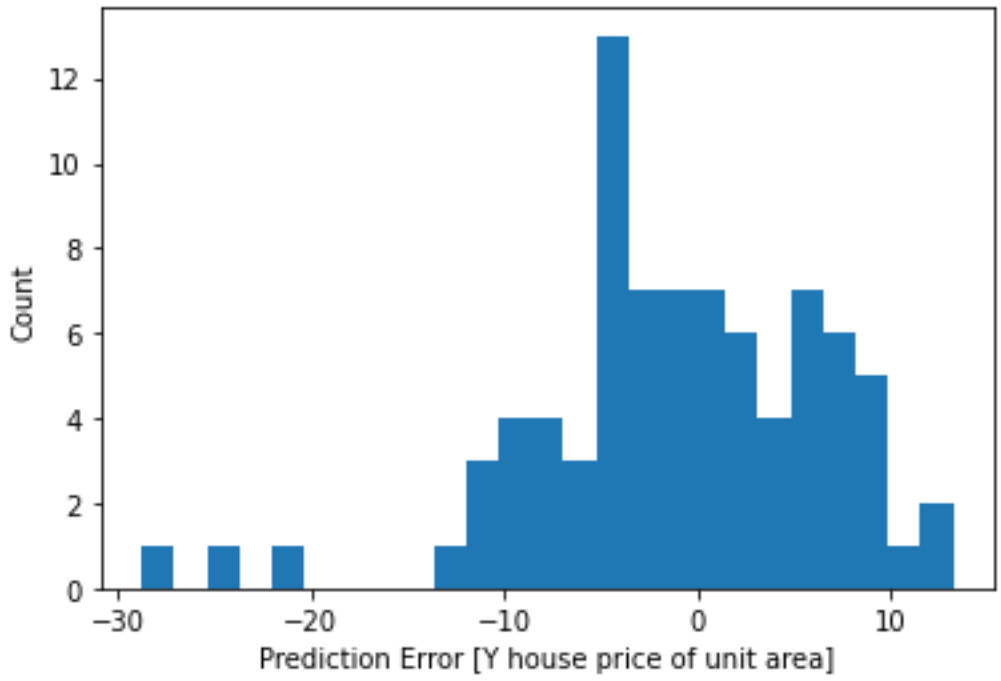
```
  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

RESULTS





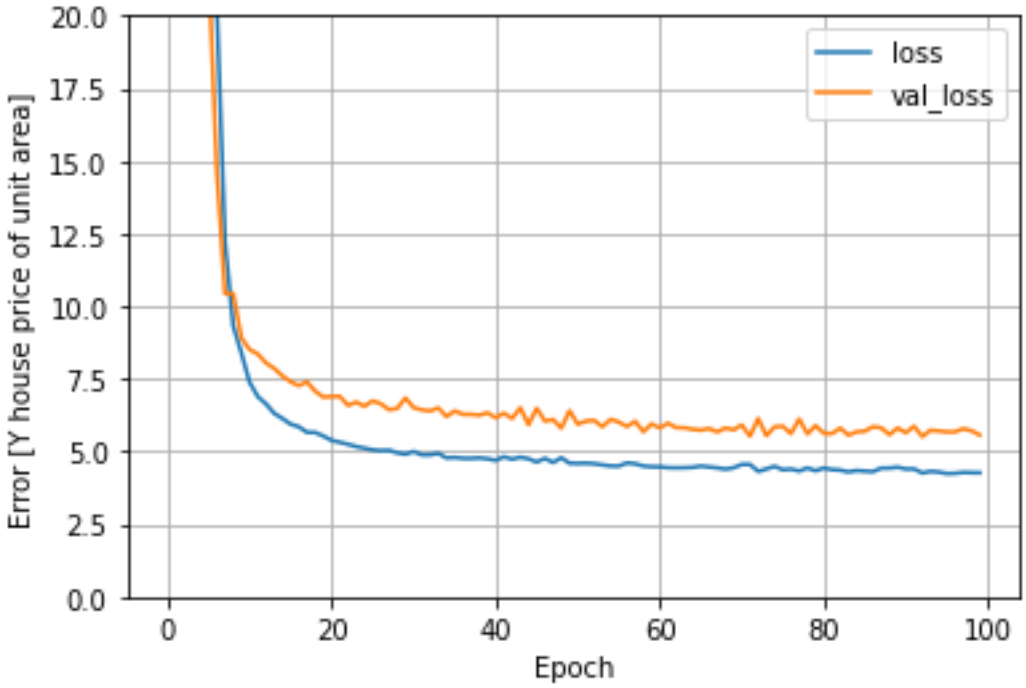


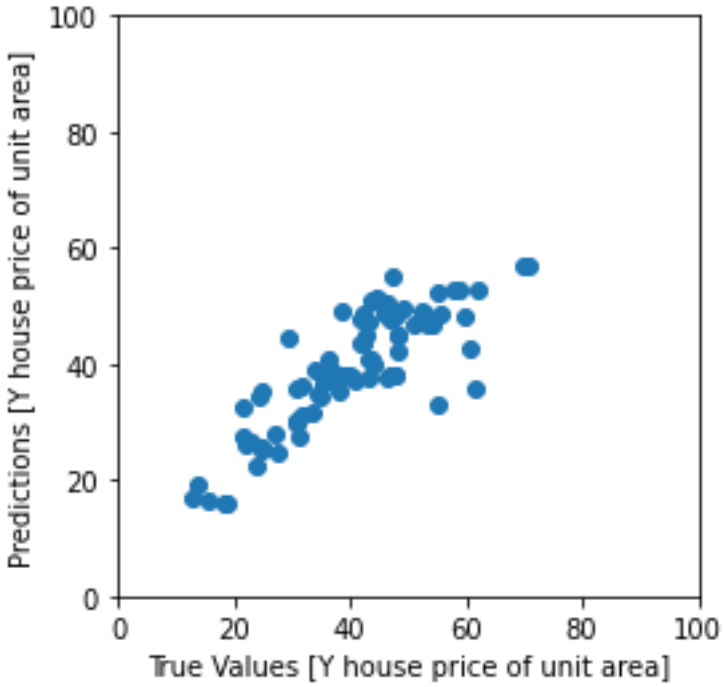


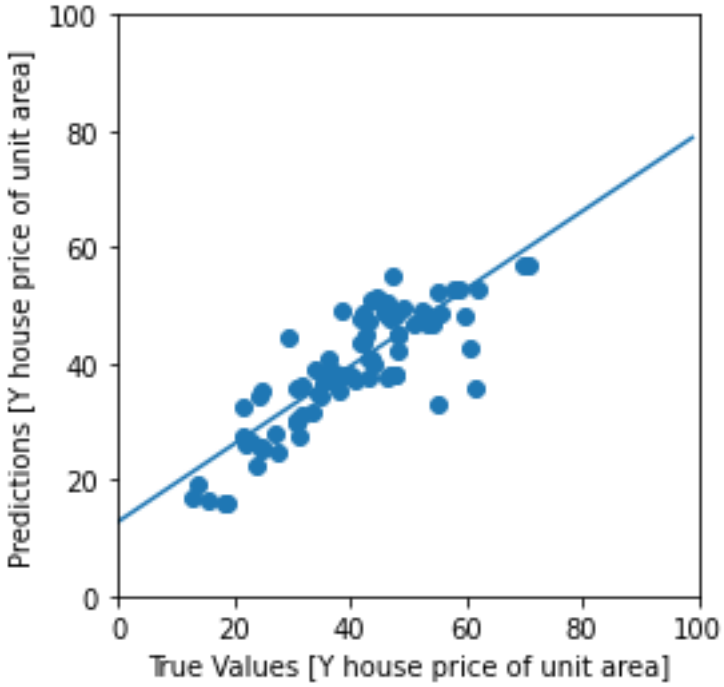

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
```

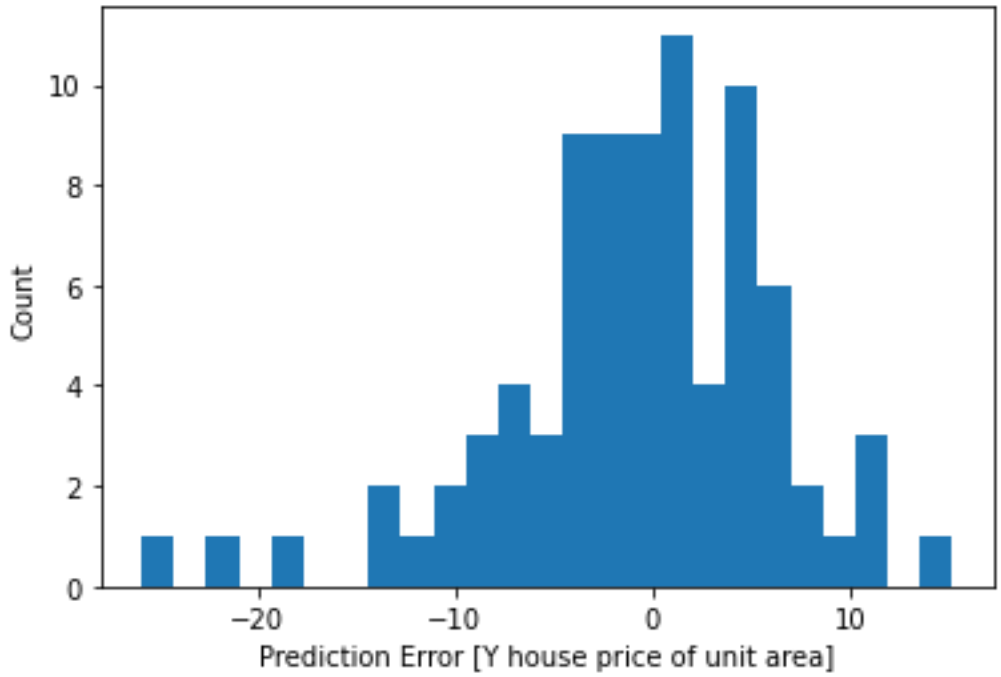
```
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

RESULTS



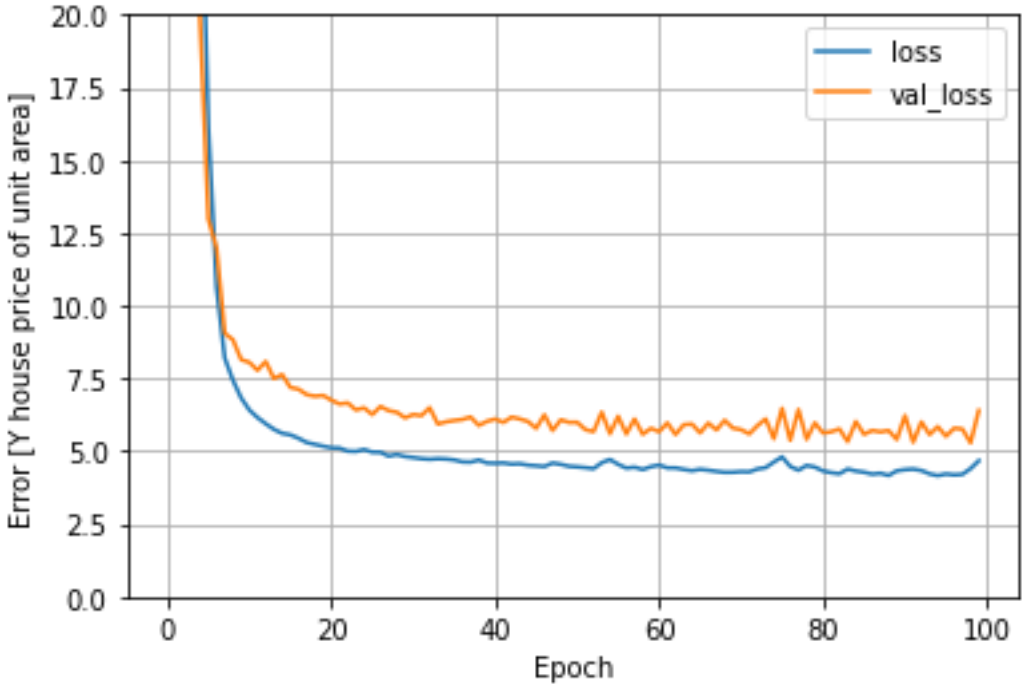

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

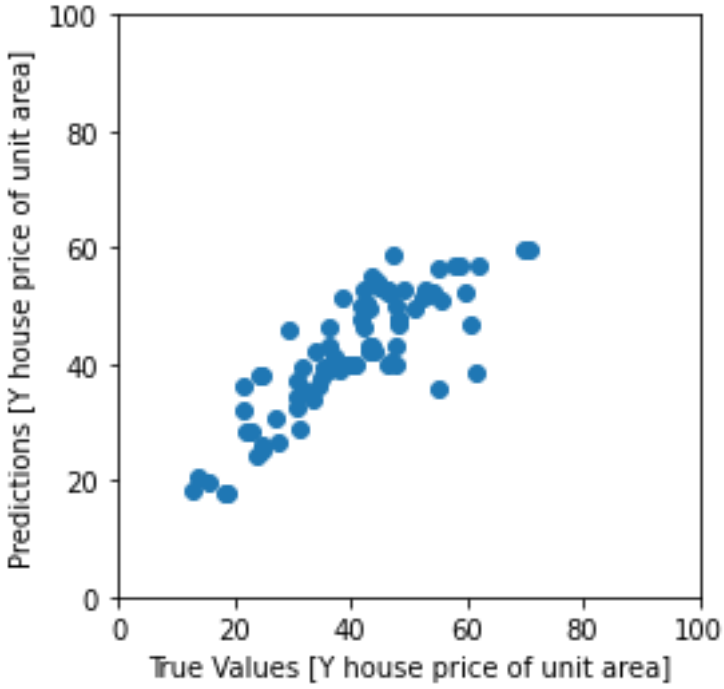

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

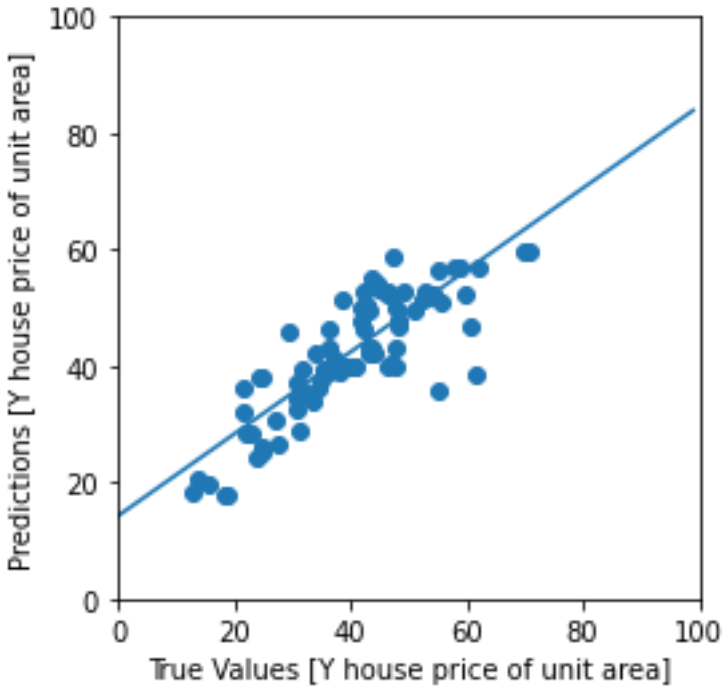


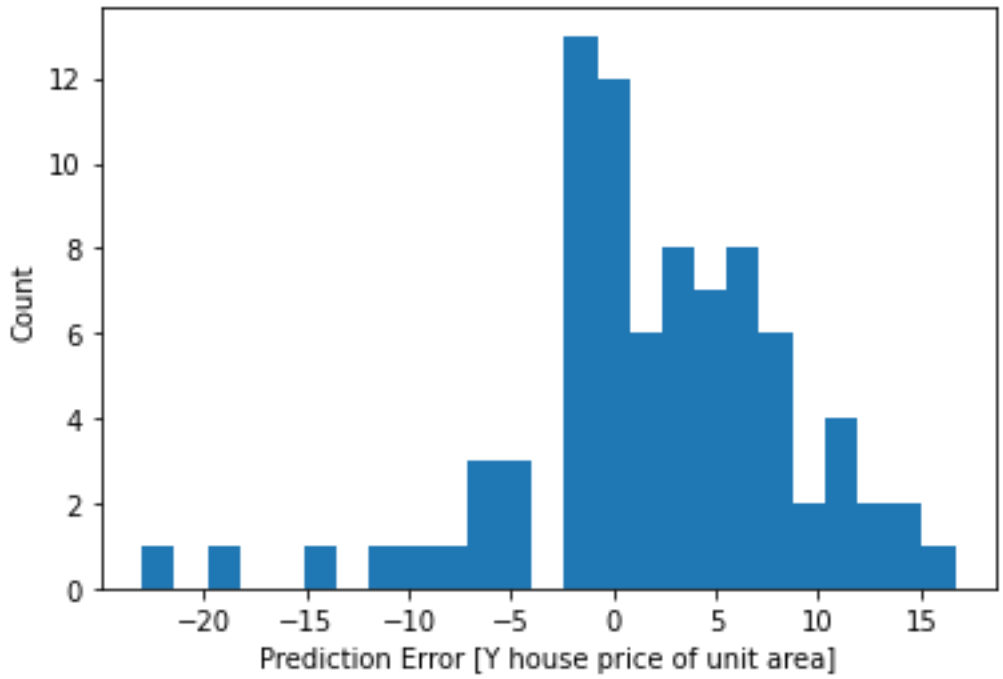

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

RESULTS


loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100


Predictions [Y house price of unit area]
True Values [Y house price of unit area]
100
80
60
40
20
0
0
20
40
60
80
100

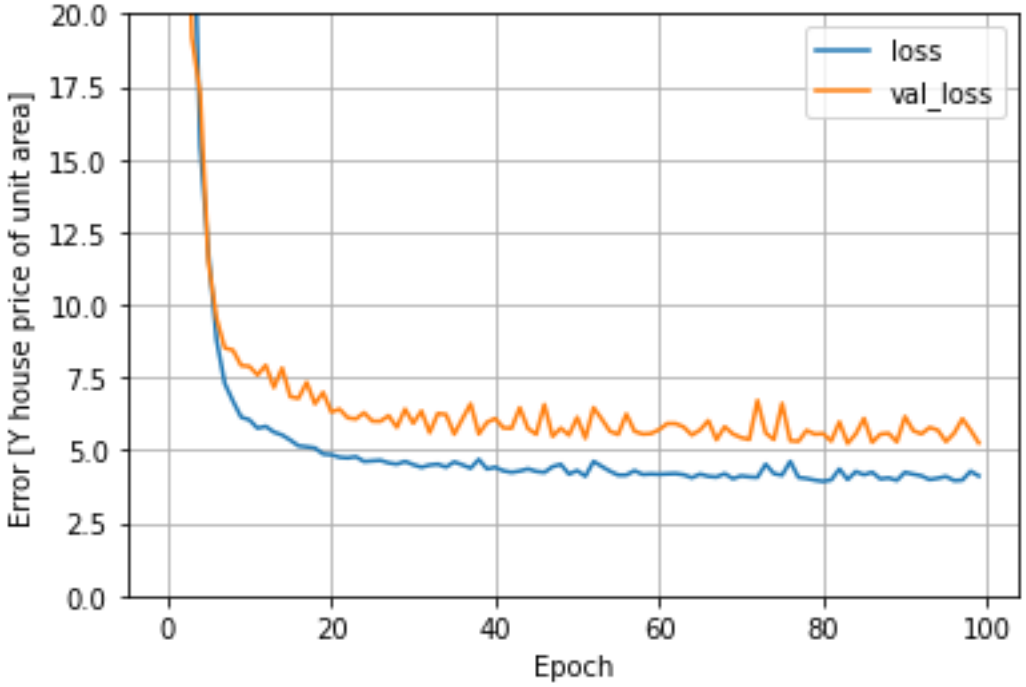


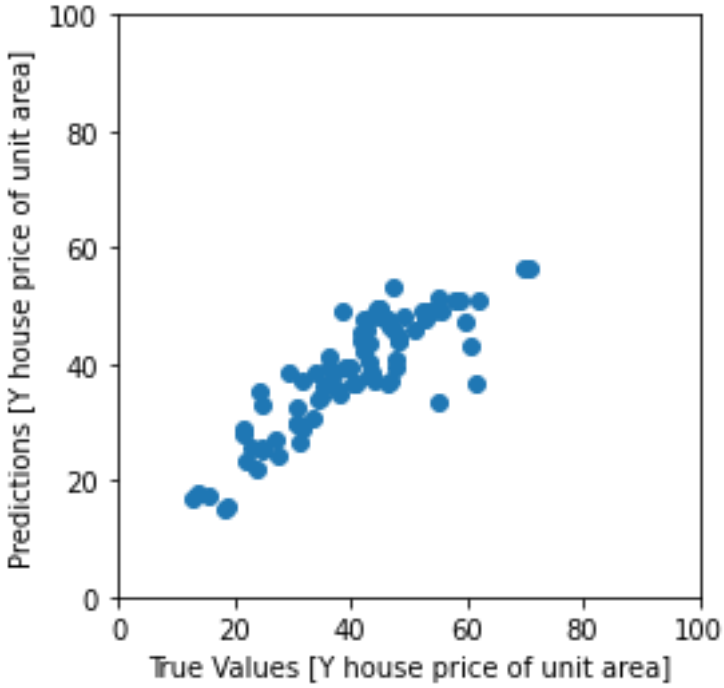


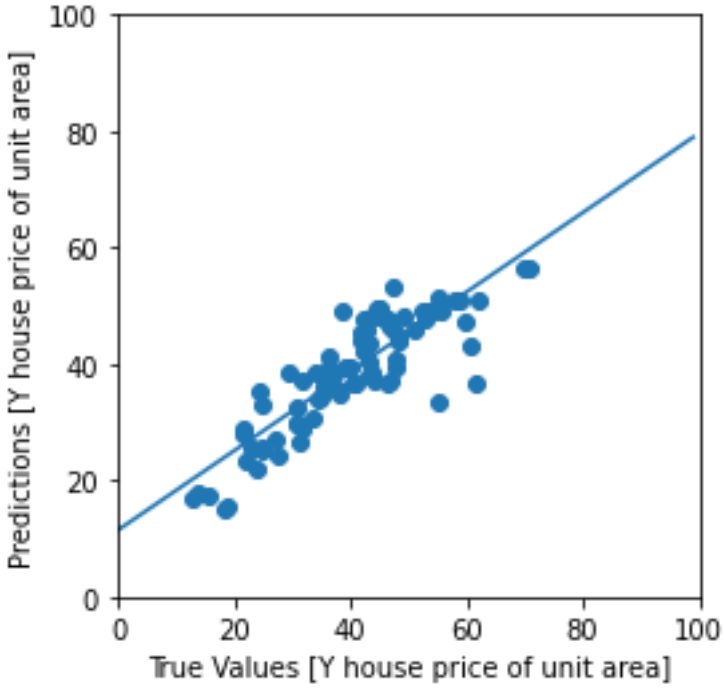

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
```

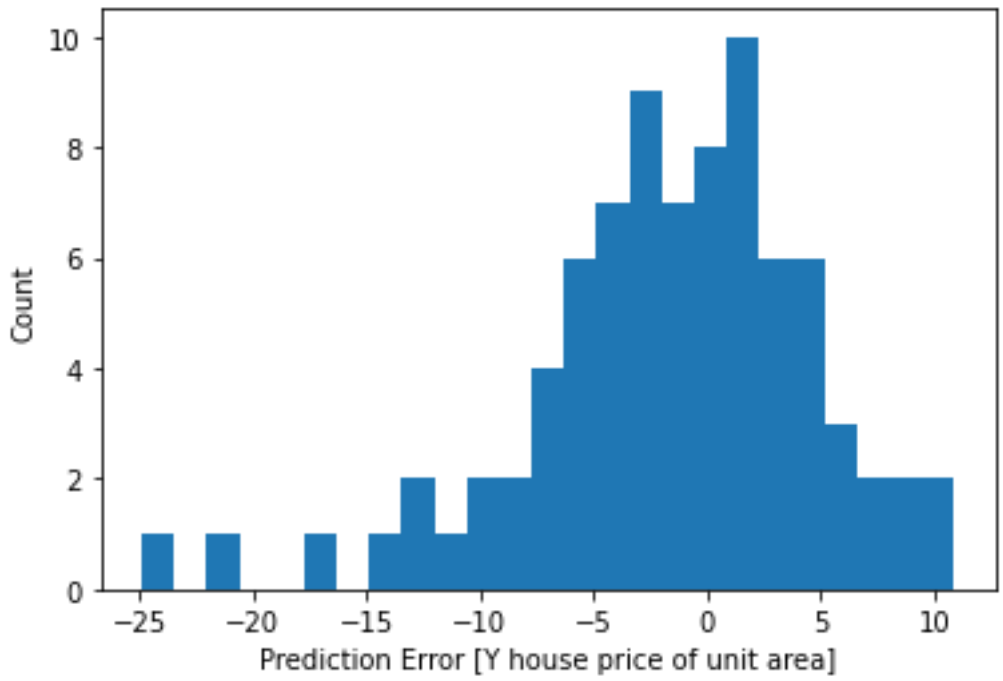
```
    layers.Dense(64, activation='relu'),
    layers.Dense(1)
])

model.compile(loss='mean_absolute_error',
              optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
return model
```

RESULTS

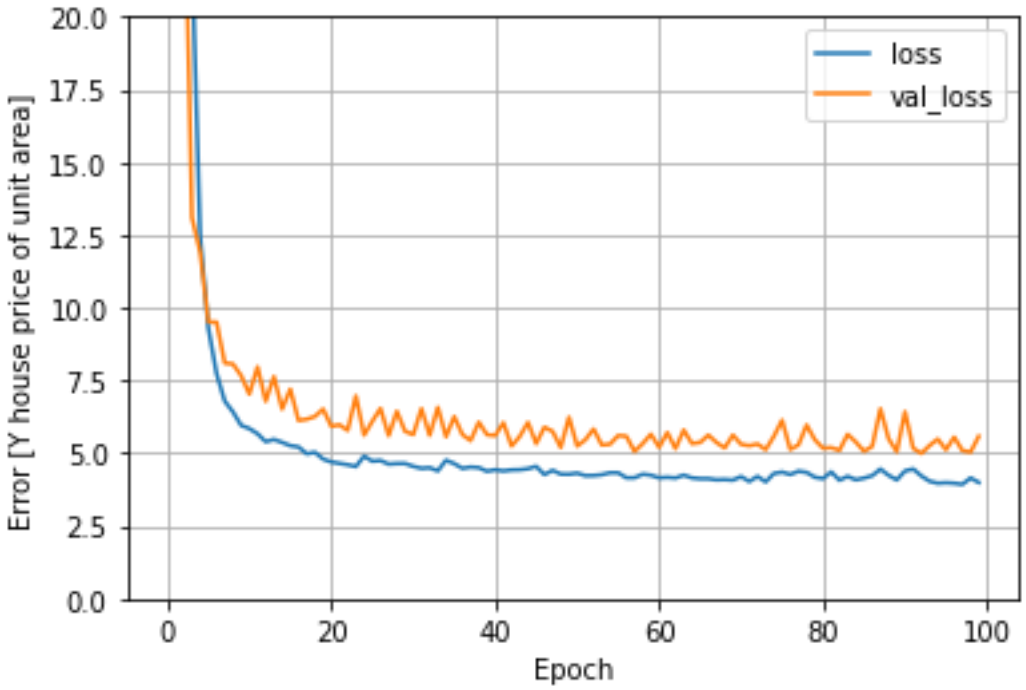


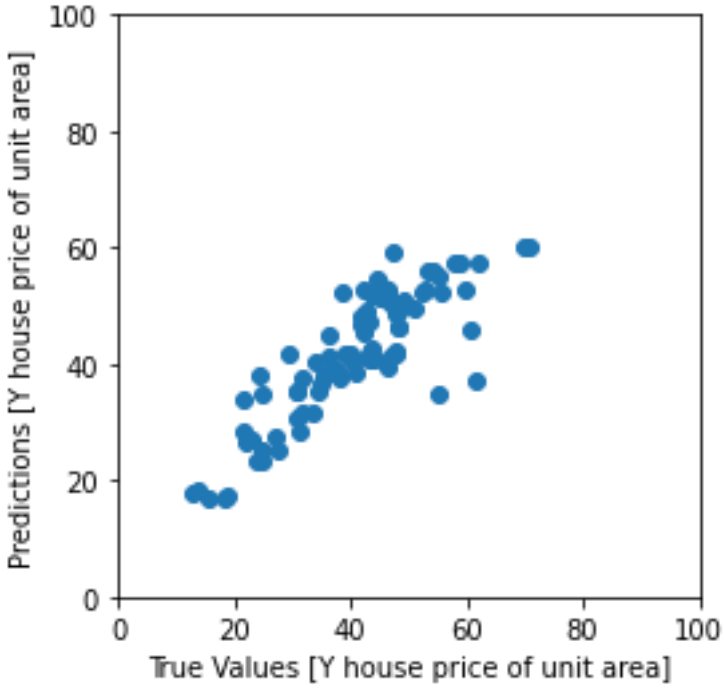

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

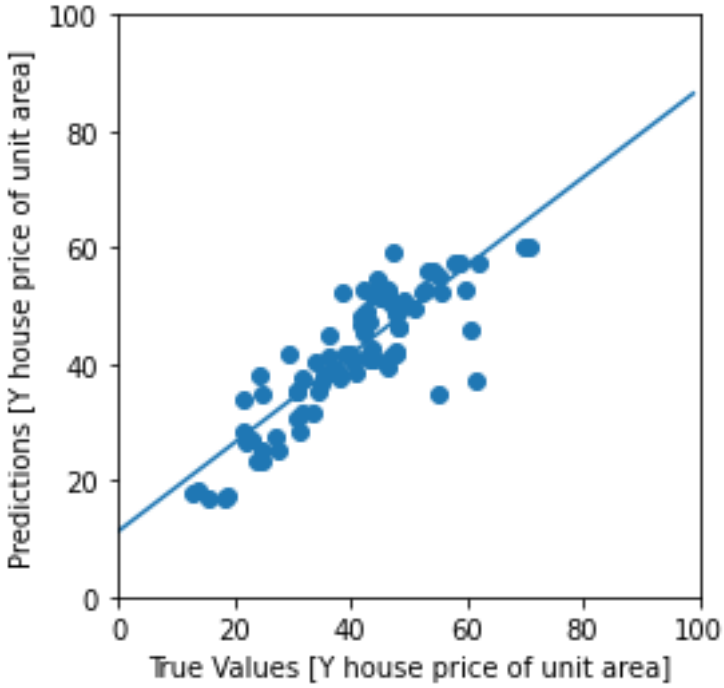

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

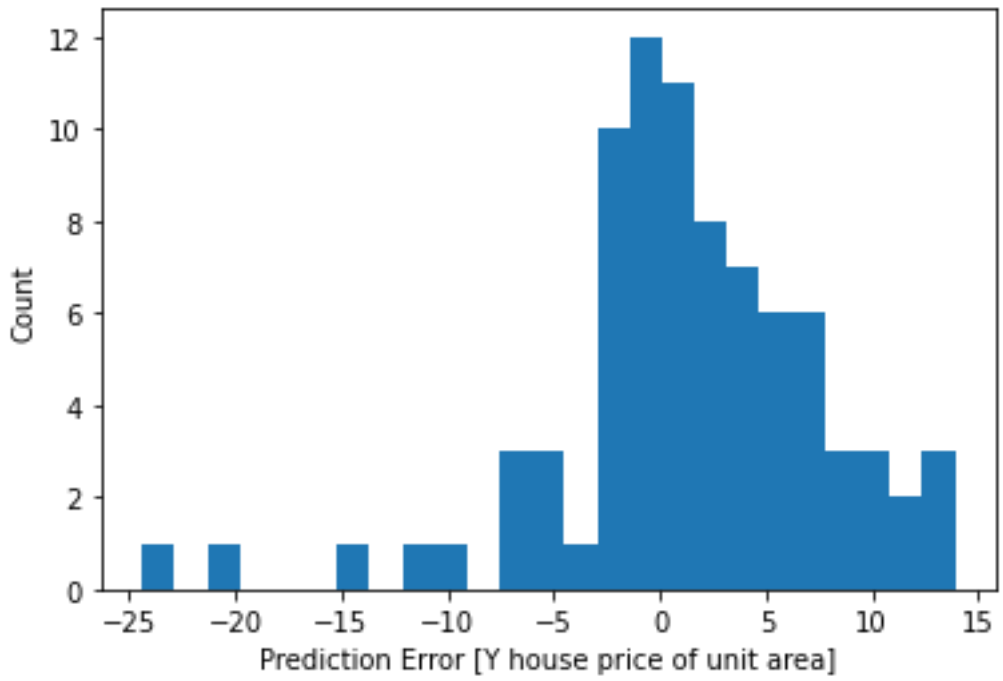



```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```


loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100

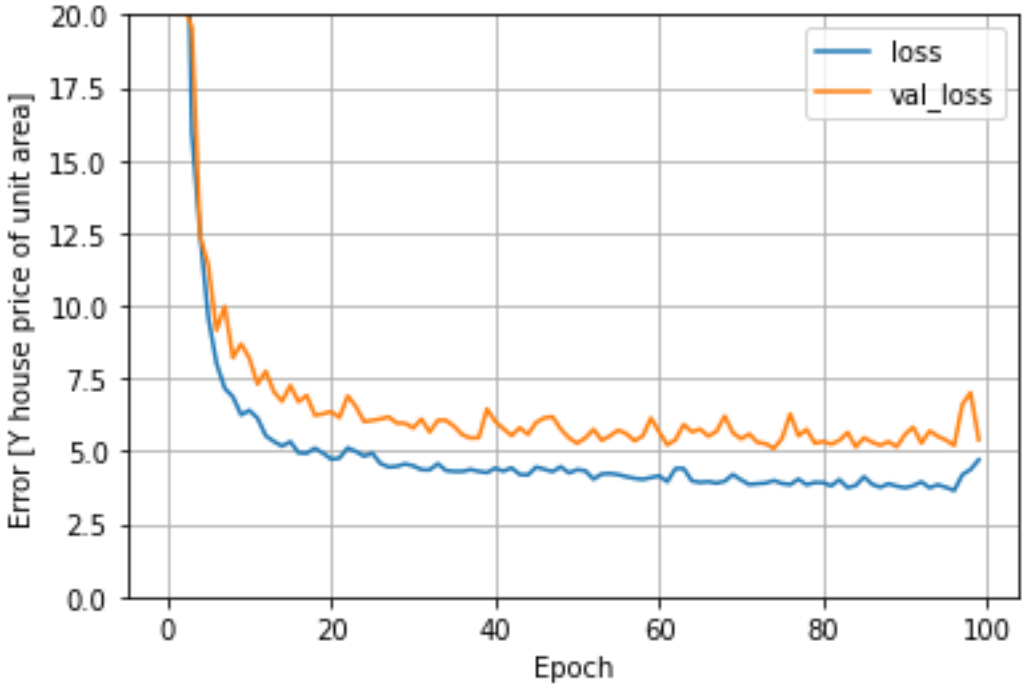

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

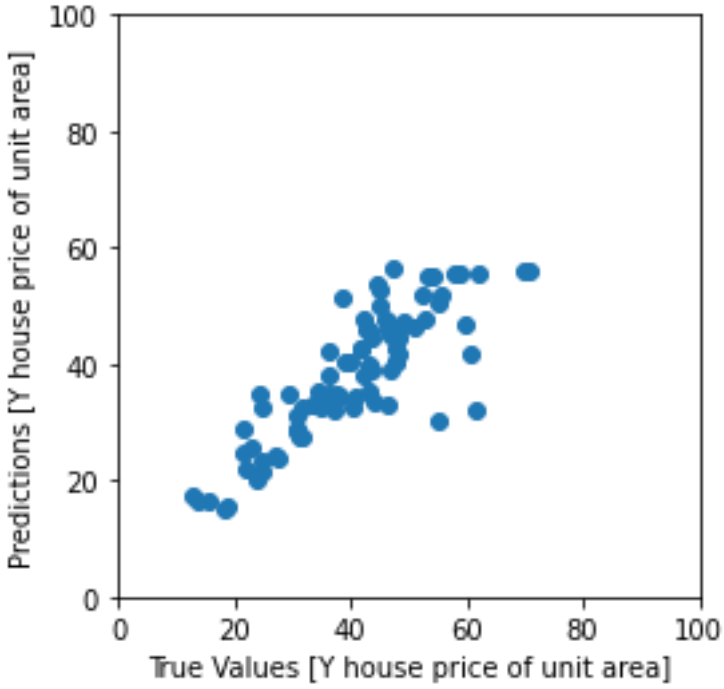

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

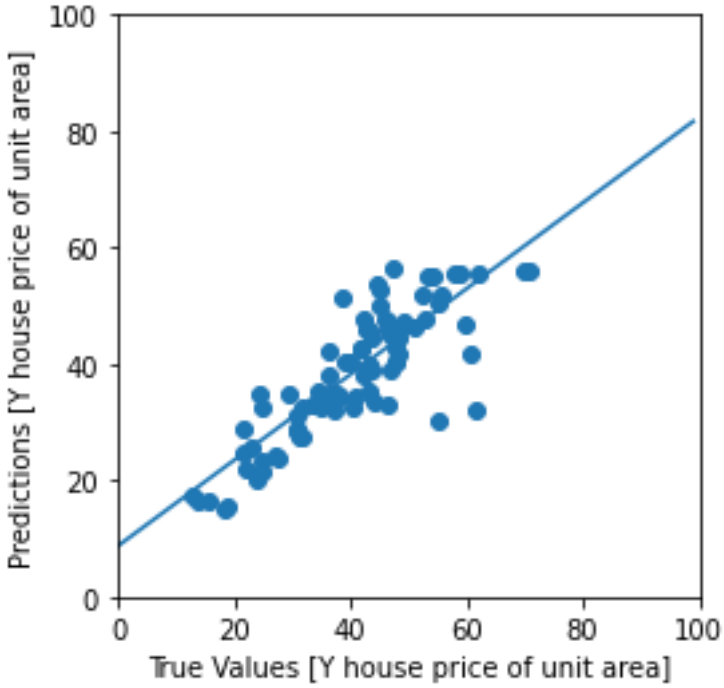


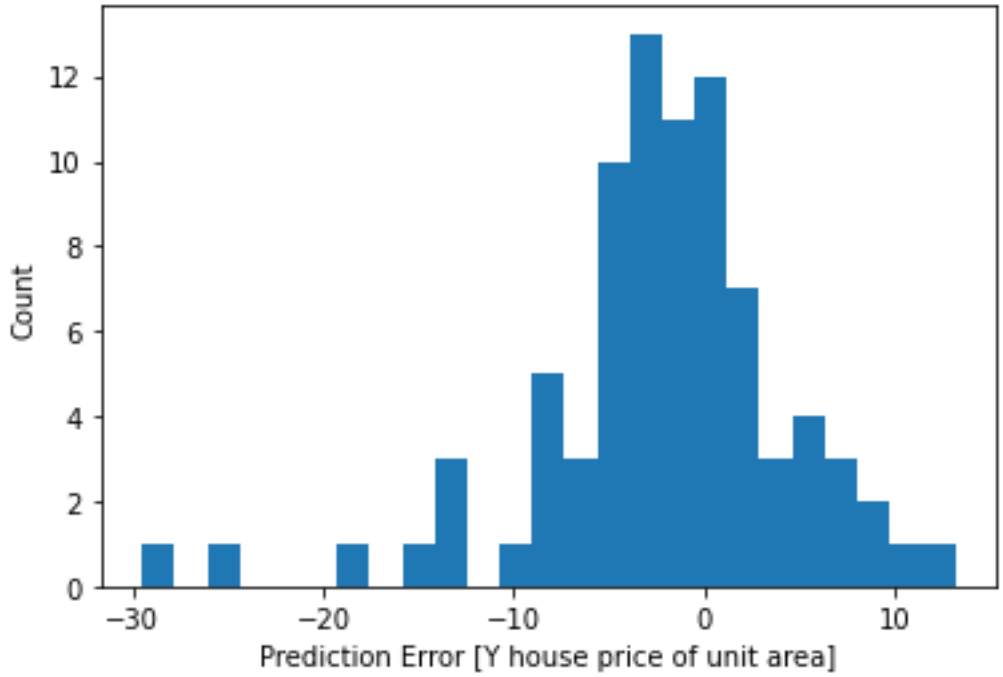

```
def build_and_compile_model(norm):
  model = keras.Sequential([
      norm,
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(64, activation='relu'),
      layers.Dense(1)
  ])

  model.compile(loss='mean_absolute_error',
                optimizer=tf.keras.optimizers.Adam(0.001)) #loss = 'mse'
  return model
```

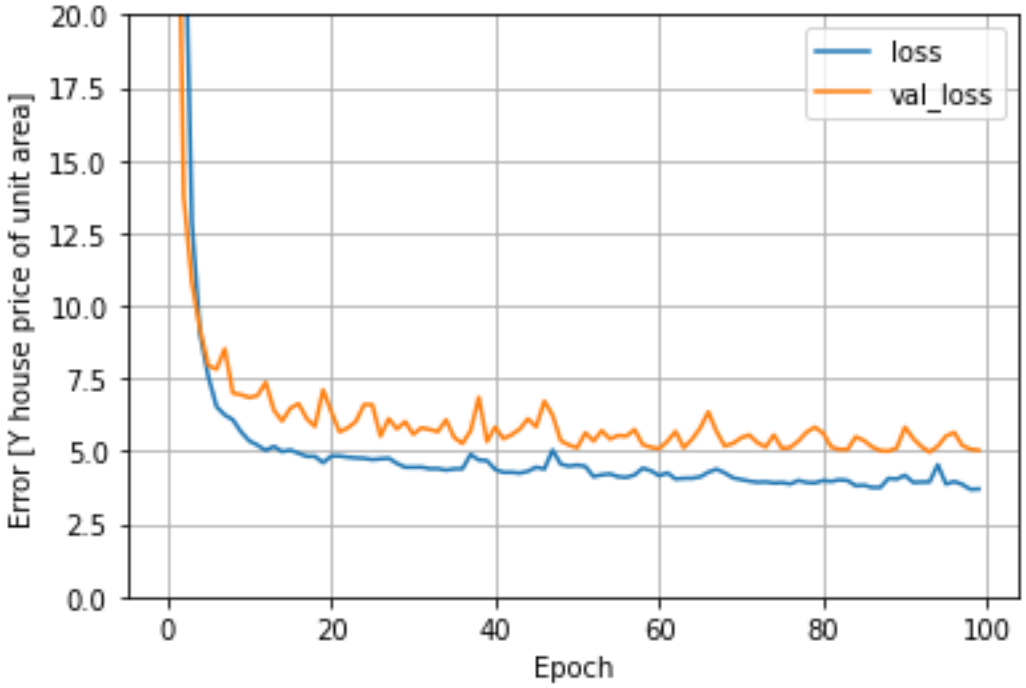
RESULTS

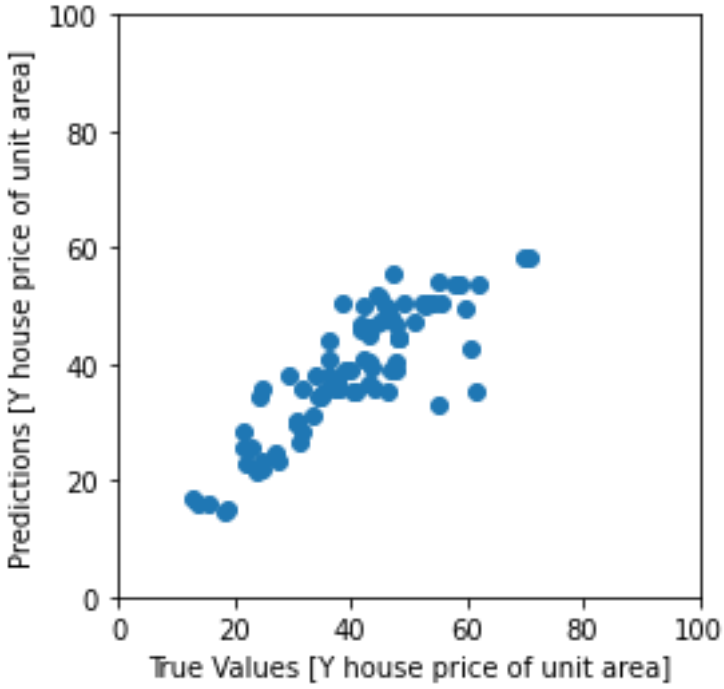

loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100

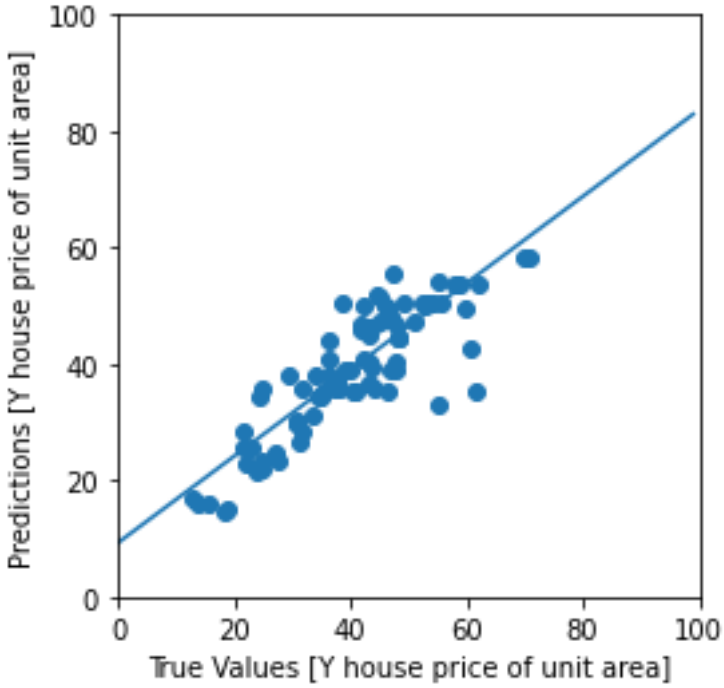

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
100
80
60
40
20
0
0
20
40
60
80
100

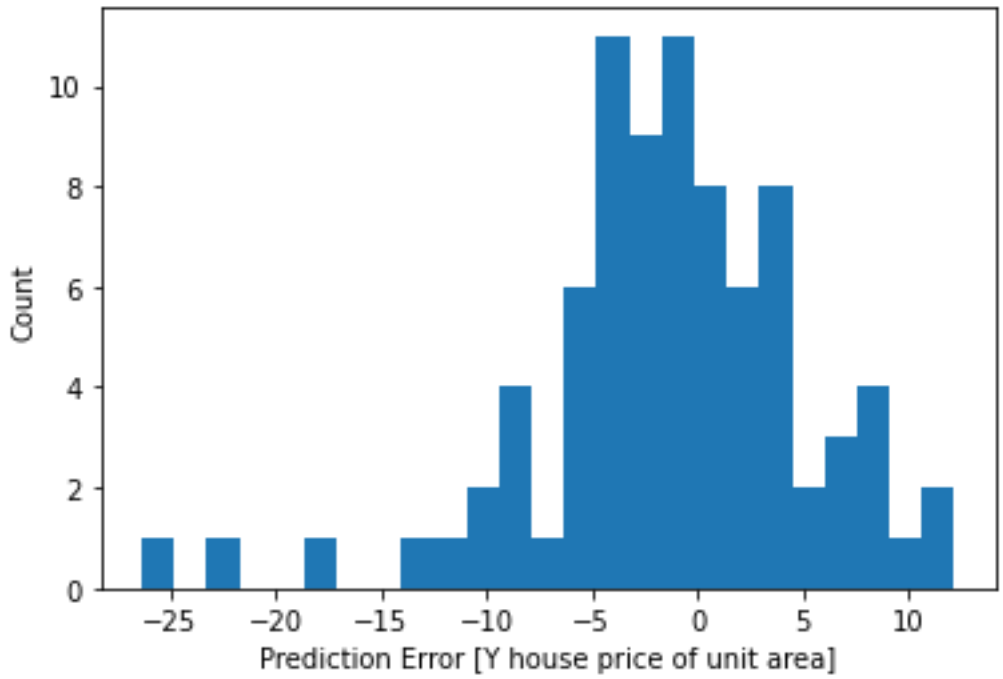

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100

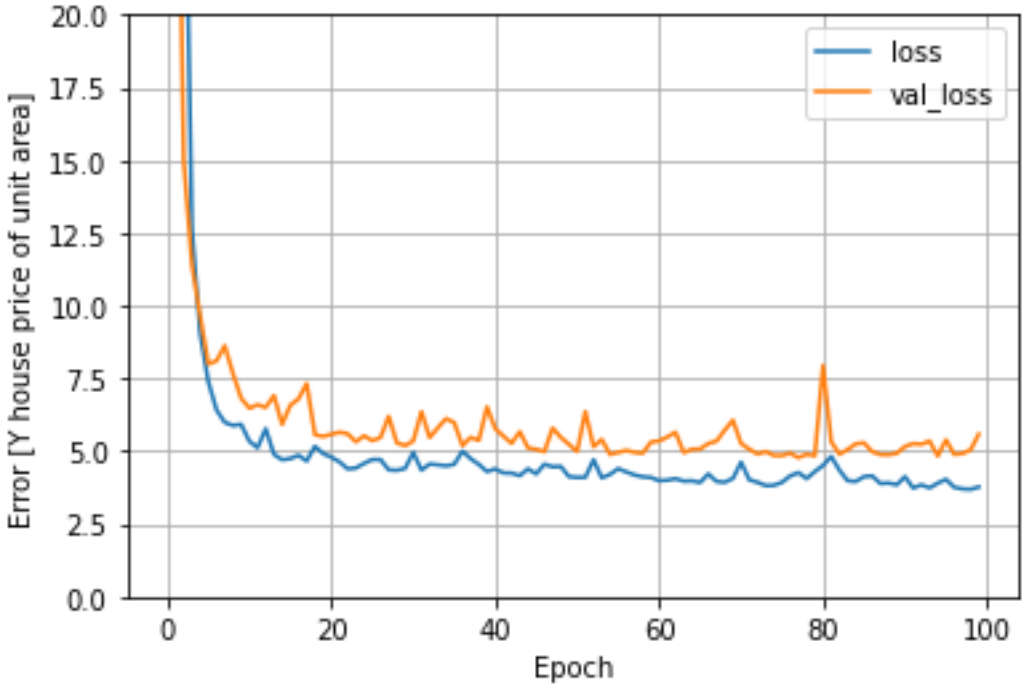

Count
Prediction Error [Y house price of unit area]
0
2
4
6
8
10
12
−30
−20
−10
0
10

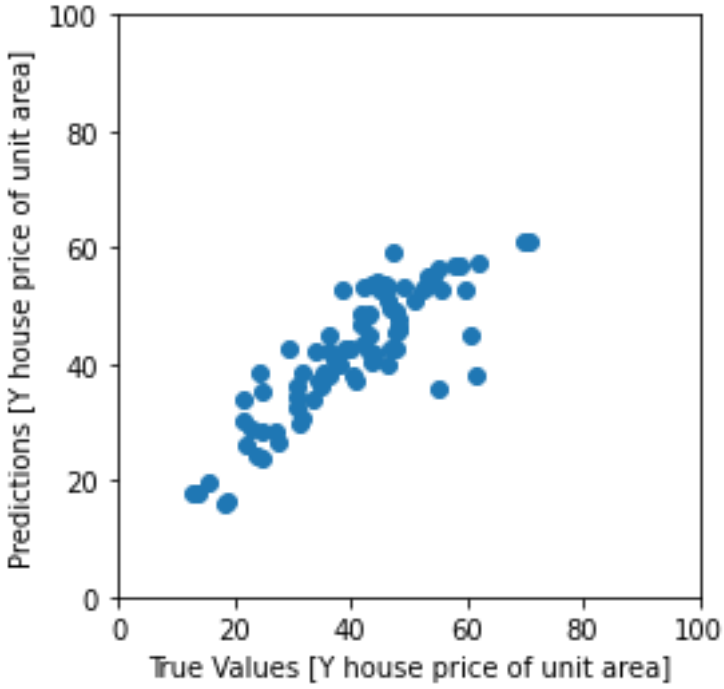

loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100

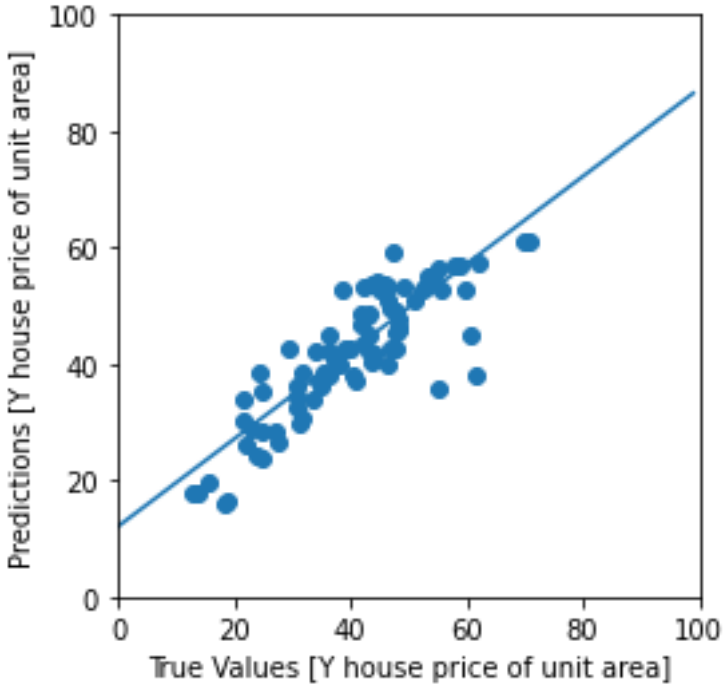

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
100
80
60
40
20
0
0
20
40
60
80
100

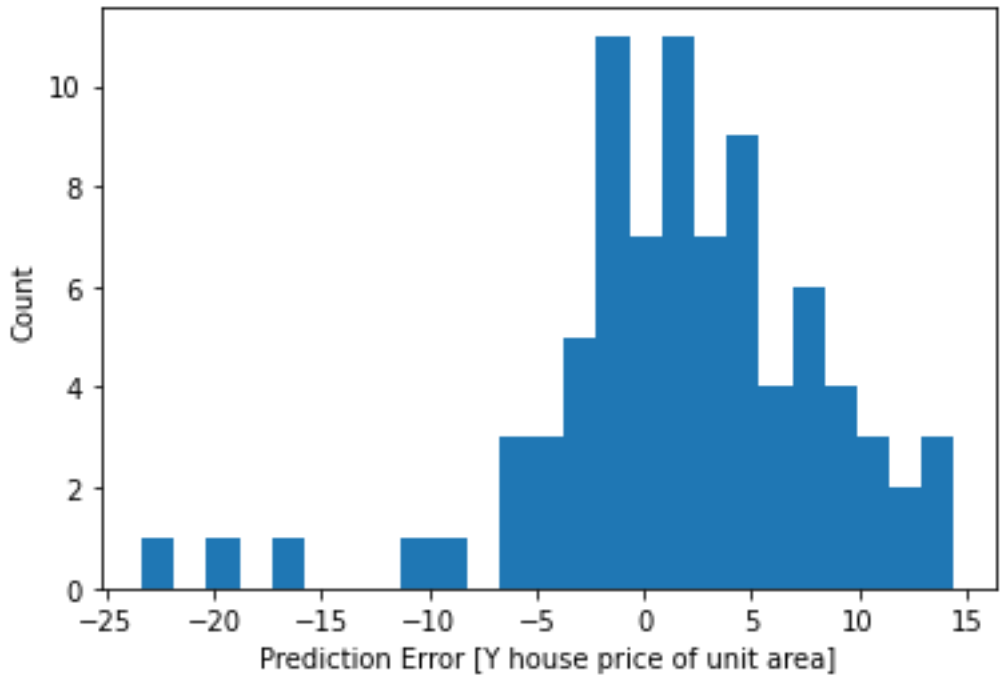

Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100


Count
Prediction Error [Y house price of unit area]
0
2
4
6
8
10
−25
−20
−15
−10
−5
0
5
10


loss
val_loss
Error [Y house price of unit area]
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100


Predictions [Y house price of unit area]
True Values [Y house price of unit area]
100
80
60
40
20
0
0
20
40
60
80
100


Predictions [Y house price of unit area]
True Values [Y house price of unit area]
0
20
40
60
80
100


Count
Prediction Error [Y house price of unit area]
0
2
4
6
8
10
−25
−20
−15
−10
−5
0
5
10
15