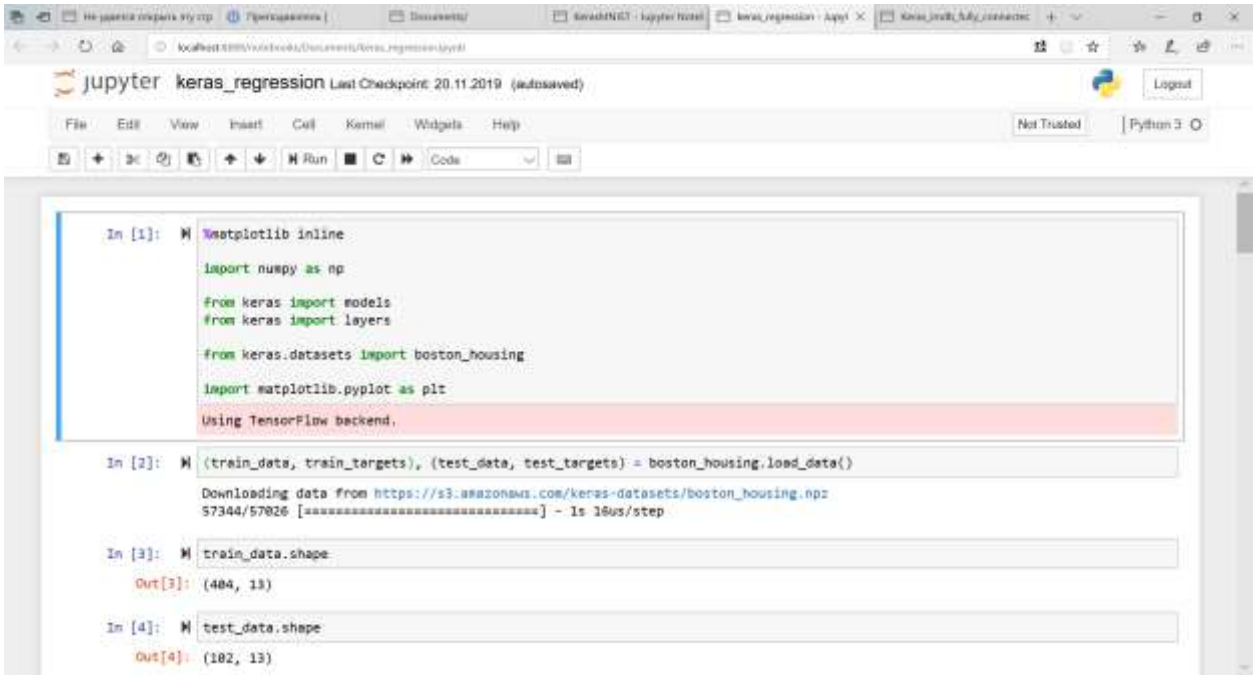
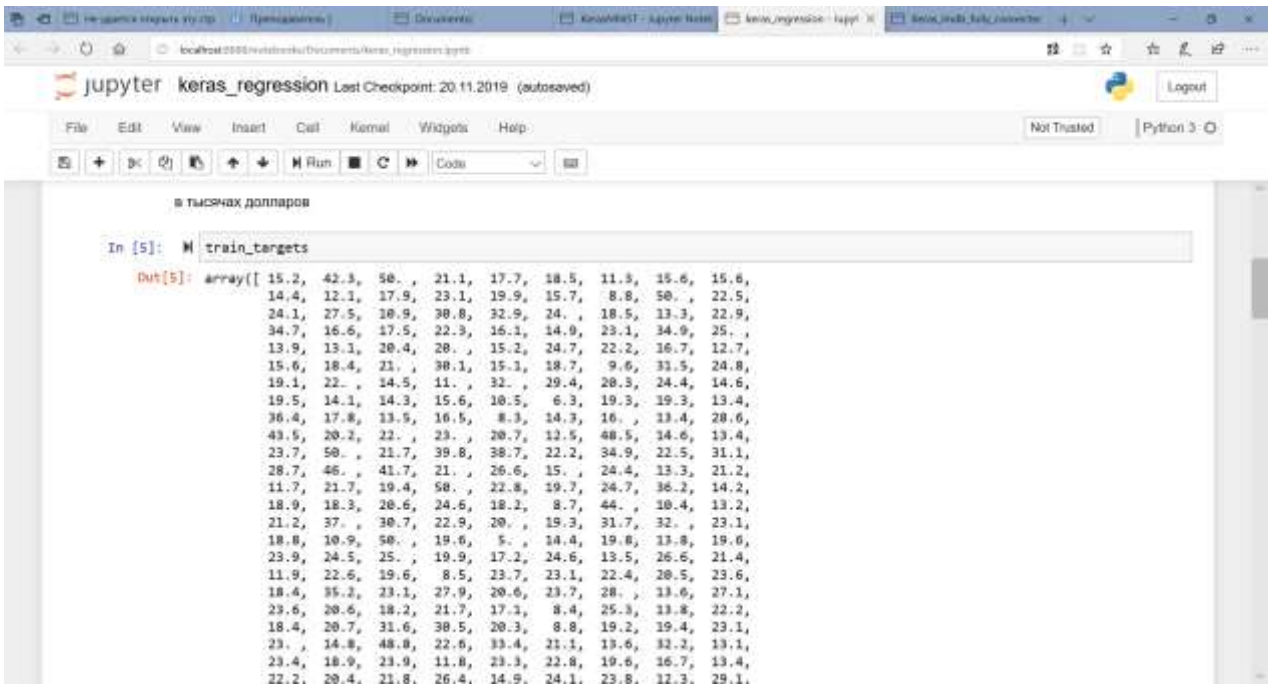
Лабораторное занятие 14. Кросс валидация в временных рядах, линейные модели. (с помощью нейрнных сетей)**1**

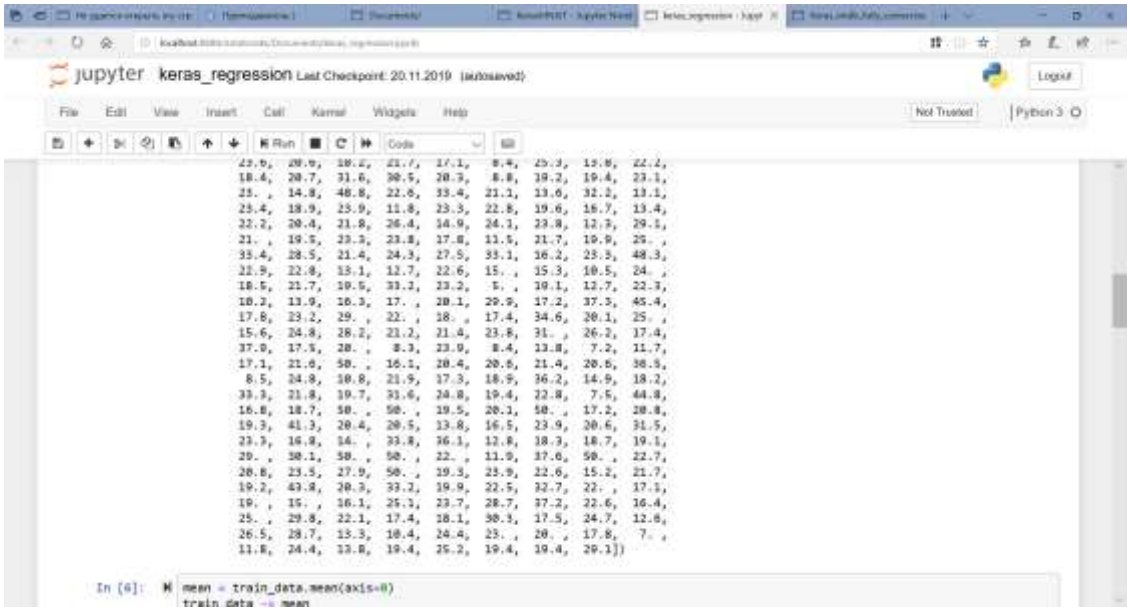
15-зертханалық сабақ. 15-зертханалық сабақ. Уақытқа байланысты қатарларда кросс валидация, сызықты модельдер(нейро желі көмегімен) 15-зертханалық сабақ. Уақытқа байланысты қатарларда кросс валидация, сызықты модельдер

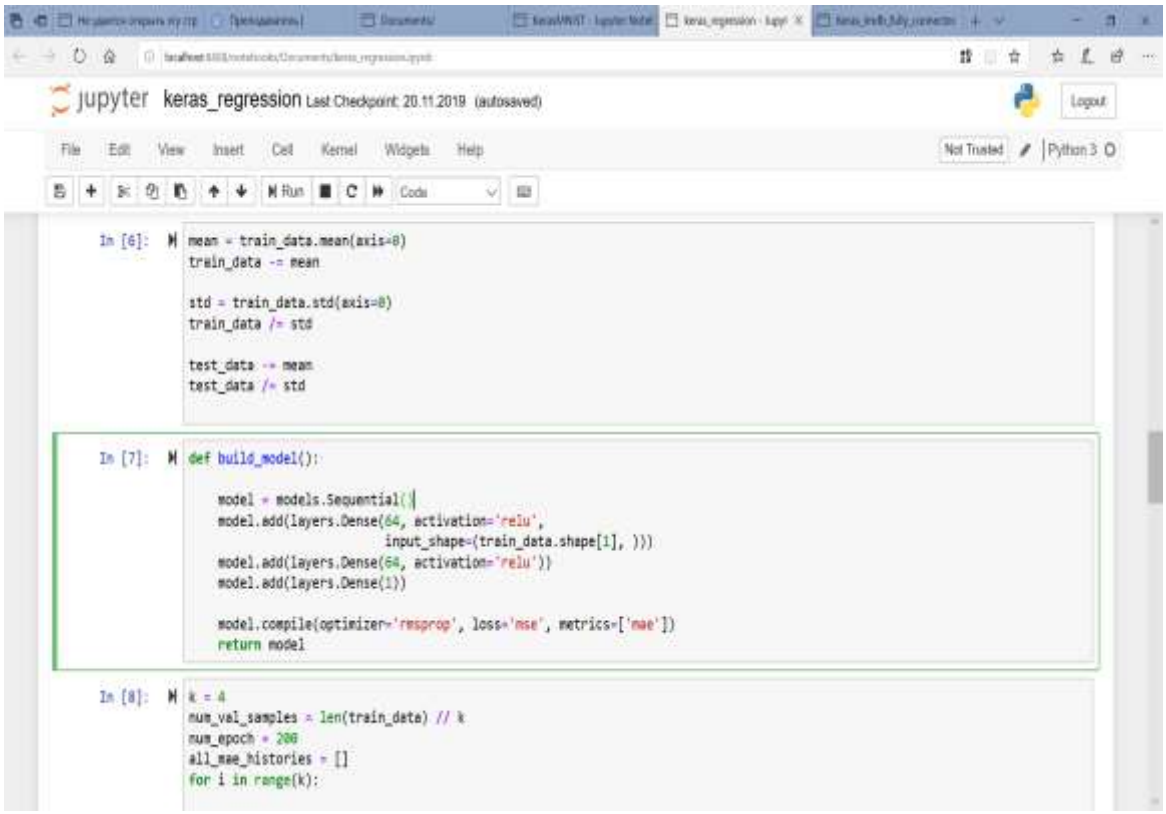
Python және Anaconda да мысал. Сіз өз деректер қорына жасаныз!

Пример на Python и Anaconde. Ниже приведен пример. Каждый должен сделать для своего датасета.






jupyter keras_regression Last Checkpoint: 20.11.2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
23.6, 28.0, 18.2, 21.7, 17.1, 9.4, 25.3, 13.8, 22.2,
18.4, 20.7, 31.6, 30.5, 20.3, 8.8, 19.2, 19.4, 23.1,
23. , 14.8, 48.8, 22.6, 33.4, 21.1, 13.6, 32.2, 13.1,
23.4, 18.9, 23.9, 11.8, 23.3, 22.8, 19.6, 16.7, 13.4,
22.2, 20.4, 21.8, 26.4, 14.9, 24.1, 23.8, 12.3, 29.1,
21. , 19.5, 23.3, 23.8, 17.8, 11.5, 21.7, 19.9, 25. ,
33.4, 28.5, 21.4, 24.3, 27.5, 33.1, 16.2, 23.3, 48.3,
22.9, 22.8, 13.1, 12.7, 22.6, 15. , 15.3, 10.5, 24. ,
18.5, 21.7, 19.5, 33.2, 23.2, 5. , 19.1, 12.7, 22.3,
10.2, 13.9, 16.3, 17. , 20.1, 29.9, 17.2, 37.3, 45.4,
17.8, 23.2, 29. , 22. , 18. , 17.4, 34.6, 20.1, 25. ,
15.6, 24.8, 28.2, 21.2, 21.4, 23.8, 31. , 26.2, 17.4,
37.9, 17.5, 20. , 8.3, 23.9, 8.4, 13.8, 7.2, 11.7,
17.1, 21.6, 50. , 16.1, 20.4, 20.6, 21.4, 20.6, 36.5,
8.5, 24.8, 10.8, 21.9, 17.3, 18.9, 36.2, 14.9, 18.2,
33.3, 21.8, 19.7, 31.6, 24.8, 19.4, 22.8, 7.5, 44.8,
16.8, 18.7, 50. , 50. , 19.5, 20.1, 50. , 17.2, 20.8,
19.3, 41.3, 20.4, 20.5, 13.8, 16.5, 23.9, 20.6, 31.5,
23.3, 16.8, 14. , 33.8, 36.1, 12.8, 18.3, 18.7, 19.1,
29. , 30.1, 50. , 50. , 22. , 11.9, 37.6, 50. , 22.7,
20.8, 23.5, 27.9, 50. , 19.3, 23.9, 22.6, 15.2, 21.7,
19.2, 43.8, 20.3, 33.2, 19.9, 22.5, 32.7, 22. , 17.1,
19. , 15. , 16.1, 25.1, 23.7, 28.7, 37.2, 22.6, 16.4,
25. , 29.8, 22.1, 17.4, 18.1, 30.3, 17.5, 24.7, 12.6,
26.5, 28.7, 13.3, 10.4, 24.4, 23. , 20. , 17.8, 7. ,
11.8, 24.4, 13.8, 19.4, 25.2, 19.4, 19.4, 29.1])
In [6]:
mean = train_data.mean(axis=0)
train_data -= mean


jupyter keras_regression Last Checkpoint: 20.11.2019 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [6]:
mean = train_data.mean(axis=0)
train_data -= mean
std = train_data.std(axis=0)
train_data /= std
test_data -= mean
test_data /= std
In [7]:
def build_model():
model = models.Sequential()
model.add(layers.Dense(64, activation='relu',
input_shape=(train_data.shape[1],)))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(1))
model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
return model
In [8]:
k = 4
num_val_samples = len(train_data) // k
num_epoch = 200
all_mae_histories = []
for i in range(k):

jupyter keras_regression Last Checkpoint: 20.11.2019 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Run Code

```
print("FOLD №", i)

val_data = train_data[i * num_val_samples: (i + 1) * num_val_samples]
val_targets = train_targets[i * num_val_samples:
                            (i + 1) * num_val_samples]

partial_train_data = np.concatenate(
    [
        train_data[: i * num_val_samples],
        train_data[(i + 1) * num_val_samples :]
    ], axis = 0)
partial_train_targets = np.concatenate(
    [
        train_targets[: i * num_val_samples],
        train_targets[(i + 1) * num_val_samples :]
    ], axis = 0)

model = build_model()
history = model.fit(partial_train_data, partial_train_targets,
                    validation_data = (val_data, val_targets),
                    epochs=num_epoch, batch_size=1, verbose=0)

mae_history = history.history['val_mean_absolute_error']
all_mae_histories.append(mae_history)
```

```
FOLD №  0
FOLD №  1
FOLD №  2
FOLD №  3
```

jupyter keras_regression Last Checkpoint: 20.11.2019 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

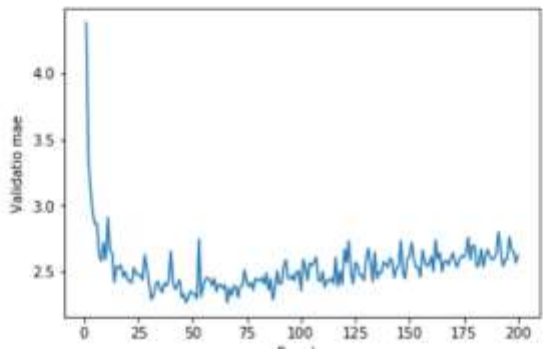
FOLD № 0
FOLD № 1
FOLD № 2
FOLD № 3

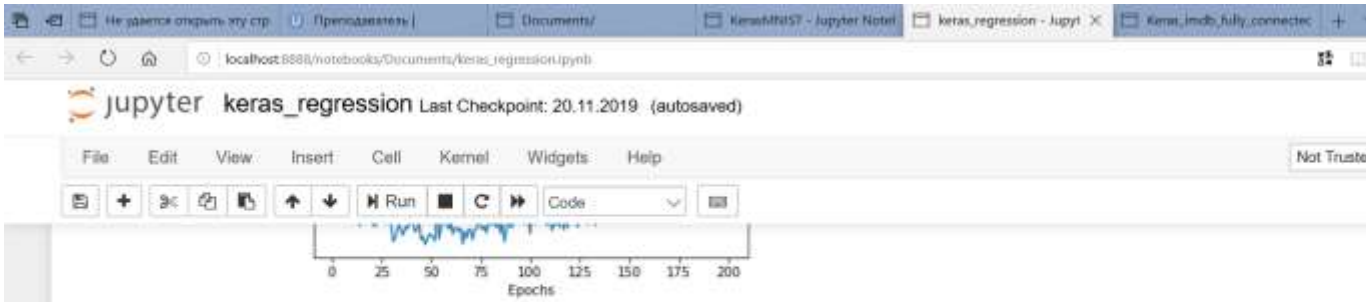
In [9]:

```
average_mae_history = [np.mean([x[i] for x in all_mae_histories])
                       for i in range(num_epoch)]
```

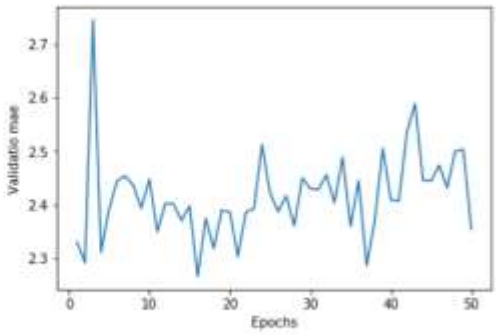
In [10]:

```
plt.plot(range(1, len(average_mae_history) + 1), average_mae_history)
plt.xlabel("Epochs")
plt.ylabel('Validatio mae')
plt.show()
```

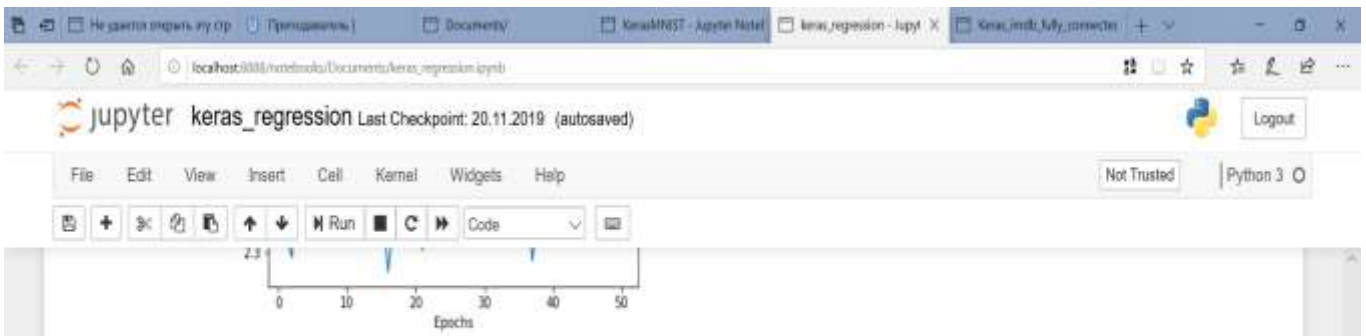




In [13]:
```
plt.plot(range(1, len(average_mae_history[50:100]) + 1), average_mae_history[50:100])
plt.xlabel("Epochs")
plt.ylabel('Validatio mae')
plt.show()
```

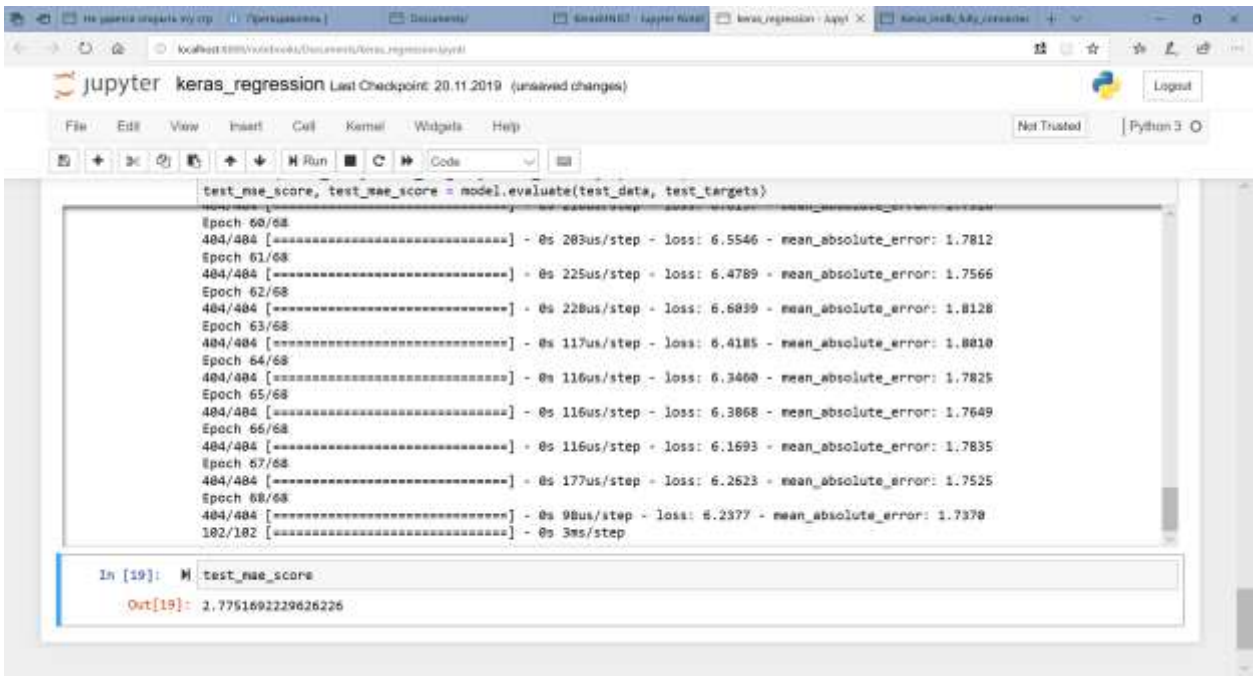


In [18]:
```
model = build_model()
model.fit(train_data, train_targets, batch_size=16, epochs=68)
```



In [18]:
```
model = build_model()
model.fit(train_data, train_targets, batch_size=16, epochs=68)
test_mse_score, test_mae_score = model.evaluate(test_data, test_targets)
```

```
Epoch 1/68
404/404 [==============================] - 1s 3ms/step - loss: 507.7415 - mean_absolute_error: 20.5262
Epoch 2/68
404/404 [==============================] - 0s 517us/step - loss: 342.2839 - mean_absolute_error: 16.2195
Epoch 3/68
404/404 [==============================] - 0s 470us/step - loss: 160.4718 - mean_absolute_error: 10.2928
Epoch 4/68
404/404 [==============================] - 0s 696us/step - loss: 63.7542 - mean_absolute_error: 5.8373
Epoch 5/68
404/404 [==============================] - 0s 1ms/step - loss: 37.2715 - mean_absolute_error: 4.4180
Epoch 6/68
404/404 [==============================] - 0s 866us/step - loss: 27.1400 - mean_absolute_error: 3.74450s - loss: 30.4682
- mean_absolute_error:
Epoch 7/68
404/404 [==============================] - 0s 619us/step - loss: 22.0638 - mean_absolute_error: 3.3515
Epoch 8/68
404/404 [==============================] - 0s 488us/step - loss: 19.2295 - mean_absolute_error: 3.08710s - loss: 22.6960
- mean_absolute_error:
Epoch 9/68
404/404 [==============================] - 0s 364us/step - loss: 16.8524 - mean_absolute_error: 2.9179
```


jupyter keras_regression Last Checkpoint: 20.11.2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
test_mse_score, test_mae_score = model.evaluate(test_data, test_targets)
Epoch 60/68
404/404 [==============================] - 0s 203us/step - loss: 6.5546 - mean_absolute_error: 1.7812
Epoch 61/68
404/404 [==============================] - 0s 225us/step - loss: 6.4789 - mean_absolute_error: 1.7566
Epoch 62/68
404/404 [==============================] - 0s 228us/step - loss: 6.6039 - mean_absolute_error: 1.8128
Epoch 63/68
404/404 [==============================] - 0s 117us/step - loss: 6.4185 - mean_absolute_error: 1.8010
Epoch 64/68
404/404 [==============================] - 0s 116us/step - loss: 6.3460 - mean_absolute_error: 1.7825
Epoch 65/68
404/404 [==============================] - 0s 116us/step - loss: 6.3868 - mean_absolute_error: 1.7649
Epoch 66/68
404/404 [==============================] - 0s 116us/step - loss: 6.1693 - mean_absolute_error: 1.7835
Epoch 67/68
404/404 [==============================] - 0s 177us/step - loss: 6.2623 - mean_absolute_error: 1.7525
Epoch 68/68
404/404 [==============================] - 0s 98us/step - loss: 6.2377 - mean_absolute_error: 1.7370
102/102 [==============================] - 0s 3ms/step
In [19]:
test_mae_score
Out[19]:
2.7751692229626226