



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

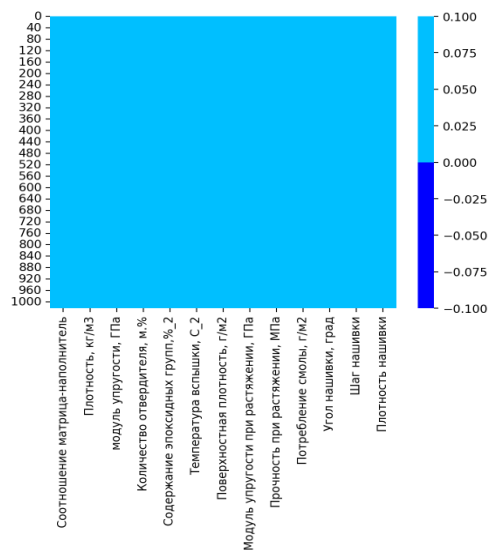
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА по курсу «Data Science»

Прогнозирование конечных свойств новых материалов
(композиционных материалов)

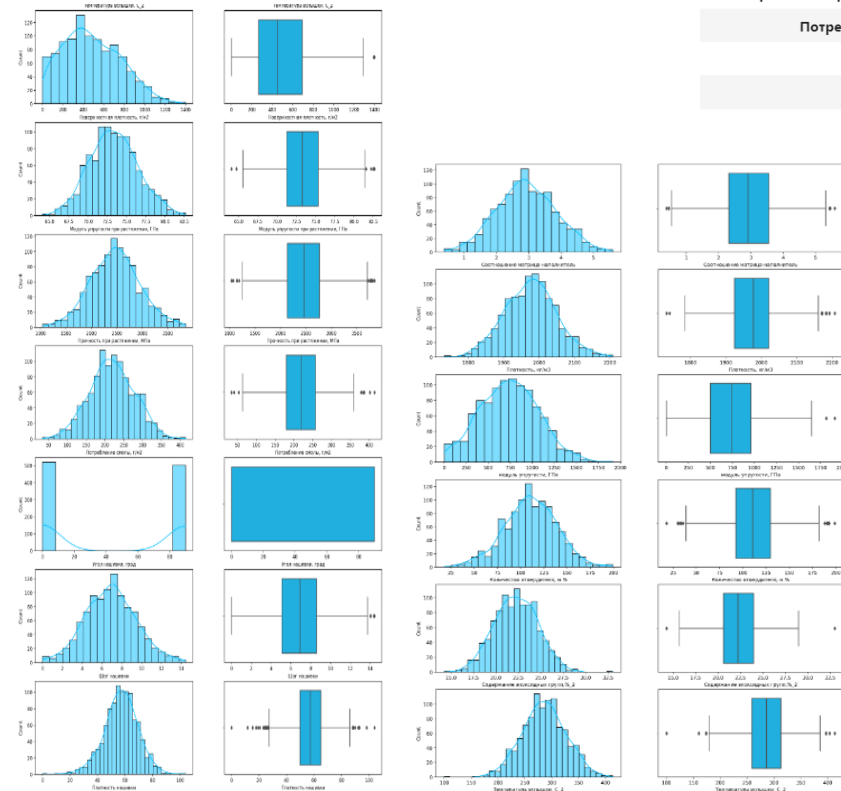
Кузьмина Екатерина Вячеславовна



ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

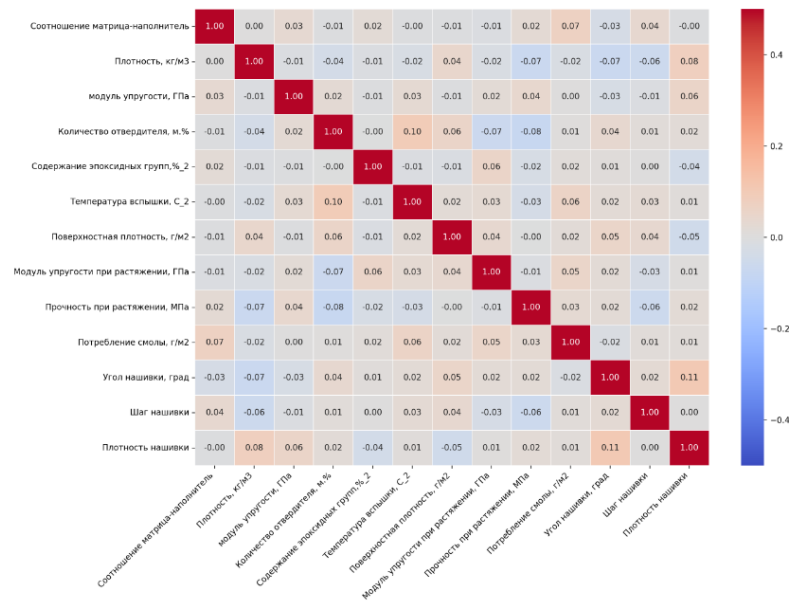


	0	1	2	3	4
Соотношение матрица-наполнитель	1.857143	1.857143	1.857143	1.857143	2.771331
Плотность, кг/м3	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000
модуль упругости, ГПа	738.736842	738.736842	738.736842	738.736842	753.000000
Количество отвердителя, м.%	30.000000	50.000000	49.900000	129.000000	111.860000
Содержание эпоксидных групп, % 2	22.267857	23.750000	33.000000	21.250000	22.267857
Температура вспышки, С 2	100.000000	284.615385	284.615385	300.000000	284.615385
Поверхностная плотность, г/м2	210.000000	210.000000	210.000000	210.000000	210.000000
Модуль упругости при растяжении, ГПа	70.000000	70.000000	70.000000	70.000000	70.000000
Прочность при растяжении, МПа	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000
Потребление смолы, г/м2	220.000000	220.000000	220.000000	220.000000	220.000000
Угол нашивки, град	0.000000	0.000000	0.000000	0.000000	0.000000
Шаг нашивки	4.000000	4.000000	4.000000	5.000000	5.000000
Плотность нашивки	57.000000	60.000000	70.000000	47.000000	57.000000



Разведочный анализ данных

	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп, % 2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С 2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

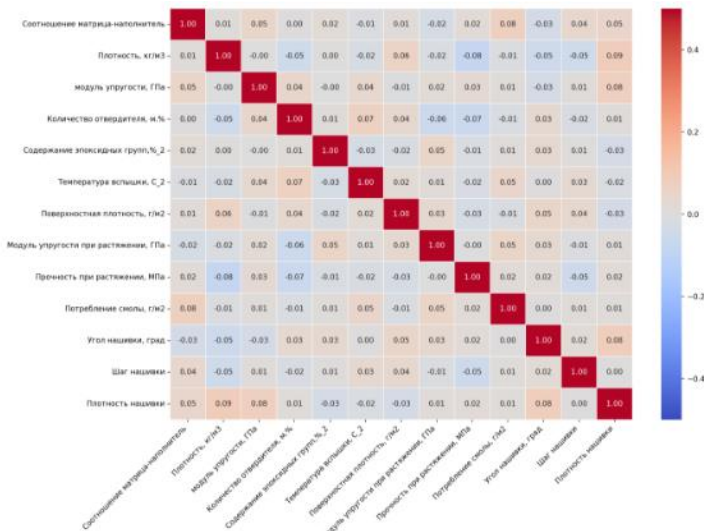
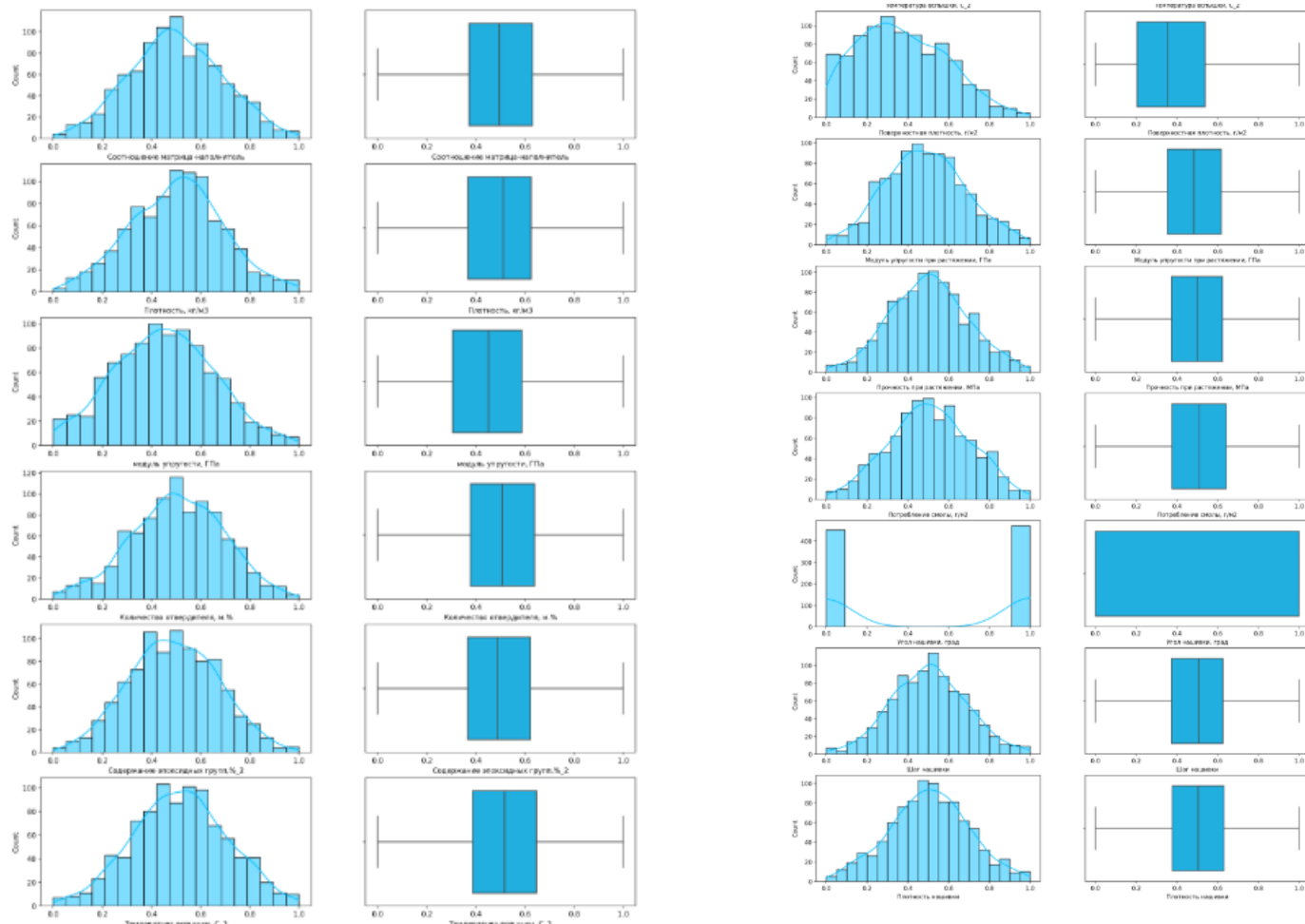




```
q1 = df.quantile(0.25)
q3 = df.quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
outliers = (df < lower_bound) | (df > upper_bound)
outliers_count = outliers.sum()
print(outliers_count)
```

```
scaler = MinMaxScaler()
df=pd.DataFrame(scaler.fit_transform(df),
                columns=df.columns, index=df.index)
```

Предобработка данных



	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	922.0	0.499412	0.167858	0.0	0.371909	0.495189	0.629774	1.0
Плотность, кг/м3	922.0	0.502904	0.188395	0.0	0.368184	0.511396	0.624719	1.0
модуль упругости, ГПа	922.0	0.451341	0.201534	0.0	0.305188	0.451377	0.587193	1.0
Количество отвердителя, м.%	922.0	0.506200	0.186876	0.0	0.378514	0.506382	0.638735	1.0
Содержание эпоксидных групп, %_2	922.0	0.490578	0.180548	0.0	0.366571	0.488852	0.623046	1.0
Температура вспышки, C_2	922.0	0.516739	0.190721	0.0	0.386228	0.516931	0.646553	1.0
Поверхностная плотность, г/м2	922.0	0.373295	0.217269	0.0	0.204335	0.354161	0.538397	1.0
Модуль упругости при растяжении, ГПа	922.0	0.487343	0.196366	0.0	0.353512	0.483718	0.617568	1.0
Прочность при растяжении, МПа	922.0	0.503776	0.188668	0.0	0.373447	0.501481	0.624299	1.0
Потребление смолы, г/м2	922.0	0.507876	0.199418	0.0	0.374647	0.510143	0.642511	1.0
Угол нашивки, град	922.0	0.510846	0.500154	0.0	0.000000	1.000000	1.000000	1.0
Шаг нашивки	922.0	0.503426	0.183587	0.0	0.372844	0.506414	0.626112	1.0
Плотность нашивки	922.0	0.503938	0.193933	0.0	0.376869	0.504310	0.630842	1.0



Обучение моделей. Модуль упругости при растяжении

Lasso

Модуль упругости при растяжении
Лучшие найденные параметры: Lasso(alpha=0.005)
MAE: 0.16762
MSE: 0.04215
MAPE: 0.34831
R2: -0.00986

Ridge

Модуль упругости при растяжении
Лучшие найденные параметры: Ridge(alpha=5)
MAE: 0.16915
MSE: 0.04263
MAPE: 0.35258
R2: -0.02151

LinearRegression

Модуль упругости при растяжении
Лучшие найденные параметры: LinearRegression(fit_intercept='True')
MAE: 0.16944
MSE: 0.04276
MAPE: 0.35336
R2: -0.02448

RandomForestRegressor

Модуль упругости при растяжении
Лучшие найденные параметры: RandomForestRegressor(max_depth=3, max_features='sqrt', n_estimators=40)
MAE: 0.1681
MSE: 0.04224
MAPE: 0.34843
R2: -0.01212

GradientBoostingRegressor

Модуль упругости при растяжении
Лучшие найденные параметры: GradientBoostingRegressor(learning_rate=0.001, max_depth=1, max_features='sqrt')
MAE: 0.16751
MSE: 0.04213
MAPE: 0.34786
R2: -0.00934



Обучение моделей. Прочность при растяжении

Lasso

Прочность при растяжении
Лучшие найденные параметры: Lasso(alpha=0.005)
MAE: 0.15344
MSE: 0.03704
MAPE: 0.30484
R2: -6e-05

Ridge

Прочность при растяжении
Лучшие найденные параметры: Ridge(alpha=5)
MAE: 0.15432
MSE: 0.03709
MAPE: 0.30862
R2: -0.00141

LinearRegression

Прочность при растяжении
Лучшие найденные параметры: LinearRegression(fit_intercept='True')
MAE: 0.15476
MSE: 0.03717
MAPE: 0.31005
R2: -0.01159

RandomForestRegressor

Прочность при растяжении
Лучшие найденные параметры: RandomForestRegressor(max_depth=4, max_features='sqrt', n_estimators=20)
MAE: 0.1542
MSE: 0.03702
MAPE: 0.30865
R2: 0.00038

GradientBoostingRegressor

Прочность при растяжении
Лучшие найденные параметры: GradientBoostingRegressor(learning_rate=0.004, max_depth=9, max_features='sqrt')
MAE: 0.15449
MSE: 0.0376
MAPE: 0.32082
R2: 0.00428



Нейронная сеть

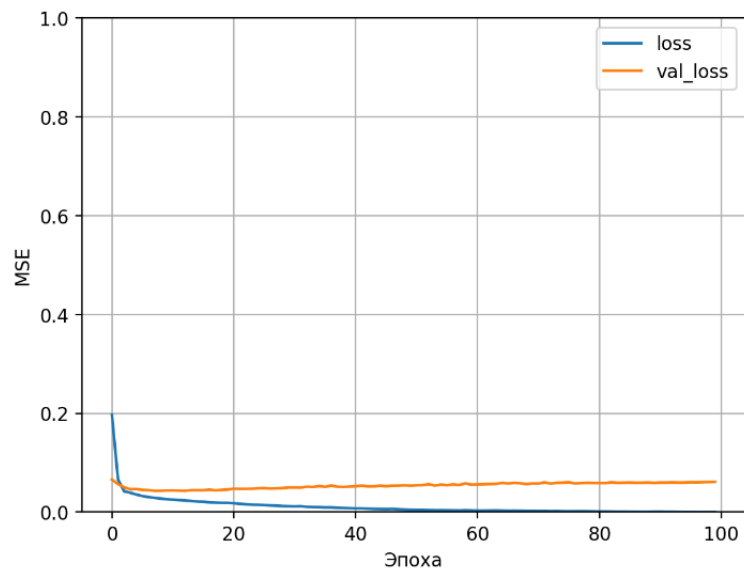
```
nn_model = Sequential([
    normalizer,
    Dense(128, activation = 'relu'),
    Dense(64, activation = 'relu'),
    Dense(32, activation = 'relu'),
    Dense(1)
])

nn_model.compile(loss = 'mean_squared_error', optimizer = tf.keras.optimizers.Adam(0.0005))
```

```
nn_model.summary()
```

Model: "sequential_4"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	3
dense_15 (Dense)	(None, 128)	1664
dense_16 (Dense)	(None, 64)	8256
dense_17 (Dense)	(None, 32)	2080
dense_18 (Dense)	(None, 1)	33
Total params: 12,036		
Trainable params: 12,033		
Non-trainable params: 3		



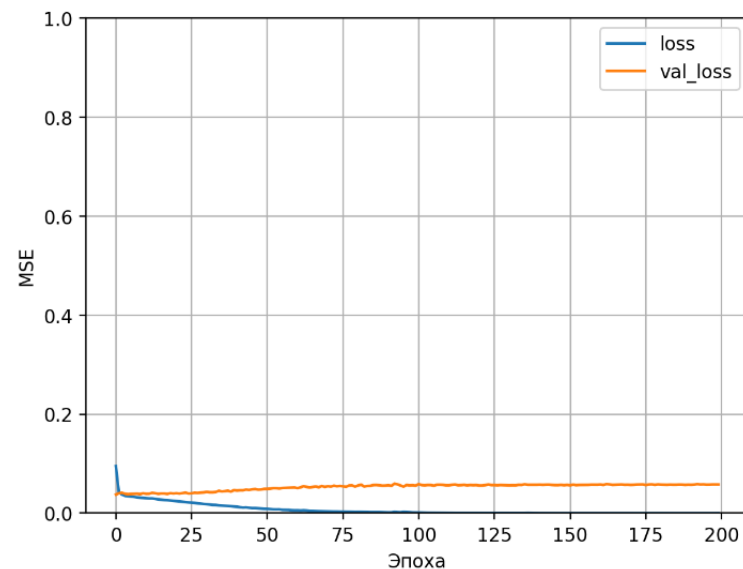
```
nn_model = Sequential([
    normalizer,
    Dense(128, activation = 'relu'),
    Dense(64, activation = 'relu'),
    Dense(32, activation = 'sigmoid'),
    Dense(1)
])

nn_model.compile(loss = 'mean_squared_error', optimizer = tf.keras.optimizers.Adam(0.0005))
```

```
nn_model.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	3
dense_19 (Dense)	(None, 128)	1664
dense_20 (Dense)	(None, 64)	8256
dense_21 (Dense)	(None, 32)	2080
dense_22 (Dense)	(None, 1)	33
Total params: 12,036		
Trainable params: 12,033		
Non-trainable params: 3		





Нейронная сеть

```
nn_model = Sequential([
    normalizer,
    Dense(8, activation = 'relu'),
    Dense(8, activation = 'relu'),
    Dense(1)
])

nn_model.compile(loss = 'mean_squared_error', optimizer = tf.keras.optimizers.Adam(0.0005))

nn_model.summary()
```

Model: "sequential_6"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	3
dense_23 (Dense)	(None, 8)	104
dense_24 (Dense)	(None, 8)	72
dense_25 (Dense)	(None, 1)	9
Total params: 188		
Trainable params: 185		
Non-trainable params: 3		

```
nn_model = Sequential([
    normalizer,
    Dense(8, activation = 'relu'),
    Dense(8, activation = 'selu'),
    Dense(1)
])

nn_model.compile(loss = 'mean_squared_error', optimizer = tf.keras.optimizers.Adam(0.0005))

nn_model.summary()
```

Model: "sequential_7"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	3
dense_26 (Dense)	(None, 8)	104
dense_27 (Dense)	(None, 8)	72
dense_28 (Dense)	(None, 1)	9
Total params: 188		
Trainable params: 185		
Non-trainable params: 3		

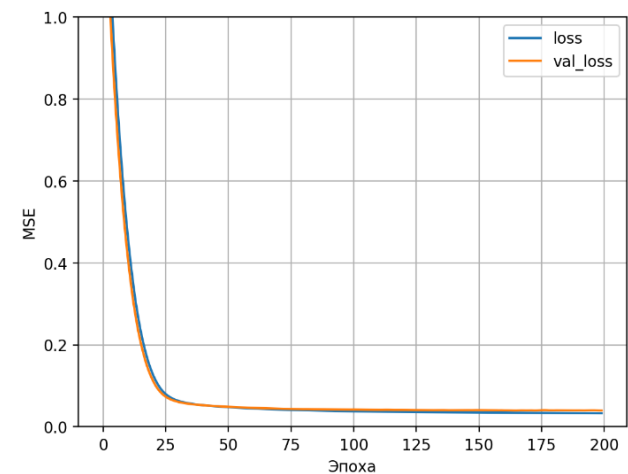
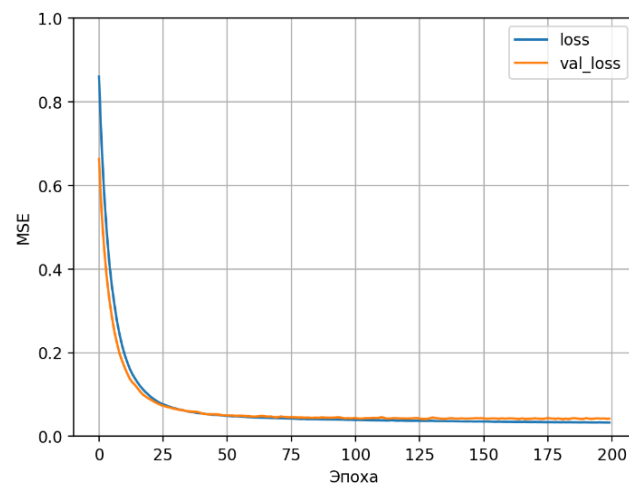
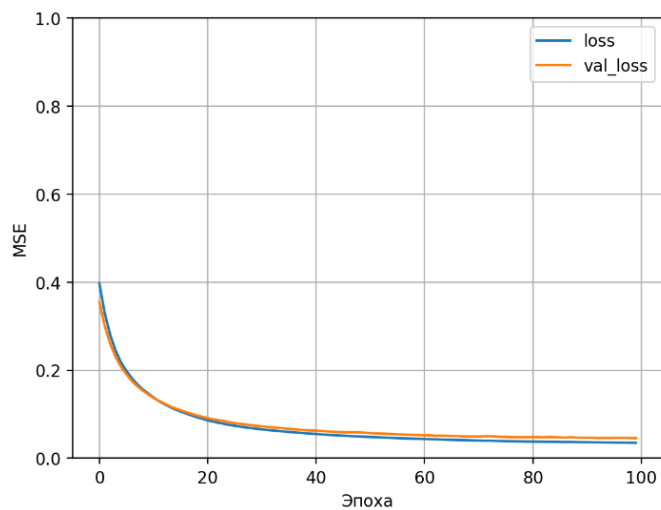
```
nn_model = Sequential([
    normalizer,
    Dense(8, activation = 'relu'),
    Dense(8, activation = 'sigmoid'),
    Dense(1)
])

nn_model.compile(loss = 'mean_squared_error', optimizer = tf.keras.optimizers.Adam(0.0005))

nn_model.summary()
```

Model: "sequential_9"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 12)	3
dense_32 (Dense)	(None, 8)	104
dense_33 (Dense)	(None, 8)	72
dense_34 (Dense)	(None, 1)	9
Total params: 188		
Trainable params: 185		
Non-trainable params: 3		





ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана



do.bmstu.ru