РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

ОТЧЕТ ПО КОНТРОЛЬНОЙ РАБОТЕ № 2

Дисциплина: Методы машинного обучения

Студент: Петров Артем Евгеньевич

Группа: НКНбд-01-21

Москва 2024

Контрольная работа 2 – Вариант 10

1. Набор данных: cherry_blossoms
2. Независимая переменная: temp
3. Зависимая переменная: temp_lower
4. Доп. признак: имеющий минимальную корреляцию с независимой переменной
5. Визуализация доп. признака – эмпирическая плотность распределения
6. Показатель качества регрессии – MSE (mean squared error)
7. Степень полинома: 3
8. Параметры глубокой нейронной сети: кол-во скрытых слоев – 3, кол-во нейронов в скрытом слое – 128, функция активации – сигмоида.

1. Загрузите заданный в индивидуальном задании набор данных из Tensorflow Datasets, включая указанные в задании независимый признак и зависимый признак (отклик). Оставьте в наборе признаки, принимающие числовые значения.
2. Удалите из набора точки с выбросами при помощи стандартизованной оценки (Z-score) таким образом, чтобы точки с выбросами составляли от 5% до 10% всех точек набора данных. Визуализируйте точки исходного набора данных на плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y –

зависимый признак), показывая оставленные в наборе точки и удаленные точки разными цветами, подписывая оси и рисунок и создавая легенду.

3. Выполните стандартизацию независимого признака и масштабирование на интервал [-1, 1] зависимого признака. Решите задачи линейной регрессии и полиномиальной регрессии для степени полинома, указанной в индивидуальном задании, при помощи нейронных сетей с одним нейроном и оцените качество полученных моделей по показателю, указанному в индивидуальном задании. Отследите обучение нейронных сетей, изменяя, при необходимости, гиперпараметры (функцию потерь, оптимизатор, шаг обучения и т.п.) или применяя регуляризацию.
4. Постройте кривые обучения для построенных нейронных сетей с зависимостью от количества эпох на одной визуализации. На визуализации создайте легенду.
5. Визуализируйте точки набора данных на плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y – зависимый признак), а также линии линейной и полиномиальной регрессий (другими цветами), подписывая оси и рисунок и создавая легенду.
6. Определите в исходном наборе данных признак (отличный от независимого и зависимого признаков), принимающий непрерывные значения и имеющий свойства, указанные в индивидуальном задании.
7. Стандартизуйте этот признак и визуализируйте его в соответствии с индивидуальным заданием.
8. Сформируйте набор входных данных из двух стандартизованных признаков набора данных (независимый признак и определенный признак), постройте нейронную сеть (нелинейный регресор) с количеством скрытых слоев, количеством нейронов и функцией активации, указанными в индивидуальном задании, и одним нейроном в выходном слое и обучите ее на наборе данных из двух признаков и отклика. Отследите обучение нейронной сети, изменяя, при необходимости, гиперпараметры (функцию потерь, оптимизатор, шаг обучения и т.п.) или применяя регуляризацию.
9. Визуализируйте набор данных в виде диаграммы рассеяния и прогноз нейронной сети в виде поверхности в трехмерном пространстве, подписывая оси и рисунок.
10. Разбейте набор данных из двух признаков и отклика на обучающую и тестовую выборки и постройте кривые обучения для заданного показателя качества в зависимости от количества точек в обучающей выборке, подписывая оси и рисунок и создавая легенду.

Выполнение лабораторной работы:

1. Загрузите заданный в индивидуальном задании набор данных из Tensorflow

Datasets, включая указанные в задании независимый признак и зависимый признак (отклик). Оставьте в наборе признаки, принимающие числовые значения.

In [436…
```
import pandas as pd
```

In [437…
```
df = pd.read_csv('https://raw.githubusercontent.com/rmcelreath/rethinking/master/da
```

In [438…
```
print(df.info())
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1215 entries, 0 to 1214
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   year        1215 non-null   int64  
 1   doy         827 non-null    float64
 2   temp        1124 non-null   float64
 3   temp_upper  1124 non-null   float64
 4   temp_lower  1124 non-null   float64
dtypes: float64(4), int64(1)
memory usage: 47.6 KB
None
```

Out[438…

	year	doy	temp	temp_upper	temp_lower
0	801	NaN	NaN	NaN	NaN
1	802	NaN	NaN	NaN	NaN
2	803	NaN	NaN	NaN	NaN
3	804	NaN	NaN	NaN	NaN
4	805	NaN	NaN	NaN	NaN

In [439…
```
def columns_with_missing_data(df):
    for column in df.columns[df.isnull().any()]:
        print(f"{column:<20}\t{df[column].isnull().mean():.2f}")

print(columns_with_missing_data(df))
```

```
doy                     0.32
temp                    0.07
temp_upper              0.07
temp_lower              0.07
None
```

In [440…
```
df.isna().sum()
```

Out[440…
```
year            0
doy           388
temp           91
temp_upper     91
temp_lower     91
dtype: int64
```

In [441…
```
df = df.dropna()
display(df)
```

	year	doy	temp	temp_upper	temp_lower
50	851	108.0	7.38	12.10	2.66
63	864	100.0	6.42	8.69	4.14
65	866	106.0	6.44	8.11	4.77
88	889	104.0	6.83	8.48	5.19
90	891	109.0	6.98	8.96	5.00
...	...	...	...	...	...
1175	1976	99.0	8.20	8.77	7.63
1176	1977	93.0	8.22	8.78	7.66
1177	1978	104.0	8.20	8.78	7.61
1178	1979	97.0	8.28	8.83	7.73
1179	1980	102.0	8.30	8.86	7.74

787 rows × 5 columns

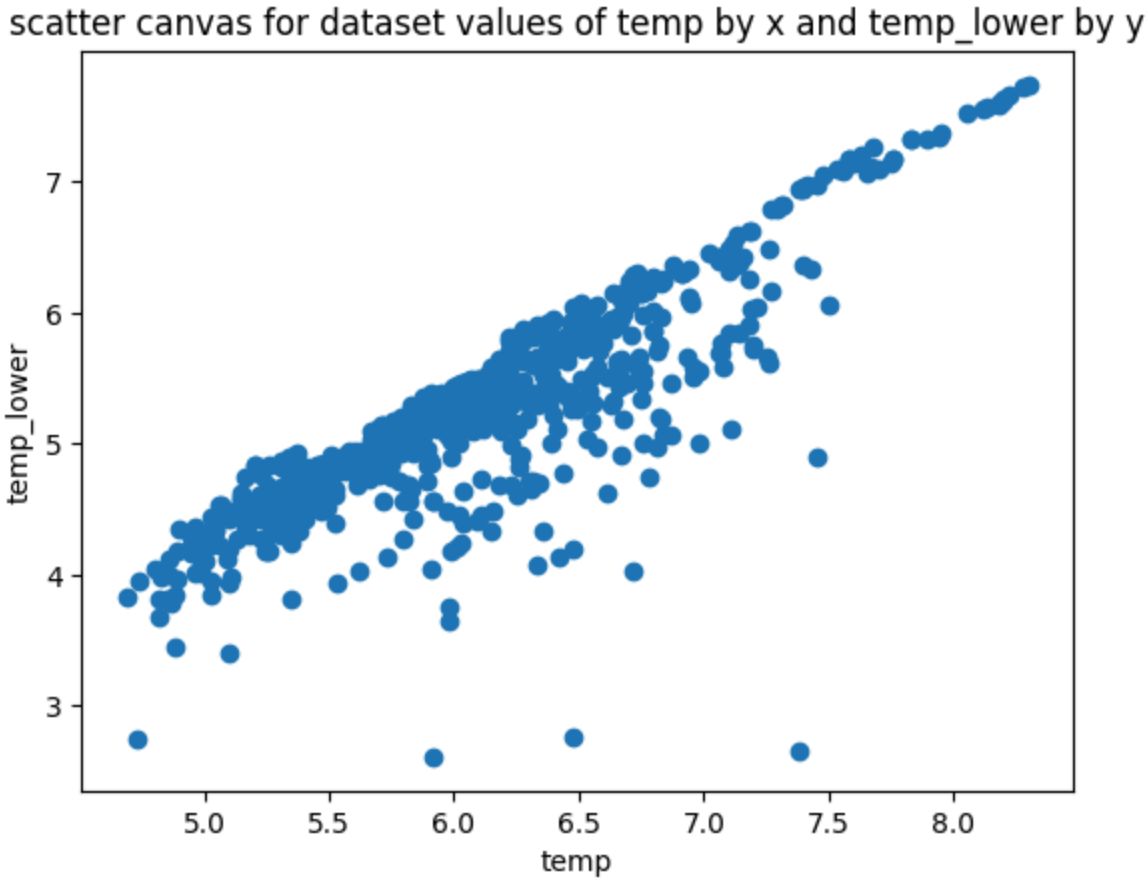
2. Удалите из набора точки с выбросами при помощи стандартизованной оценки (Z-score) таким образом, чтобы точки с выбросами составляли от 5% до 10% всех точек набора данных. Визуализируйте точки исходного набора данных на плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y – зависимый признак), показывая оставленные в наборе точки и удаленные точки разными цветами,

подписывая оси и рисунок и создавая легенду.

In [442…
```
import matplotlib.pyplot as plt
import numpy as np
```

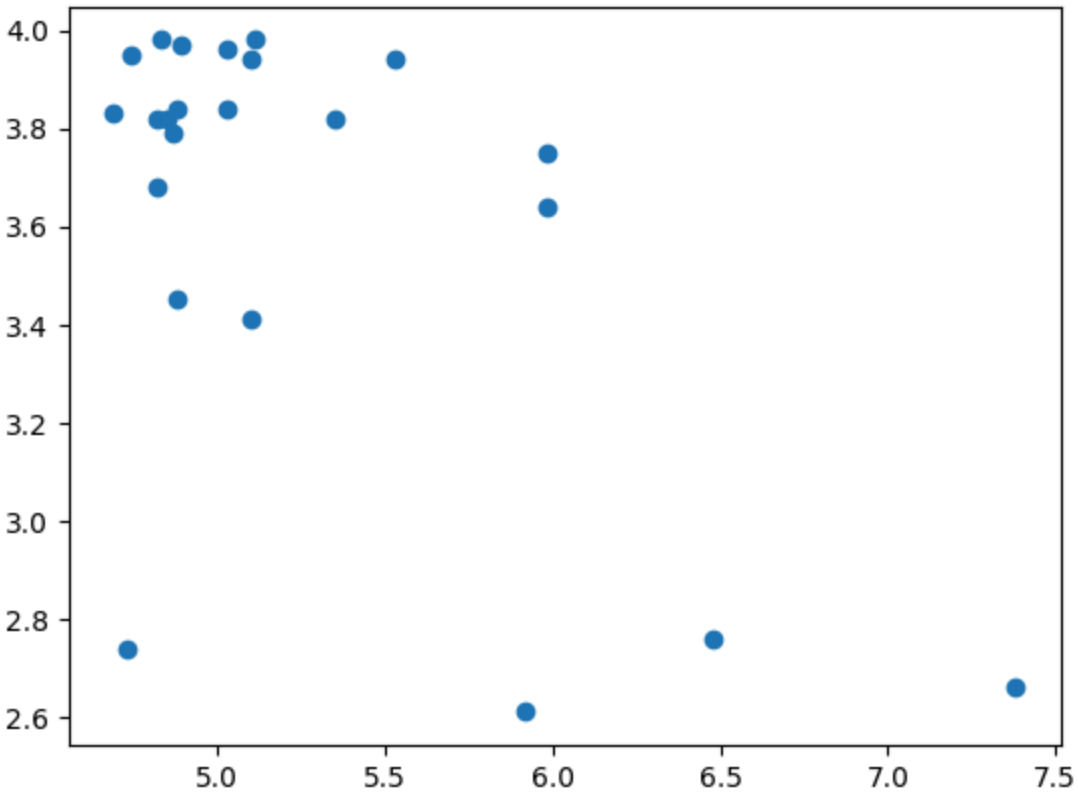
In [443…
```
plt.scatter(df['temp'], df['temp_lower'])
plt.xlabel('temp')
plt.ylabel('temp_lower')
plt.title('scatter canvas for dataset values of temp by x and temp_lower by y')
```

Out[443…
```
Text(0.5, 1.0, 'scatter canvas for dataset values of temp by x and temp_lower by y')
```



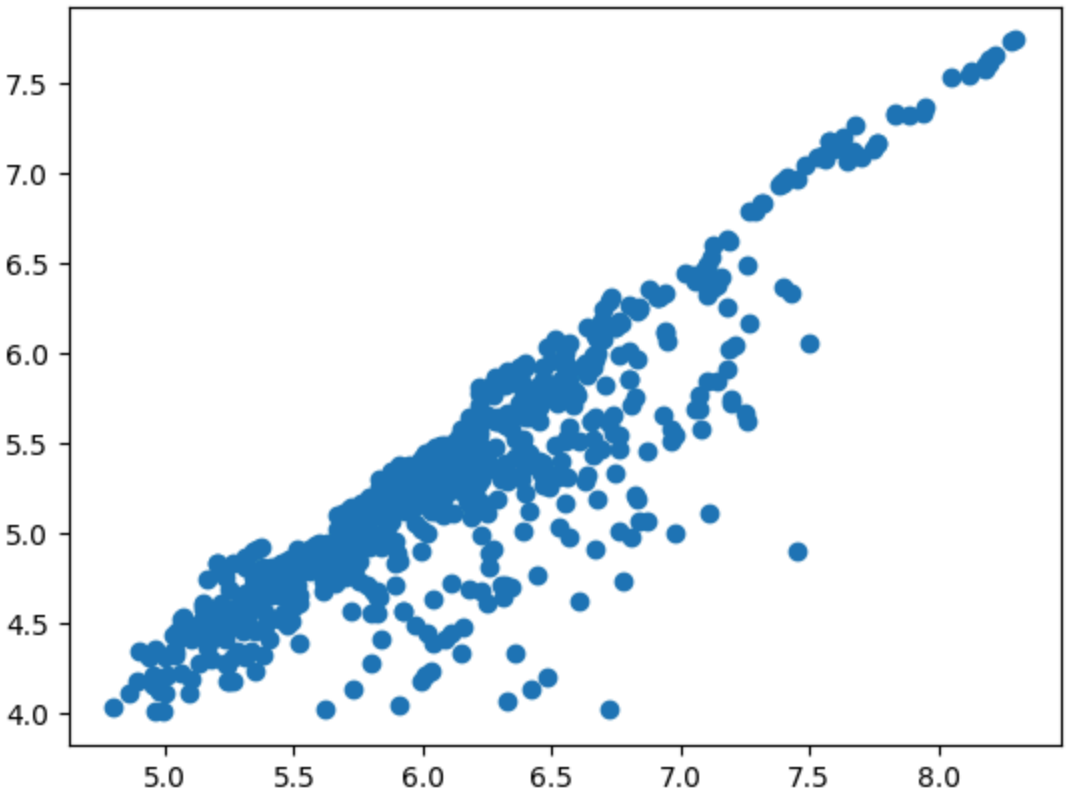
In [444…
```
discarded_df = df[(df['temp_lower'] < 4) & (df['temp'] > 4.5)]
discarded_df.head()
plt.scatter(discarded_df['temp'], discarded_df['temp_lower'])
```

Out[444…
```
<matplotlib.collections.PathCollection at 0x296378ec4a0>
```



In [445…
```
data = df[(df['temp_lower'] > 4)]
data.head()
plt.scatter(data['temp'], data['temp_lower'])
```

Out[445…
```
<matplotlib.collections.PathCollection at 0x2963780cec0>
```

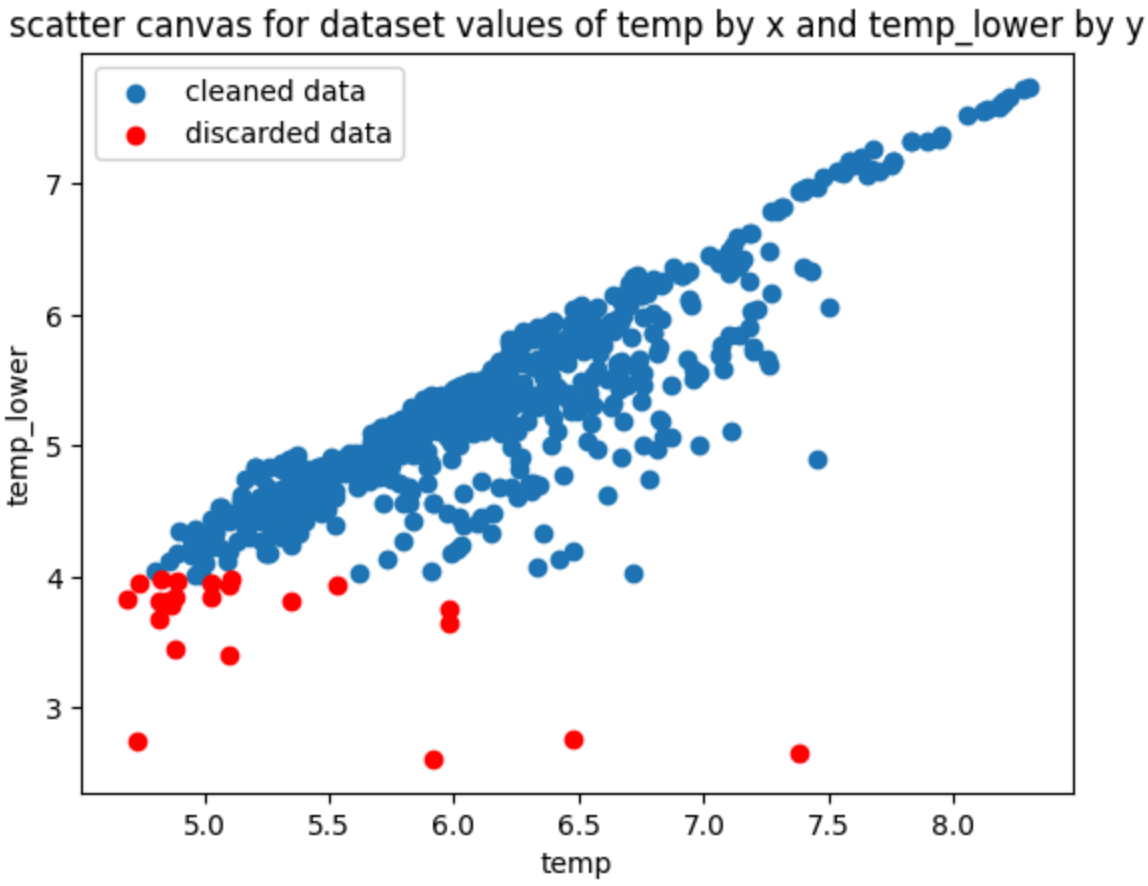


In [446…

```
plt.scatter(data['temp'], data['temp_lower'], label = 'cleaned data')
plt.scatter(discarded_df['temp'], discarded_df['temp_lower'], c='r', label='discard
plt.xlabel('temp')
plt.ylabel('temp_lower')
plt.title('scatter canvas for dataset values of temp by x and temp_lower by y')
plt.legend()
```

Out[446…

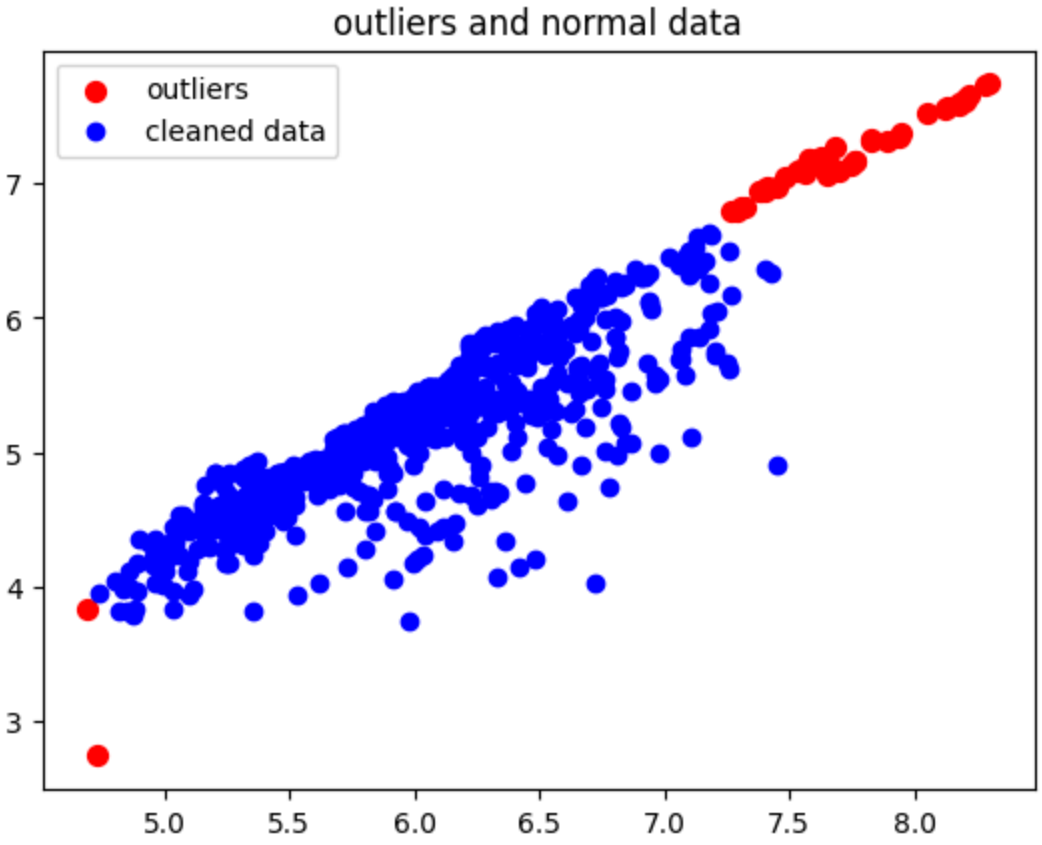
```
<matplotlib.legend.Legend at 0x296403b23c0>
```



In [447…

```
def z_score(df):
    # cols = list(df.columns)
    tmp = df.copy()
    tmp['z_temp_lower']= (tmp['temp_lower'] - tmp['temp_lower'].mean())/tmp['temp_l
    tmp['z_temp']= (tmp['temp'] - tmp['temp'].mean())/tmp['temp'].std(ddof=0)
    # print(tmp.shape)
    tmp1 = tmp[(tmp['z_temp_lower'] < 2) & (tmp['z_temp_lower'] > -2) & (tmp['z_tem
    tmp2 = tmp[(tmp['z_temp_lower'] > 2) | (tmp['z_temp_lower'] < -2) & (tmp['z_tem
    # display(tmp2)
    plt.scatter(tmp2['temp'], tmp2['temp_lower'], c='r', label='outliers', s = 50)
    plt.scatter(tmp1['temp'], tmp1['temp_lower'], c='b', label='cleaned data')

    plt.title('outliers and normal data')
    plt.legend()
    plt.show()
    return tmp1
```

In [448…
```
data = z_score(df)
# data.shape
print("Процент выборки с z_score от изначальной: ", round(data.shape[0]/df.shape[0]
```




```
Процент выборки с z_score от изначальной:  93.27
```

Таким образом, мы избавились от 7 процентов выборки

3. Выполните стандартизацию независимого признака и масштабирование на интервал [-1, 1] зависимого признака. Решите задачи линейной регрессии и полиномиальной регрессии для степени полинома, указанной в индивидуальном задании, при помощи нейронных сетей с одним нейроном и оцените качество полученных моделей по показателю, указанному в индивидуальном задании.

Отследите обучение нейронных сетей, изменяя, при необходимости, гиперпараметры (функцию потерь, оптимизатор, шаг обучения и т.п.) или применяя регуляризацию.

Независимая переменная: temp Зависимая переменная: temp_lower

In [449…

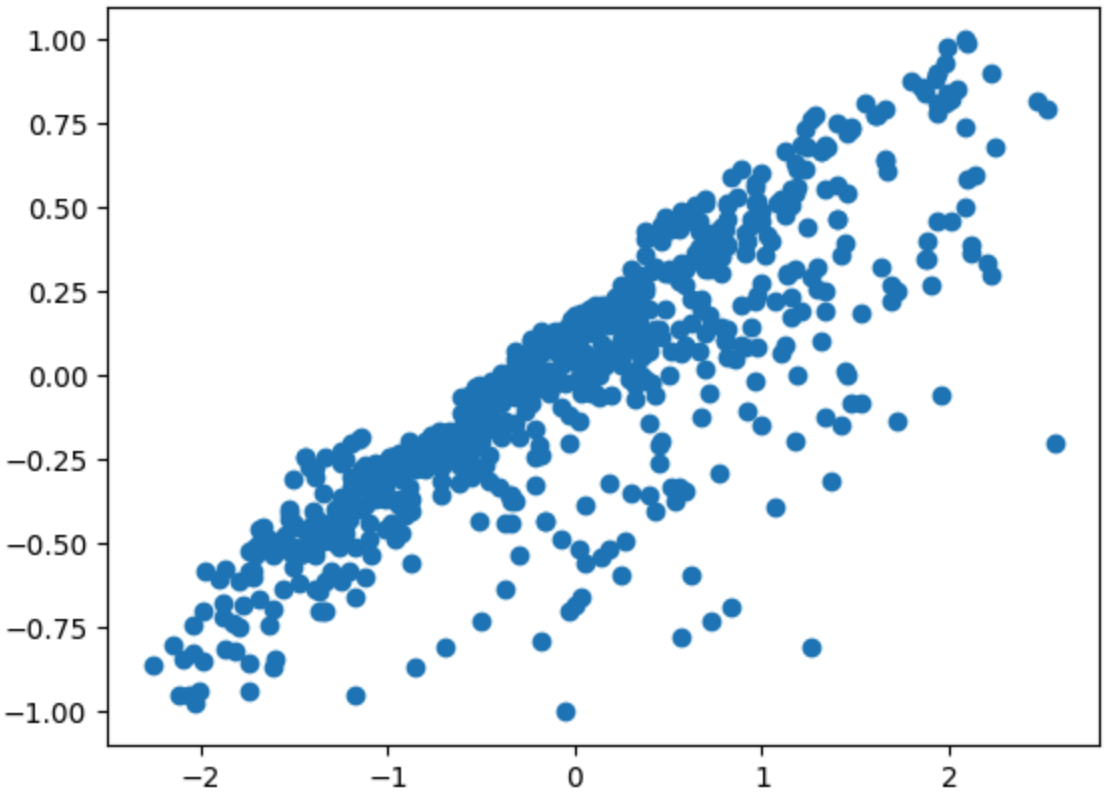
```
train_list = [(2*((data['temp_lower'] - data['temp_lower'].min())/(data['temp_lower
# train_list = [(2*((data['temp_lower'] - data['temp_lower'].min())/(data['temp_low
ds = pd.DataFrame(columns=['temp', 'temp_lower'])
ds['temp_lower'] = train_list[0]
ds['temp'] = train_list[1]

print(train.shape, ds.shape)

train = ds
test = ds.iloc[300:500, :]

plt.scatter(train['temp'], train['temp_lower'], label='data')
display(test.head())
train.tail()
```

```
(734, 2) (734, 2)
```

	temp	temp_lower
300	0.782793	0.361111
301	0.782793	0.305556
302	0.658285	0.222222
303	0.622711	0.159722
304	0.587137	0.090278

Out[449…

	temp	temp_lower
729	1.921154	0.881944
730	1.974515	0.930556
731	2.099023	0.993056
732	1.992302	0.979167
733	2.081236	1.000000



In [450…
```
import tensorflow as tf
```

Линейная регрессия

In [451…
```
# Нормализация не требуется, так как выборка очень хорошо очищена, что видно на гра
df_normalizer = tf.keras.layers.Normalization(axis = None)
df_normalizer.adapt(train['temp'].values)
```

In [452…
```
lin_reg = tf.keras.Sequential([
    # df_normalizer,
    tf.keras.layers.Dense(units = 1,
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01))
])
```

In [453…
```
lin_reg.compile(
    optimizer = tf.optimizers.Adam(learning_rate=0.01),
    loss = 'mse',
    metrics = ['r2_score']
)
```

In [454…
```
%%time
hist = lin_reg.fit(
    train['temp'], train['temp_lower'],
    epochs = 100,
    verbose = 1,
    validation_split = 0.4
)
```

```
Epoch 1/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 3s 47ms/step - loss: 3.2319 - r2_score: -15.5275 - val_lo
ss: 1.8089 - val_r2_score: -14.5206
Epoch 2/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 2.6630 - r2_score: -13.6352 - val_los
s: 1.5560 - val_r2_score: -12.3597
Epoch 3/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 2.1616 - r2_score: -10.6579 - val_los
s: 1.3271 - val_r2_score: -10.4024
Epoch 4/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1.9041 - r2_score: -8.4323 - val_los
s: 1.1145 - val_r2_score: -8.5820
Epoch 5/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 1.5059 - r2_score: -6.5282 - val_los
s: 0.9233 - val_r2_score: -6.9427
Epoch 6/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 1.2098 - r2_score: -5.5763 - val_los
s: 0.7436 - val_r2_score: -5.4009
Epoch 7/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.9810 - r2_score: -4.3440 - val_los
s: 0.5971 - val_r2_score: -4.1430
Epoch 8/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.7953 - r2_score: -3.3008 - val_los
s: 0.4695 - val_r2_score: -3.0469
Epoch 9/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.7100 - r2_score: -2.5302 - val_los
s: 0.3674 - val_r2_score: -2.1691
Epoch 10/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5056 - r2_score: -1.7808 - val_los
s: 0.2859 - val_r2_score: -1.4680
Epoch 11/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.4331 - r2_score: -1.1398 - val_los
s: 0.2229 - val_r2_score: -0.9252
Epoch 12/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.3276 - r2_score: -0.7195 - val_los
s: 0.1724 - val_r2_score: -0.4900
Epoch 13/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.2730 - r2_score: -0.3656 - val_los
s: 0.1345 - val_r2_score: -0.1633
Epoch 14/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.2228 - r2_score: -0.0910 - val_los
s: 0.1063 - val_r2_score: 0.0803
Epoch 15/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1703 - r2_score: 0.1136 - val_loss:
0.0859 - val_r2_score: 0.2575
Epoch 16/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1431 - r2_score: 0.2686 - val_loss:
0.0711 - val_r2_score: 0.3863
Epoch 17/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1152 - r2_score: 0.3809 - val_loss:
0.0605 - val_r2_score: 0.4785
Epoch 18/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0977 - r2_score: 0.4359 - val_loss:
0.0548 - val_r2_score: 0.5284
Epoch 19/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0918 - r2_score: 0.5082 - val_loss:
```

```
0.0498 - val_r2_score: 0.5724
Epoch 20/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0803 - r2_score: 0.5910 - val_loss:
0.0450 - val_r2_score: 0.6147
Epoch 21/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0761 - r2_score: 0.5813 - val_loss:
0.0455 - val_r2_score: 0.6120
Epoch 22/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0770 - r2_score: 0.6197 - val_loss:
0.0445 - val_r2_score: 0.6206
Epoch 23/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0639 - r2_score: 0.6506 - val_loss:
0.0450 - val_r2_score: 0.6170
Epoch 24/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0717 - r2_score: 0.6345 - val_loss:
0.0454 - val_r2_score: 0.6142
Epoch 25/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0610 - r2_score: 0.6827 - val_loss:
0.0435 - val_r2_score: 0.6310
Epoch 26/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0716 - r2_score: 0.6243 - val_loss:
0.0450 - val_r2_score: 0.6184
Epoch 27/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0682 - r2_score: 0.6411 - val_loss:
0.0443 - val_r2_score: 0.6250
Epoch 28/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0591 - r2_score: 0.6968 - val_loss:
0.0450 - val_r2_score: 0.6195
Epoch 29/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0664 - r2_score: 0.6524 - val_loss:
0.0462 - val_r2_score: 0.6091
Epoch 30/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0550 - r2_score: 0.7080 - val_loss:
0.0458 - val_r2_score: 0.6125
Epoch 31/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0699 - r2_score: 0.6552 - val_loss:
0.0451 - val_r2_score: 0.6184
Epoch 32/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0610 - r2_score: 0.6732 - val_loss:
0.0465 - val_r2_score: 0.6071
Epoch 33/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0537 - r2_score: 0.7071 - val_loss:
0.0455 - val_r2_score: 0.6152
Epoch 34/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0614 - r2_score: 0.6487 - val_loss:
0.0471 - val_r2_score: 0.6020
Epoch 35/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0610 - r2_score: 0.6656 - val_loss:
0.0478 - val_r2_score: 0.5957
Epoch 36/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0617 - r2_score: 0.6742 - val_loss:
0.0459 - val_r2_score: 0.6119
Epoch 37/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0643 - r2_score: 0.6809 - val_loss:
0.0452 - val_r2_score: 0.6186
Epoch 38/100
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0682 - r2_score: 0.6406 - val_loss:
0.0481 - val_r2_score: 0.5933
Epoch 39/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0671 - r2_score: 0.6631 - val_loss:
0.0472 - val_r2_score: 0.6010
Epoch 40/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0635 - r2_score: 0.6621 - val_loss:
0.0466 - val_r2_score: 0.6063
Epoch 41/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0639 - r2_score: 0.6607 - val_loss:
0.0467 - val_r2_score: 0.6057
Epoch 42/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0697 - r2_score: 0.6270 - val_loss:
0.0480 - val_r2_score: 0.5945
Epoch 43/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0614 - r2_score: 0.6787 - val_loss:
0.0462 - val_r2_score: 0.6095
Epoch 44/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0674 - r2_score: 0.6370 - val_loss:
0.0472 - val_r2_score: 0.6007
Epoch 45/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0590 - r2_score: 0.6808 - val_loss:
0.0450 - val_r2_score: 0.6201
Epoch 46/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0574 - r2_score: 0.7006 - val_loss:
0.0481 - val_r2_score: 0.5936
Epoch 47/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0636 - r2_score: 0.6455 - val_loss:
0.0489 - val_r2_score: 0.5862
Epoch 48/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0650 - r2_score: 0.6402 - val_loss:
0.0454 - val_r2_score: 0.6167
Epoch 49/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0617 - r2_score: 0.6892 - val_loss:
0.0447 - val_r2_score: 0.6225
Epoch 50/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0627 - r2_score: 0.6852 - val_loss:
0.0485 - val_r2_score: 0.5901
Epoch 51/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0624 - r2_score: 0.6765 - val_loss:
0.0466 - val_r2_score: 0.6062
Epoch 52/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0652 - r2_score: 0.6513 - val_loss:
0.0469 - val_r2_score: 0.6037
Epoch 53/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0585 - r2_score: 0.6974 - val_loss:
0.0465 - val_r2_score: 0.6070
Epoch 54/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0580 - r2_score: 0.6877 - val_loss:
0.0464 - val_r2_score: 0.6083
Epoch 55/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0631 - r2_score: 0.6687 - val_loss:
0.0485 - val_r2_score: 0.5900
Epoch 56/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0603 - r2_score: 0.6782 - val_loss:
0.0463 - val_r2_score: 0.6089
```

```
Epoch 57/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0587 - r2_score: 0.6767 - val_loss:
0.0446 - val_r2_score: 0.6236
Epoch 58/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0617 - r2_score: 0.6867 - val_loss:
0.0468 - val_r2_score: 0.6043
Epoch 59/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0640 - r2_score: 0.6624 - val_loss:
0.0490 - val_r2_score: 0.5856
Epoch 60/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0616 - r2_score: 0.6669 - val_loss:
0.0462 - val_r2_score: 0.6098
Epoch 61/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0639 - r2_score: 0.6384 - val_loss:
0.0466 - val_r2_score: 0.6065
Epoch 62/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0525 - r2_score: 0.7353 - val_loss:
0.0447 - val_r2_score: 0.6225
Epoch 63/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0574 - r2_score: 0.7073 - val_loss:
0.0490 - val_r2_score: 0.5857
Epoch 64/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0656 - r2_score: 0.6844 - val_loss:
0.0470 - val_r2_score: 0.6033
Epoch 65/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0620 - r2_score: 0.6868 - val_loss:
0.0457 - val_r2_score: 0.6141
Epoch 66/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0636 - r2_score: 0.6455 - val_loss:
0.0486 - val_r2_score: 0.5886
Epoch 67/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0539 - r2_score: 0.7274 - val_loss:
0.0451 - val_r2_score: 0.6190
Epoch 68/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0605 - r2_score: 0.6797 - val_loss:
0.0469 - val_r2_score: 0.6040
Epoch 69/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0646 - r2_score: 0.6496 - val_loss:
0.0484 - val_r2_score: 0.5905
Epoch 70/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0634 - r2_score: 0.6649 - val_loss:
0.0466 - val_r2_score: 0.6066
Epoch 71/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0675 - r2_score: 0.6547 - val_loss:
0.0466 - val_r2_score: 0.6064
Epoch 72/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0580 - r2_score: 0.6994 - val_loss:
0.0460 - val_r2_score: 0.6112
Epoch 73/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0621 - r2_score: 0.6725 - val_loss:
0.0483 - val_r2_score: 0.5913
Epoch 74/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0673 - r2_score: 0.6537 - val_loss:
0.0447 - val_r2_score: 0.6232
Epoch 75/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0609 - r2_score: 0.6922 - val_loss:
```

```
0.0456 - val_r2_score: 0.6152
Epoch 76/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0666 - r2_score: 0.6643 - val_loss:
0.0478 - val_r2_score: 0.5956
Epoch 77/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0632 - r2_score: 0.6947 - val_loss:
0.0466 - val_r2_score: 0.6060
Epoch 78/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0665 - r2_score: 0.6395 - val_loss:
0.0495 - val_r2_score: 0.5810
Epoch 79/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0726 - r2_score: 0.6336 - val_loss:
0.0464 - val_r2_score: 0.6084
Epoch 80/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0611 - r2_score: 0.6648 - val_loss:
0.0453 - val_r2_score: 0.6171
Epoch 81/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0615 - r2_score: 0.6830 - val_loss:
0.0475 - val_r2_score: 0.5985
Epoch 82/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0579 - r2_score: 0.6979 - val_loss:
0.0463 - val_r2_score: 0.6094
Epoch 83/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0666 - r2_score: 0.6363 - val_loss:
0.0481 - val_r2_score: 0.5932
Epoch 84/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0657 - r2_score: 0.6852 - val_loss:
0.0472 - val_r2_score: 0.6010
Epoch 85/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0642 - r2_score: 0.6579 - val_loss:
0.0477 - val_r2_score: 0.5972
Epoch 86/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0628 - r2_score: 0.6586 - val_loss:
0.0462 - val_r2_score: 0.6098
Epoch 87/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0656 - r2_score: 0.6185 - val_loss:
0.0486 - val_r2_score: 0.5888
Epoch 88/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0615 - r2_score: 0.6537 - val_loss:
0.0442 - val_r2_score: 0.6268
Epoch 89/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0633 - r2_score: 0.6718 - val_loss:
0.0467 - val_r2_score: 0.6056
Epoch 90/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.0618 - r2_score: 0.6438 - val_los
s: 0.0467 - val_r2_score: 0.6053
Epoch 91/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0640 - r2_score: 0.6948 - val_loss:
0.0505 - val_r2_score: 0.5725
Epoch 92/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0614 - r2_score: 0.6667 - val_loss:
0.0440 - val_r2_score: 0.6289
Epoch 93/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0619 - r2_score: 0.6671 - val_loss:
0.0477 - val_r2_score: 0.5971
Epoch 94/100
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0595 - r2_score: 0.6891 - val_loss:
0.0454 - val_r2_score: 0.6166
Epoch 95/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0724 - r2_score: 0.6172 - val_loss:
0.0485 - val_r2_score: 0.5896
Epoch 96/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0614 - r2_score: 0.6796 - val_loss:
0.0455 - val_r2_score: 0.6157
Epoch 97/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0598 - r2_score: 0.6986 - val_loss:
0.0458 - val_r2_score: 0.6130
Epoch 98/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0688 - r2_score: 0.6602 - val_loss:
0.0487 - val_r2_score: 0.5879
Epoch 99/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0690 - r2_score: 0.6602 - val_loss:
0.0462 - val_r2_score: 0.6100
Epoch 100/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0672 - r2_score: 0.6356 - val_loss:
0.0453 - val_r2_score: 0.6175
CPU times: total: 8.67 s
Wall time: 20.2 s
```

In [455…

```python
print(hist.history.keys())
```

```
dict_keys(['loss', 'r2_score', 'val_loss', 'val_r2_score'])
```

In [456…

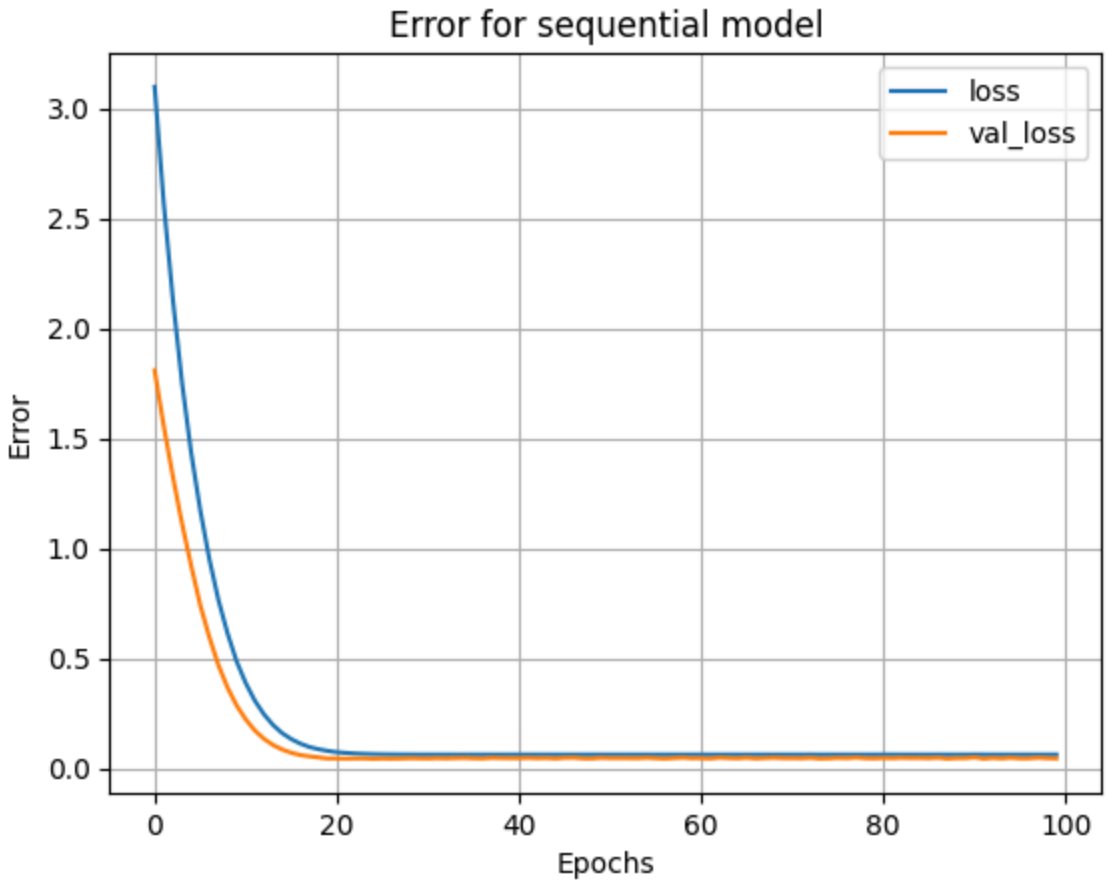
```python
def plot_loss(history):
    plt.plot(history.history['loss'], label = 'loss')
    plt.plot(history.history['val_loss'], label = 'val_loss')
    # plt.ylim([0, max(history.history['loss'])])
    plt.xlabel('Epochs')
    plt.ylabel('Error')
    plt.title('Error for sequential model')
    plt.legend()
    plt.grid(True)
```

In [457…

```python
def plot_metrics(history):
    plt.plot(history.history['r2_score'], label = 'r2_score')
    plt.plot(history.history['val_r2_score'], label = 'val_r2_score')
    # plt.ylim([0, max(history.history['r2_score'])*1])
    plt.xlabel('Epochs')
    plt.ylabel('Value')
    plt.title('R2 score for sequential model')
    plt.legend()
    plt.grid(True)
```

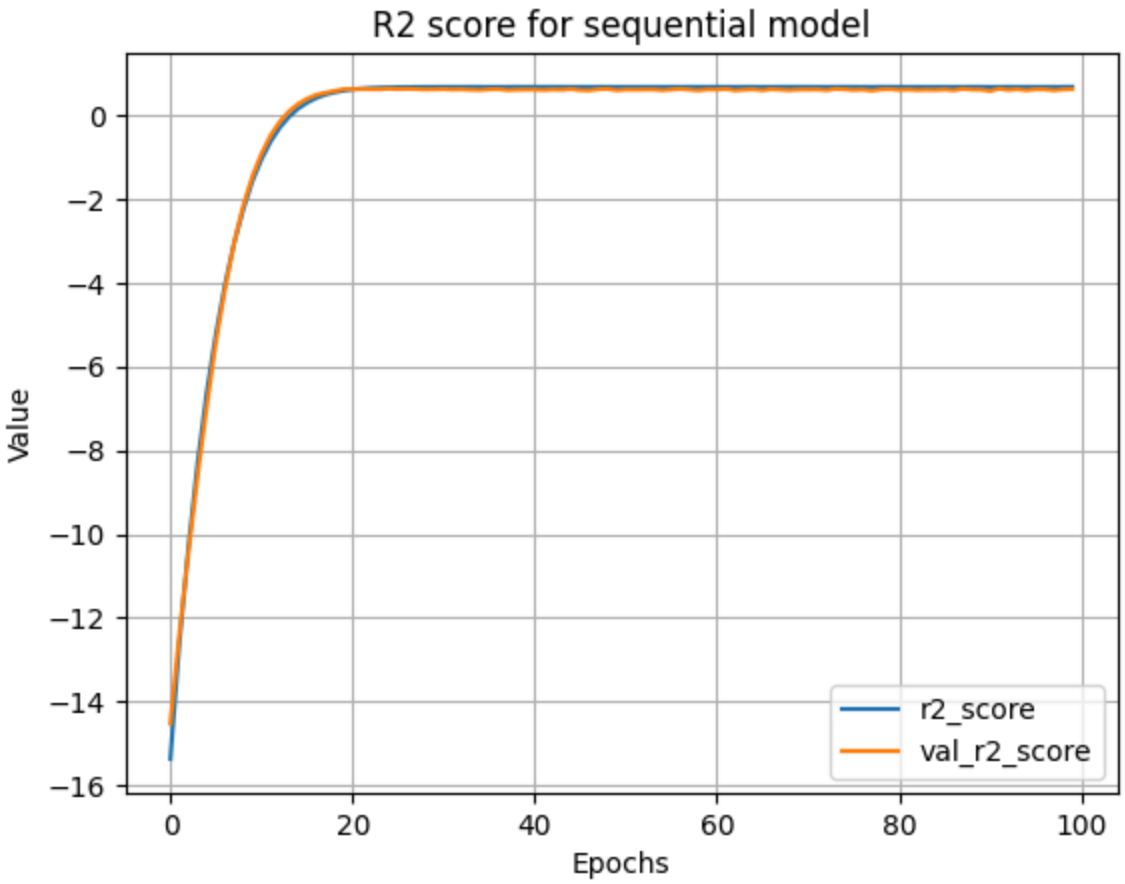
In [458…

```python
plot_loss(hist)
```



In [459…

```
plot_metrics(hist)
```

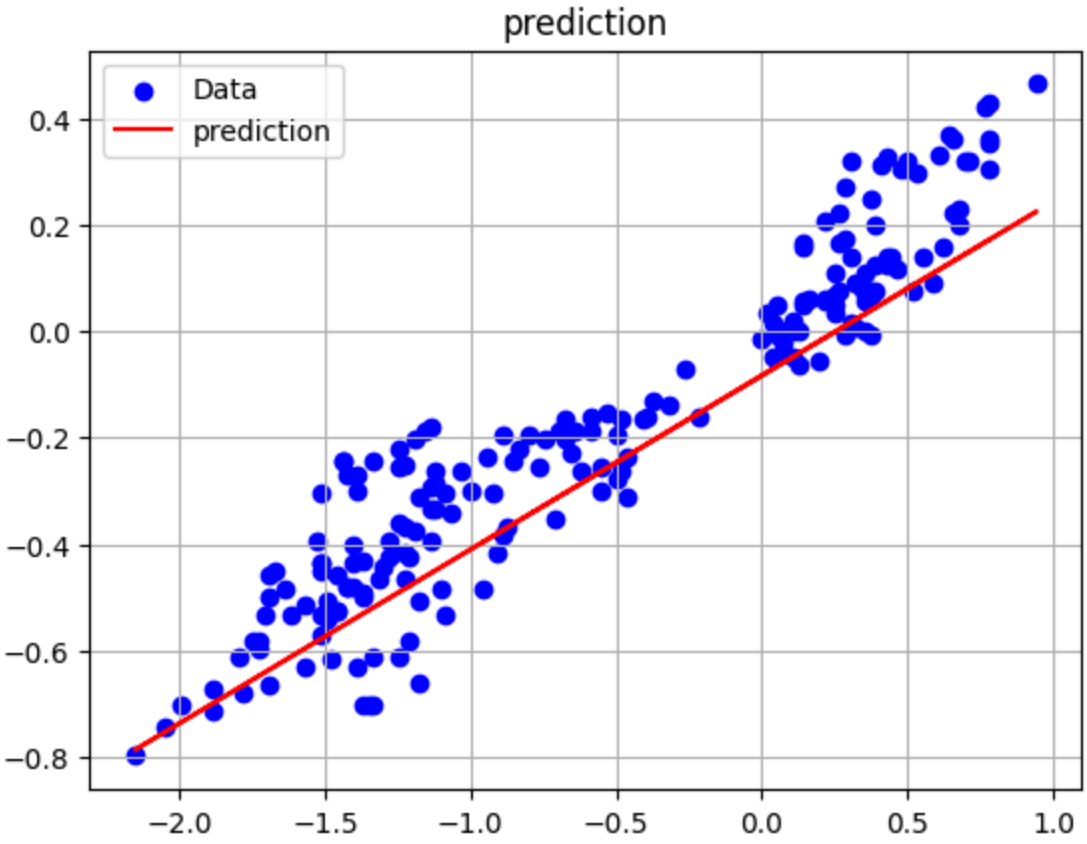


In [460…
```
def print_res(yhat):
    plt.scatter(test['temp'], test['temp_lower'], c='b', label='Data')
    plt.plot(test['temp'], yhat, c = 'r', label='prediction')
    plt.title('prediction')
    plt.legend()
    plt.grid(True)
```

In [461…
```
y_lin = lin_reg.predict(test['temp'].values)
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
```

In [462…
```
print_res(y_lin)
```



Нелинейная регрессия

In [463…
```
# tmp = train['temp'].copy()
x = np.array([train['temp'].values, train['temp'].values**2, train['temp'].values**
# print(x[0:5])
x = x.T
print(x[0:5])

# def preparePol():
```

```
[[0.7294323  0.53207149 0.38811013]
 [0.76500609 0.58523432 0.44770782]
 [1.45869498 2.12779105 3.10379812]
 [1.7254984  2.97734472 5.13740355]
 [1.95672803 3.82878457 7.49189008]]
```

In [464…
```
pol_reg = tf.keras.Sequential([
    tf.keras.Input(shape=(3, )),
    tf.keras.layers.Dense(units=1,
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01))
])
```

In [465…
```
pol_reg.summary()
```

```
Model: "sequential_19"
```

Layer (type)	Output Shape	
dense_21 (Dense)	(None, 1)	

Total params: 4 (16.00 B)

Trainable params: 4 (16.00 B)

Non-trainable params: 0 (0.00 B)

In [466…

```
pol_reg.compile(
    optimizer=tf.optimizers.Adam(learning_rate=0.01),
    loss='mse',
    metrics=['r2_score']
)
```

In [467…

```
%%time
phist = pol_reg.fit(
    x, train['temp_lower'],
    epochs = 100,
    verbose = 1,
    validation_split = 0.2
)
```

```
Epoch 1/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 2s 34ms/step - loss: 4.6855 - r2_score: -29.5507 - val_lo
ss: 1.7879 - val_r2_score: -11.3609
Epoch 2/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 2.7752 - r2_score: -16.6265 - val_los
s: 0.9629 - val_r2_score: -5.6107
Epoch 3/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 1.2248 - r2_score: -6.6892 - val_los
s: 0.6290 - val_r2_score: -3.2813
Epoch 4/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 1.0896 - r2_score: -5.0140 - val_los
s: 0.5272 - val_r2_score: -2.5708
Epoch 5/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.6914 - r2_score: -3.5912 - val_los
s: 0.5041 - val_r2_score: -2.4145
Epoch 6/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5987 - r2_score: -2.6940 - val_los
s: 0.4833 - val_r2_score: -2.2768
Epoch 7/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5567 - r2_score: -2.6199 - val_los
s: 0.4532 - val_r2_score: -2.0758
Epoch 8/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5404 - r2_score: -2.2854 - val_los
s: 0.4205 - val_r2_score: -1.8565
Epoch 9/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4690 - r2_score: -1.9349 - val_los
s: 0.3851 - val_r2_score: -1.6183
Epoch 10/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4190 - r2_score: -1.5031 - val_los
s: 0.3595 - val_r2_score: -1.4478
Epoch 11/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.3972 - r2_score: -1.3318 - val_los
s: 0.3296 - val_r2_score: -1.2466
Epoch 12/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.3811 - r2_score: -1.4181 - val_los
s: 0.2980 - val_r2_score: -1.0336
Epoch 13/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.3221 - r2_score: -0.9685 - val_los
s: 0.2704 - val_r2_score: -0.8480
Epoch 14/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2815 - r2_score: -0.6687 - val_los
s: 0.2424 - val_r2_score: -0.6582
Epoch 15/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.2745 - r2_score: -0.7090 - val_los
s: 0.2313 - val_r2_score: -0.5855
Epoch 16/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2287 - r2_score: -0.4425 - val_los
s: 0.2048 - val_r2_score: -0.4056
Epoch 17/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1947 - r2_score: -0.2813 - val_los
s: 0.1869 - val_r2_score: -0.2849
Epoch 18/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1804 - r2_score: -0.1861 - val_los
s: 0.1797 - val_r2_score: -0.2381
Epoch 19/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1721 - r2_score: -0.0469 - val_los
```

```
s: 0.1559 - val_r2_score: -0.0750
Epoch 20/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1826 - r2_score: -0.1134 - val_los
s: 0.1390 - val_r2_score: 0.0398
Epoch 21/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1290 - r2_score: 0.1462 - val_loss:
0.1267 - val_r2_score: 0.1234
Epoch 22/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1276 - r2_score: 0.1300 - val_loss:
0.1211 - val_r2_score: 0.1613
Epoch 23/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1191 - r2_score: 0.2412 - val_loss:
0.1091 - val_r2_score: 0.2436
Epoch 24/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1127 - r2_score: 0.3174 - val_loss:
0.1040 - val_r2_score: 0.2781
Epoch 25/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0863 - r2_score: 0.4397 - val_loss:
0.0942 - val_r2_score: 0.3454
Epoch 26/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0969 - r2_score: 0.3787 - val_loss:
0.0867 - val_r2_score: 0.3972
Epoch 27/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0798 - r2_score: 0.4992 - val_loss:
0.0788 - val_r2_score: 0.4520
Epoch 28/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0734 - r2_score: 0.5250 - val_loss:
0.0823 - val_r2_score: 0.4274
Epoch 29/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0782 - r2_score: 0.5415 - val_loss:
0.0739 - val_r2_score: 0.4857
Epoch 30/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0722 - r2_score: 0.5618 - val_loss:
0.0734 - val_r2_score: 0.4891
Epoch 31/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0633 - r2_score: 0.5743 - val_loss:
0.0674 - val_r2_score: 0.5315
Epoch 32/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0660 - r2_score: 0.5752 - val_loss:
0.0667 - val_r2_score: 0.5363
Epoch 33/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0569 - r2_score: 0.6124 - val_loss:
0.0631 - val_r2_score: 0.5621
Epoch 34/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0678 - r2_score: 0.5329 - val_loss:
0.0597 - val_r2_score: 0.5854
Epoch 35/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0582 - r2_score: 0.6215 - val_loss:
0.0585 - val_r2_score: 0.5943
Epoch 36/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0526 - r2_score: 0.6633 - val_loss:
0.0621 - val_r2_score: 0.5697
Epoch 37/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0591 - r2_score: 0.6060 - val_loss:
0.0603 - val_r2_score: 0.5824
Epoch 38/100
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0560 - r2_score: 0.6609 - val_loss:
0.0594 - val_r2_score: 0.5890
Epoch 39/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0535 - r2_score: 0.6692 - val_loss:
0.0556 - val_r2_score: 0.6157
Epoch 40/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0491 - r2_score: 0.6908 - val_loss:
0.0585 - val_r2_score: 0.5961
Epoch 41/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0524 - r2_score: 0.6831 - val_loss:
0.0596 - val_r2_score: 0.5884
Epoch 42/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0567 - r2_score: 0.6480 - val_loss:
0.0545 - val_r2_score: 0.6241
Epoch 43/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0511 - r2_score: 0.6765 - val_loss:
0.0574 - val_r2_score: 0.6040
Epoch 44/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0543 - r2_score: 0.6567 - val_loss:
0.0564 - val_r2_score: 0.6114
Epoch 45/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0517 - r2_score: 0.6628 - val_loss:
0.0541 - val_r2_score: 0.6274
Epoch 46/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0515 - r2_score: 0.6816 - val_loss:
0.0543 - val_r2_score: 0.6268
Epoch 47/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0489 - r2_score: 0.6654 - val_loss:
0.0548 - val_r2_score: 0.6234
Epoch 48/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0477 - r2_score: 0.6977 - val_loss:
0.0537 - val_r2_score: 0.6310
Epoch 49/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0579 - r2_score: 0.6480 - val_loss:
0.0528 - val_r2_score: 0.6378
Epoch 50/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0494 - r2_score: 0.6860 - val_loss:
0.0543 - val_r2_score: 0.6273
Epoch 51/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0488 - r2_score: 0.7047 - val_loss:
0.0565 - val_r2_score: 0.6119
Epoch 52/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0494 - r2_score: 0.6783 - val_loss:
0.0547 - val_r2_score: 0.6245
Epoch 53/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0493 - r2_score: 0.6867 - val_loss:
0.0539 - val_r2_score: 0.6303
Epoch 54/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0526 - r2_score: 0.6643 - val_loss:
0.0576 - val_r2_score: 0.6046
Epoch 55/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0597 - r2_score: 0.5960 - val_loss:
0.0540 - val_r2_score: 0.6293
Epoch 56/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0522 - r2_score: 0.6712 - val_loss:
0.0509 - val_r2_score: 0.6512
```

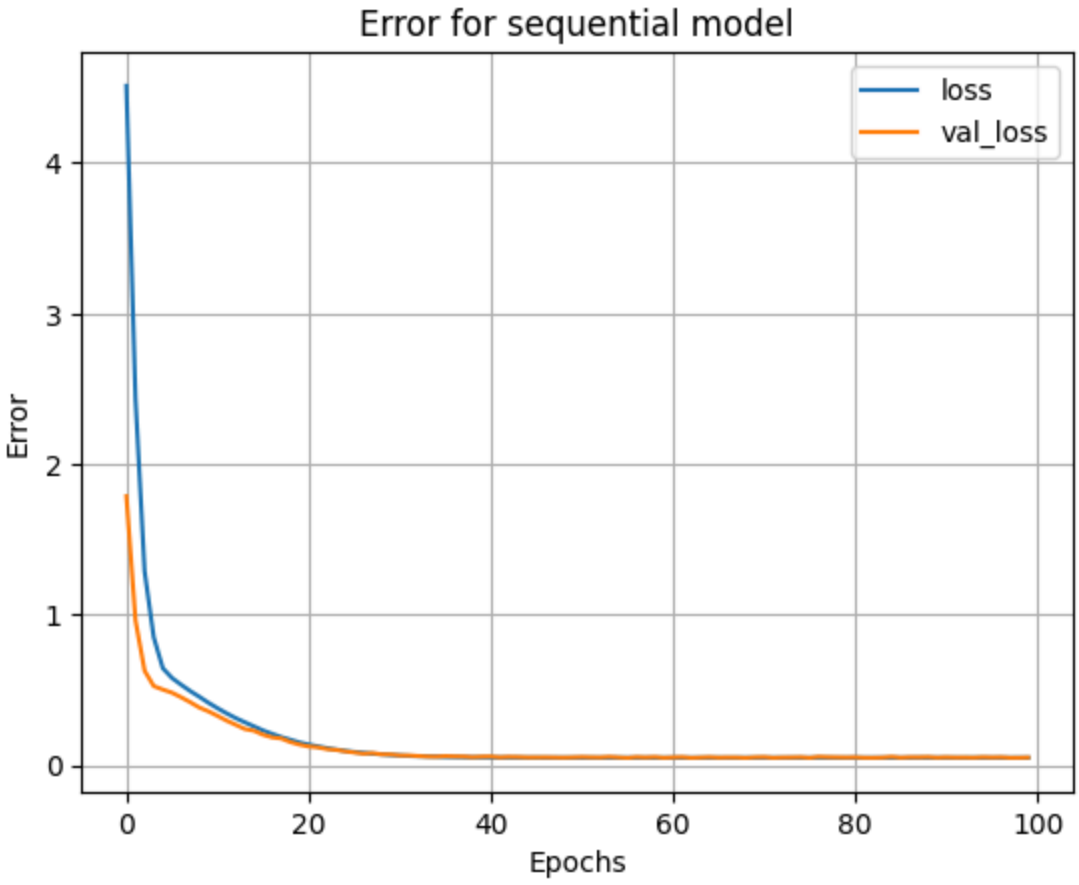
```
Epoch 57/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0521 - r2_score: 0.6852 - val_loss:
0.0563 - val_r2_score: 0.6138
Epoch 58/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0503 - r2_score: 0.6975 - val_loss:
0.0536 - val_r2_score: 0.6327
Epoch 59/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0568 - r2_score: 0.6571 - val_loss:
0.0566 - val_r2_score: 0.6118
Epoch 60/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0537 - r2_score: 0.6515 - val_loss:
0.0515 - val_r2_score: 0.6475
Epoch 61/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.0488 - r2_score: 0.6950 - val_loss:
0.0542 - val_r2_score: 0.6286
Epoch 62/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0498 - r2_score: 0.6851 - val_loss:
0.0561 - val_r2_score: 0.6151
Epoch 63/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0538 - r2_score: 0.6513 - val_loss:
0.0510 - val_r2_score: 0.6506
Epoch 64/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0534 - r2_score: 0.6762 - val_loss:
0.0533 - val_r2_score: 0.6349
Epoch 65/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0510 - r2_score: 0.6528 - val_loss:
0.0557 - val_r2_score: 0.6178
Epoch 66/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.0544 - r2_score: 0.6628 - val_loss:
0.0526 - val_r2_score: 0.6400
Epoch 67/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0582 - r2_score: 0.6546 - val_loss:
0.0539 - val_r2_score: 0.6309
Epoch 68/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0593 - r2_score: 0.6587 - val_loss:
0.0534 - val_r2_score: 0.6340
Epoch 69/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0470 - r2_score: 0.6930 - val_loss:
0.0510 - val_r2_score: 0.6514
Epoch 70/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.0488 - r2_score: 0.7152 - val_los
s: 0.0549 - val_r2_score: 0.6239
Epoch 71/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0539 - r2_score: 0.6656 - val_loss:
0.0562 - val_r2_score: 0.6150
Epoch 72/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0502 - r2_score: 0.6807 - val_loss:
0.0514 - val_r2_score: 0.6480
Epoch 73/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0483 - r2_score: 0.6812 - val_loss:
0.0535 - val_r2_score: 0.6338
Epoch 74/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0539 - r2_score: 0.6435 - val_loss:
0.0540 - val_r2_score: 0.6301
Epoch 75/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0483 - r2_score: 0.6965 - val_loss:
```

```
0.0555 - val_r2_score: 0.6198
Epoch 76/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0573 - r2_score: 0.6437 - val_loss:
0.0510 - val_r2_score: 0.6512
Epoch 77/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0453 - r2_score: 0.7025 - val_loss:
0.0581 - val_r2_score: 0.6019
Epoch 78/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0454 - r2_score: 0.7215 - val_loss:
0.0558 - val_r2_score: 0.6180
Epoch 79/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0570 - r2_score: 0.6490 - val_loss:
0.0547 - val_r2_score: 0.6255
Epoch 80/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0504 - r2_score: 0.6833 - val_loss:
0.0538 - val_r2_score: 0.6317
Epoch 81/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0579 - r2_score: 0.6779 - val_loss:
0.0548 - val_r2_score: 0.6250
Epoch 82/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0549 - r2_score: 0.6581 - val_loss:
0.0519 - val_r2_score: 0.6447
Epoch 83/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0549 - r2_score: 0.6379 - val_loss:
0.0515 - val_r2_score: 0.6475
Epoch 84/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0442 - r2_score: 0.6974 - val_loss:
0.0531 - val_r2_score: 0.6363
Epoch 85/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0488 - r2_score: 0.6977 - val_loss:
0.0580 - val_r2_score: 0.6025
Epoch 86/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0563 - r2_score: 0.6569 - val_loss:
0.0523 - val_r2_score: 0.6420
Epoch 87/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0456 - r2_score: 0.7216 - val_loss:
0.0552 - val_r2_score: 0.6219
Epoch 88/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0481 - r2_score: 0.6833 - val_loss:
0.0549 - val_r2_score: 0.6236
Epoch 89/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0498 - r2_score: 0.6901 - val_loss:
0.0565 - val_r2_score: 0.6127
Epoch 90/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0512 - r2_score: 0.6767 - val_loss:
0.0526 - val_r2_score: 0.6401
Epoch 91/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0542 - r2_score: 0.6750 - val_loss:
0.0549 - val_r2_score: 0.6240
Epoch 92/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0567 - r2_score: 0.6456 - val_loss:
0.0532 - val_r2_score: 0.6354
Epoch 93/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0548 - r2_score: 0.6482 - val_loss:
0.0537 - val_r2_score: 0.6326
Epoch 94/100
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0528 - r2_score: 0.6865 - val_loss:
0.0524 - val_r2_score: 0.6416
Epoch 95/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0561 - r2_score: 0.6568 - val_loss:
0.0556 - val_r2_score: 0.6190
Epoch 96/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0548 - r2_score: 0.6611 - val_loss:
0.0539 - val_r2_score: 0.6309
Epoch 97/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0592 - r2_score: 0.6380 - val_loss:
0.0550 - val_r2_score: 0.6229
Epoch 98/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0595 - r2_score: 0.6337 - val_loss:
0.0516 - val_r2_score: 0.6473
Epoch 99/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.0519 - r2_score: 0.6589 - val_loss:
0.0518 - val_r2_score: 0.6460
Epoch 100/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.0501 - r2_score: 0.6694 - val_loss:
0.0536 - val_r2_score: 0.6332
CPU times: total: 7.08 s
Wall time: 19.3 s
```

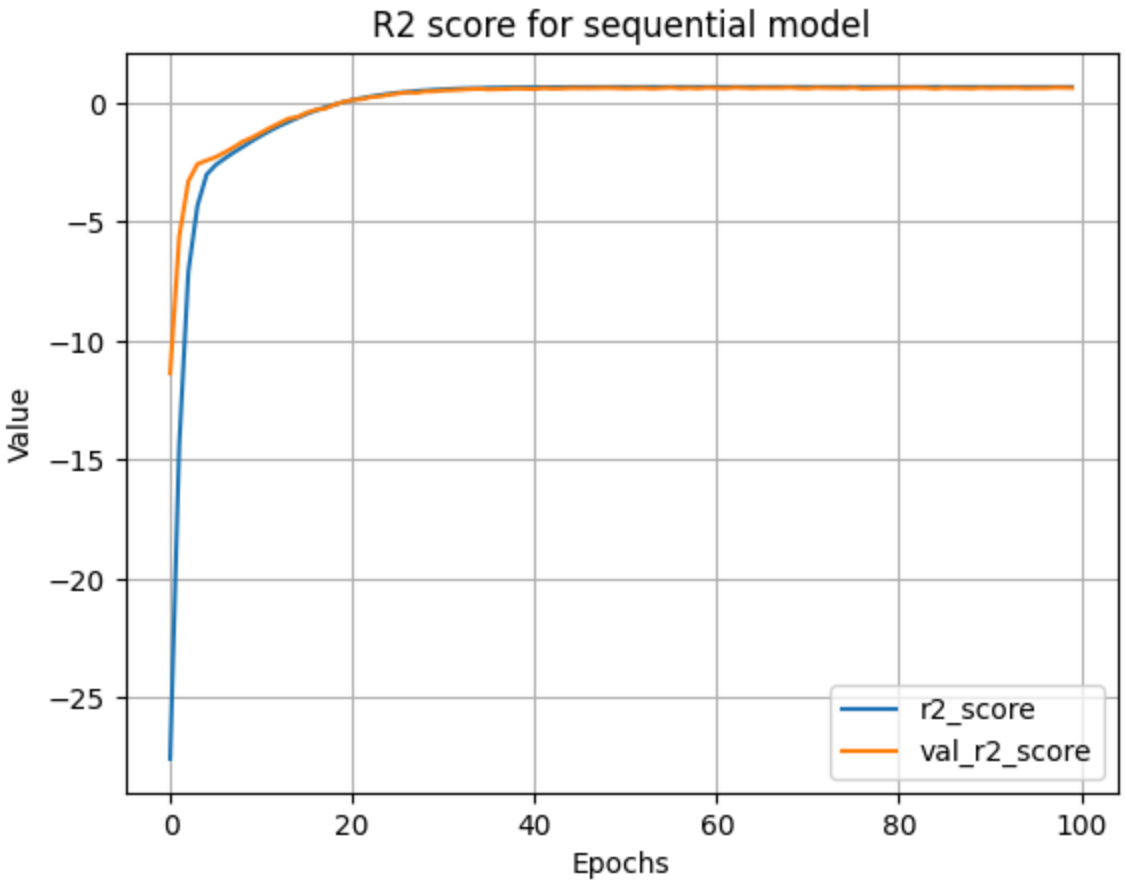
In [468…

```
plot_loss(phist)
```



In [469…

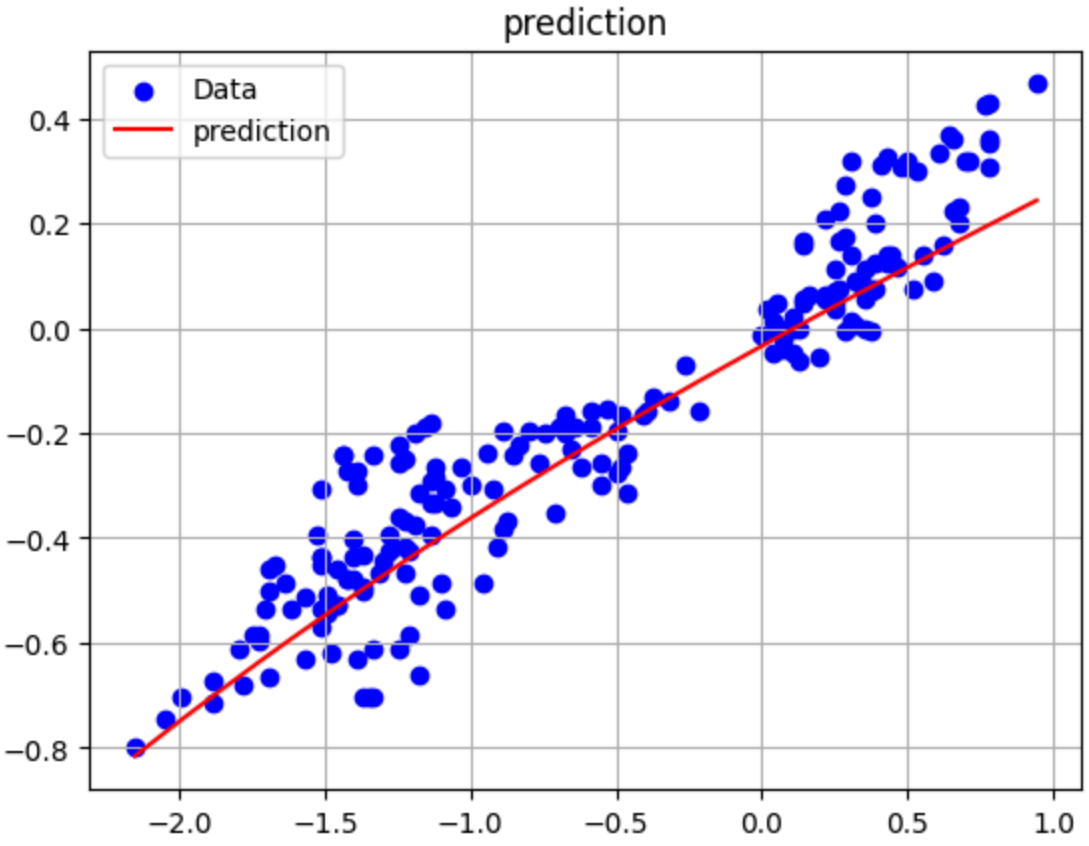
```
plot_metrics(phist)
```



In [470…
```
x_test = np.array([test['temp'].values, test['temp'].values**2, test['temp'].values
x_test = x_test.T
y_pol = pol_reg.predict(x_test)
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
```

In [471…
```
plt.scatter(test['temp'], test['temp_lower'], c='b', label='Data')
plt.plot(np.sort(test['temp']), y_pol[np.argsort(test['temp'])], c = 'r', label='pr
plt.title('prediction')
plt.legend()
plt.grid(True)
```



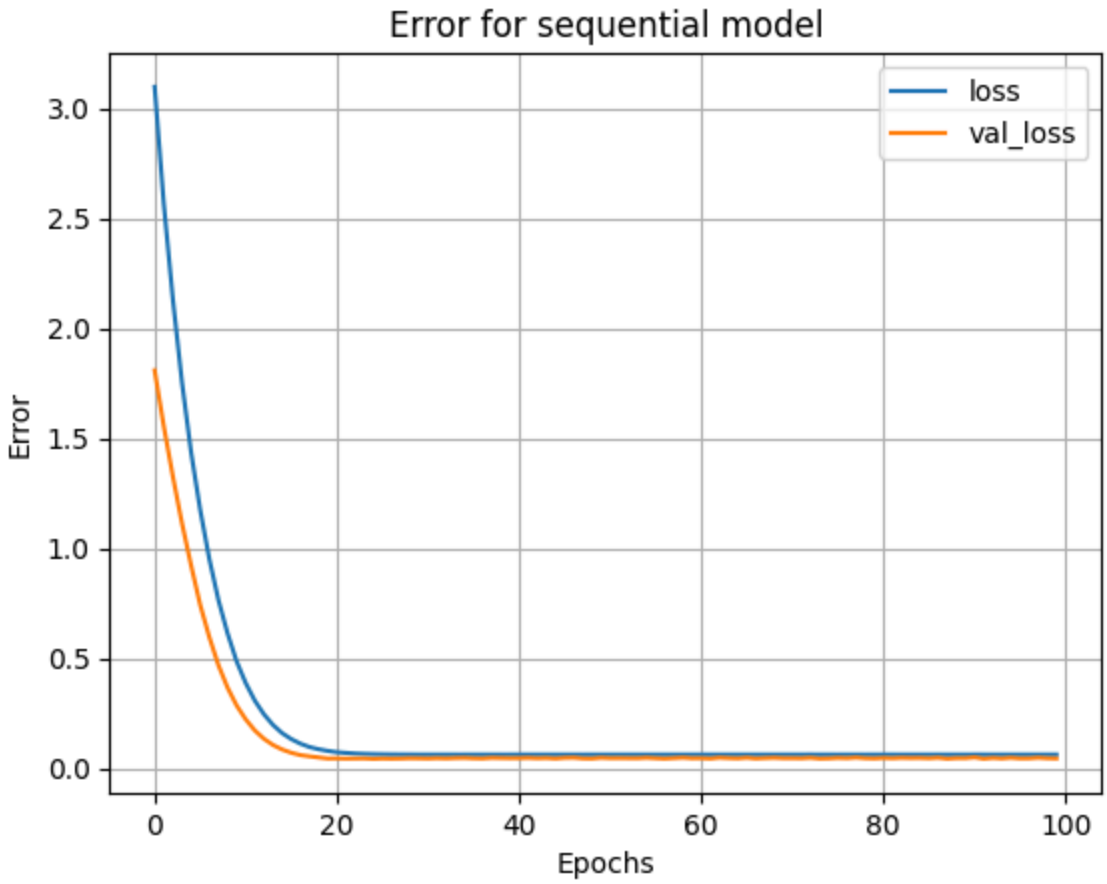
4. Визуализируйте точки набора данных на плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y – зависимый признак), а также линии линейной и полиномиальной регрессий (другими цветами), подписывая оси и рисунок и создавая легенду

!!!. Отдельно посмотреть оценку ошибки и обучаемости можно в третьем задании для каждой из регрессий

1. Оценка ошибки и обучаемости линейной регрессии

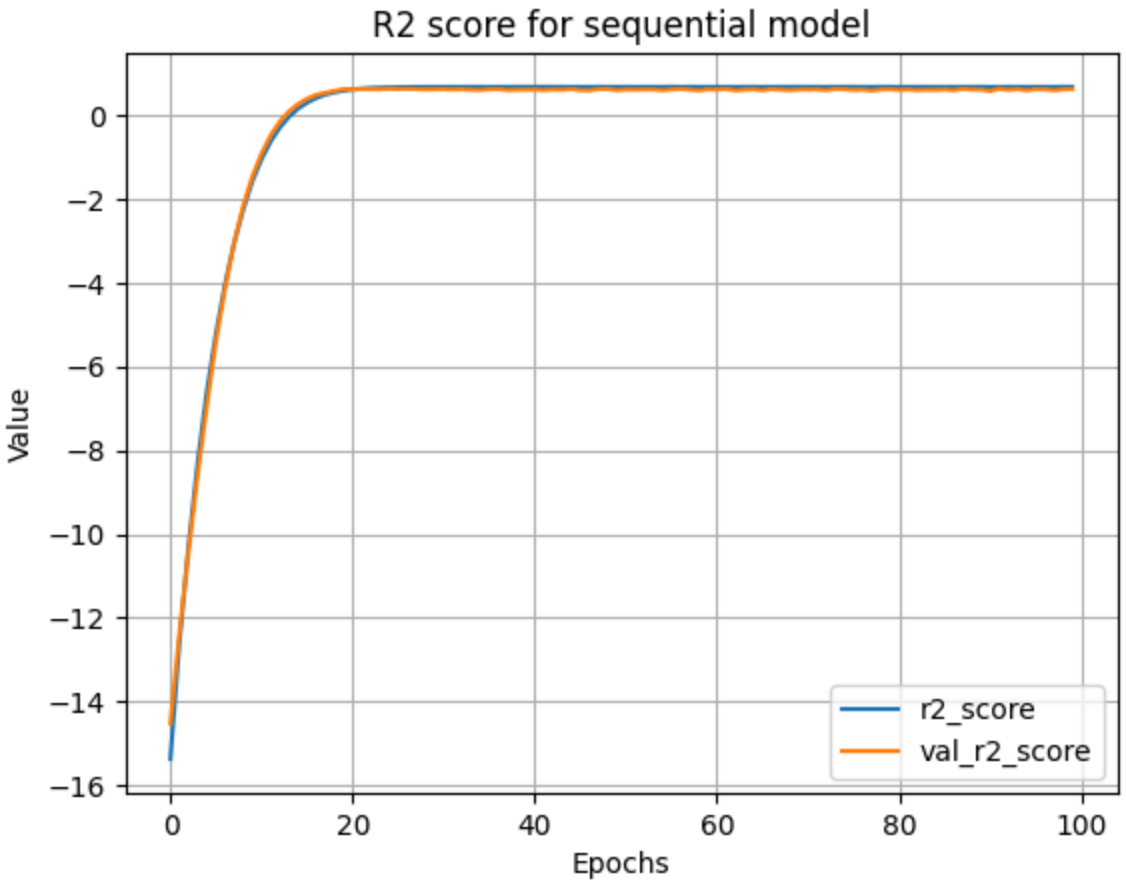
In [472…

```
plot_loss(hist)
```



In [473…

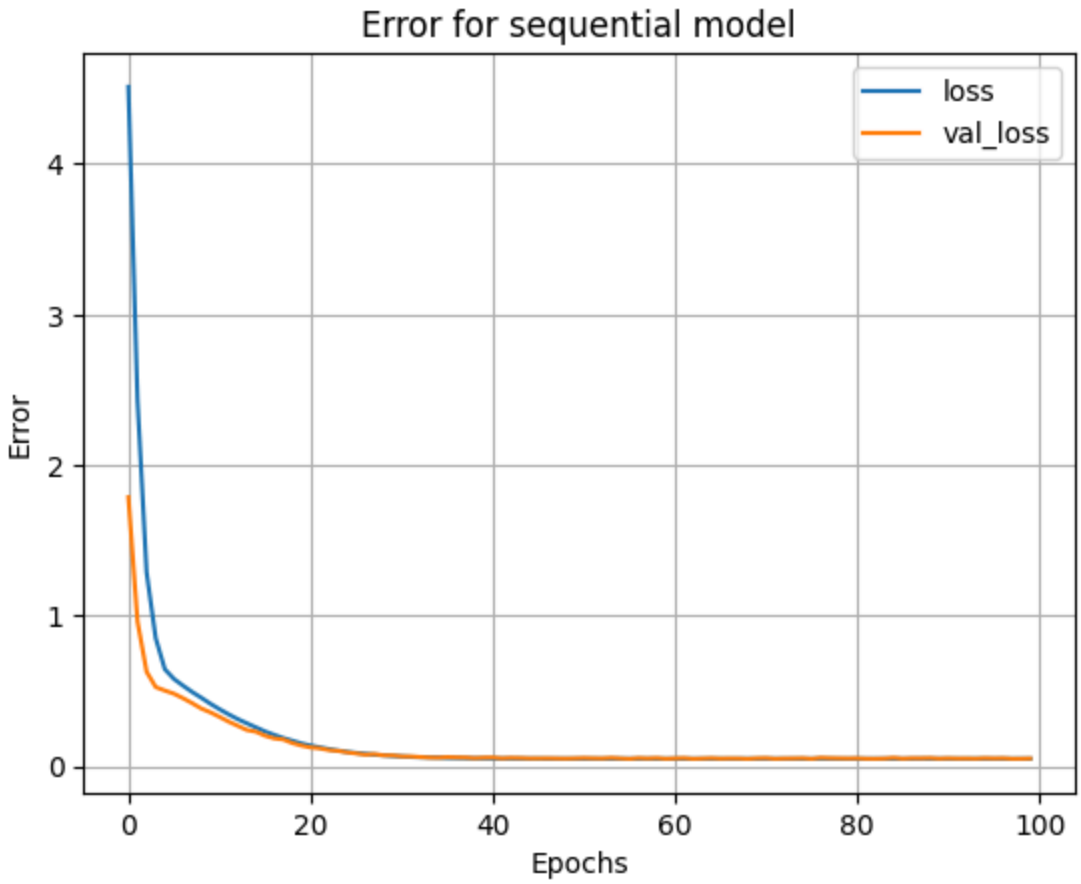
```
plot_metrics(hist)
```



2. Оценка ошибки и обучаемости полиномиальной регрессии

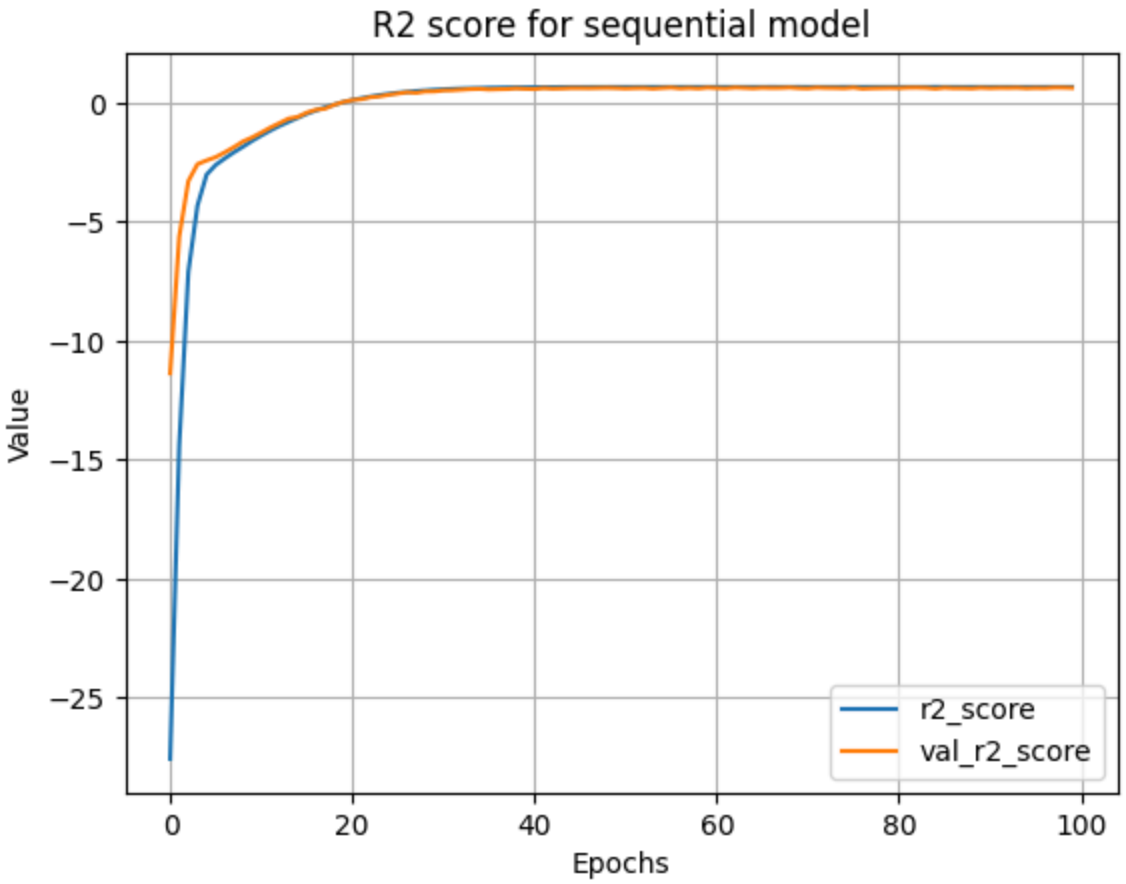
In [474…

```
plot_loss(phist)
```



In [475…

```
plot_metrics(phist)
```

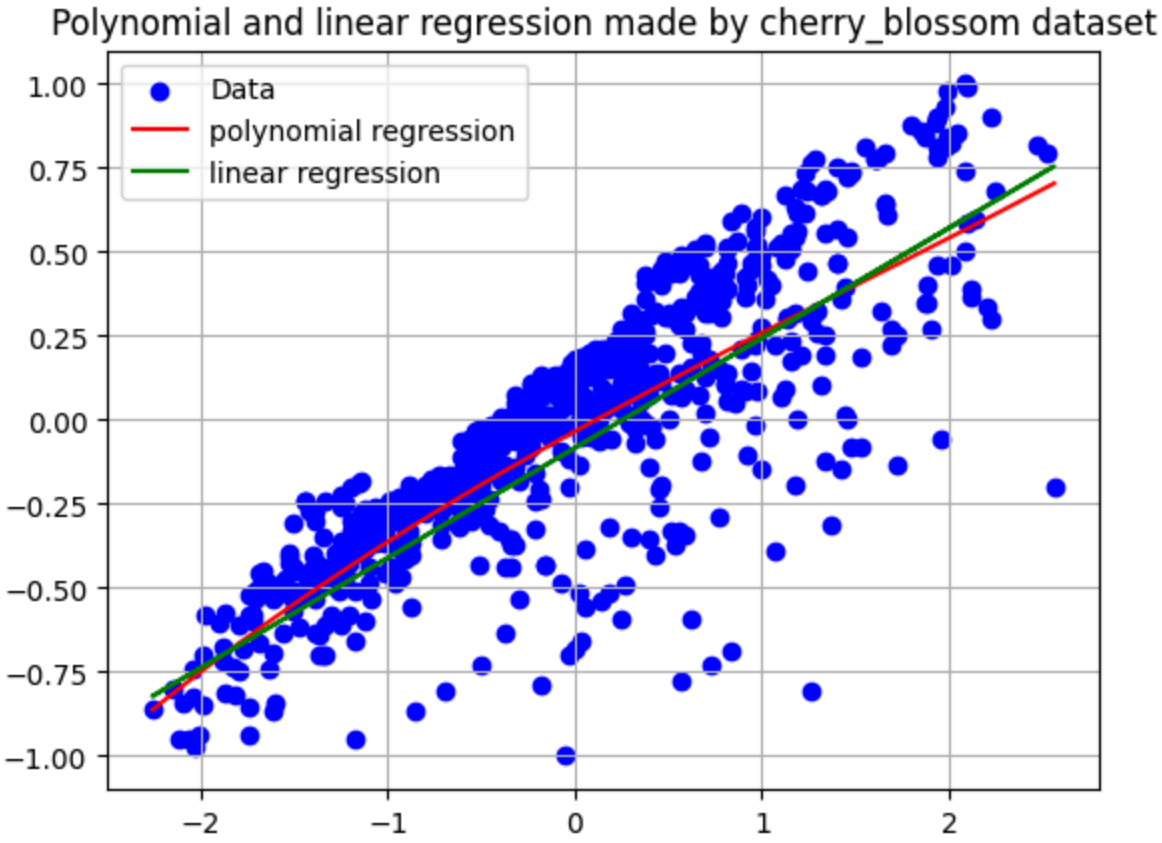


5. Визуализируйте точки набора данных на плоскости в виде диаграммы рассеяния (ось X – независимый признак, ось Y – зависимый признак), а также линии линейной и полиномиальной регрессий (другими цветами), подписывая оси и рисунок и создавая легенду

In [476…

```
y_lin_test = lin_reg.predict(train['temp'].values)
y_pol_test = pol_reg.predict(x)

plt.scatter(train['temp'], train['temp_lower'], c='b', label='Data')
plt.plot(np.sort(train['temp']), y_pol_test[np.argsort(train['temp'])], c = 'r', la
plt.plot(train['temp'], y_lin_test, c = 'g', label='linear regression')

plt.title('Polynomial and linear regression made by cherry_blossom dataset')
plt.legend()
plt.grid(True)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
```



6. Определите в исходном наборе данных признак (отличный от независимого и зависимого признаков), принимающий непрерывные значения и имеющий свойства, указанные в индивидуальном задании.

Доп. признак: имеющий минимальную корреляцию с независимой переменной

In [576…

```
df
```

Out[576…

	year	doy	temp	temp_upper	temp_lower
50	851	108.0	7.38	12.10	2.66
63	864	100.0	6.42	8.69	4.14
65	866	106.0	6.44	8.11	4.77
88	889	104.0	6.83	8.48	5.19
90	891	109.0	6.98	8.96	5.00
…	…	…	…	…	…
1175	1976	99.0	8.20	8.77	7.63
1176	1977	93.0	8.22	8.78	7.66
1177	1978	104.0	8.20	8.78	7.61
1178	1979	97.0	8.28	8.83	7.73
1179	1980	102.0	8.30	8.86	7.74

787 rows × 5 columns

In [578…
```
df.corr()['temp']
```

Out[578…
```
year          0.028033
doy          -0.326976
temp          1.000000
temp_upper    0.876747
temp_lower    0.858841
Name: temp, dtype: float64
```

In [580…
```
df.corr()['temp'].min()
```

Out[580…
```
-0.3269757034361482
```

Следовательно, выбираем признак 'doy'

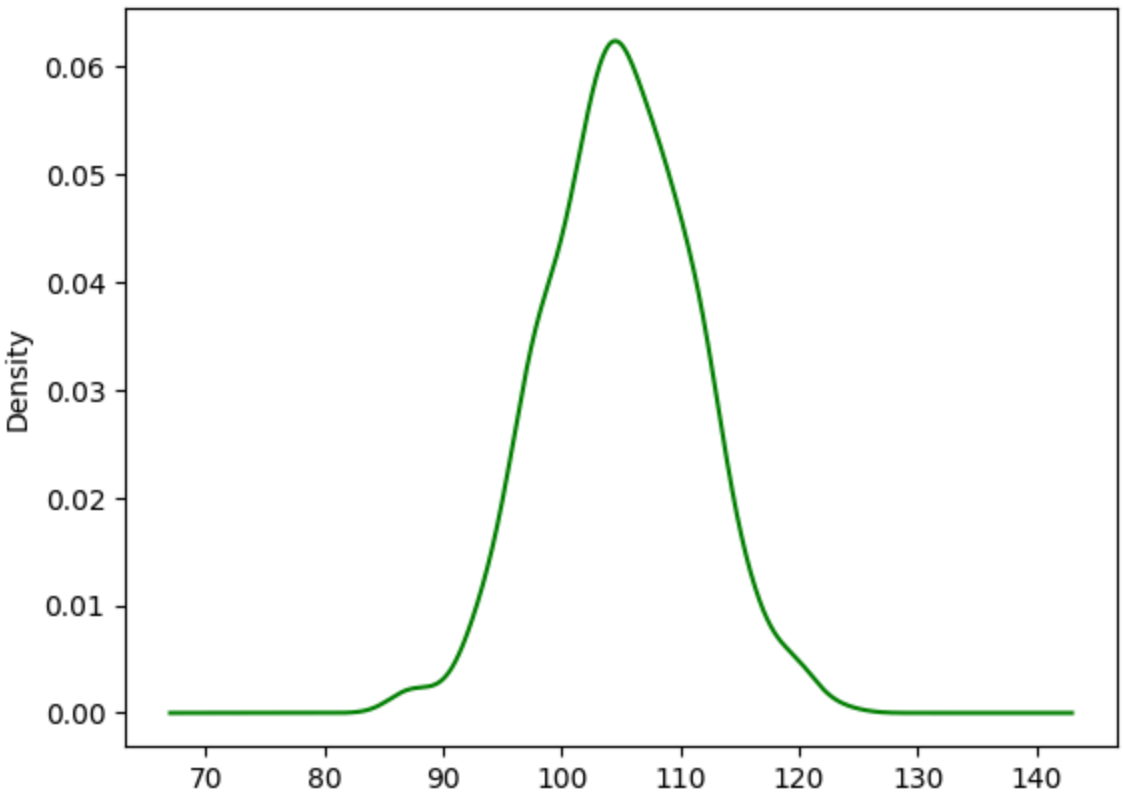
7. Стандартизуйте этот признак и визуализируйте его в соответствии с индивидуальным заданием.

Доп. признак: имеющий минимальную корреляцию с независимой переменной

Визуализация доп. признака – эмпирическая плотность распределения

In [581…
```
# plt.scatter(df['temp_lower'], df['doy'])
df.doy.plot.density(color='green')
```

Out[581… <Axes: ylabel='Density'>



In [582…

```
# df['doy'].mean()
tmp = df.copy()
tmp.reset_index(inplace=True, drop=True)
tmp
```

Out[582…

	year	doy	temp	temp_upper	temp_lower
0	851	108.0	7.38	12.10	2.66
1	864	100.0	6.42	8.69	4.14
2	866	106.0	6.44	8.11	4.77
3	889	104.0	6.83	8.48	5.19
4	891	109.0	6.98	8.96	5.00
...	...	...	...	...	...
782	1976	99.0	8.20	8.77	7.63
783	1977	93.0	8.22	8.78	7.66
784	1978	104.0	8.20	8.78	7.61
785	1979	97.0	8.28	8.83	7.73
786	1980	102.0	8.30	8.86	7.74

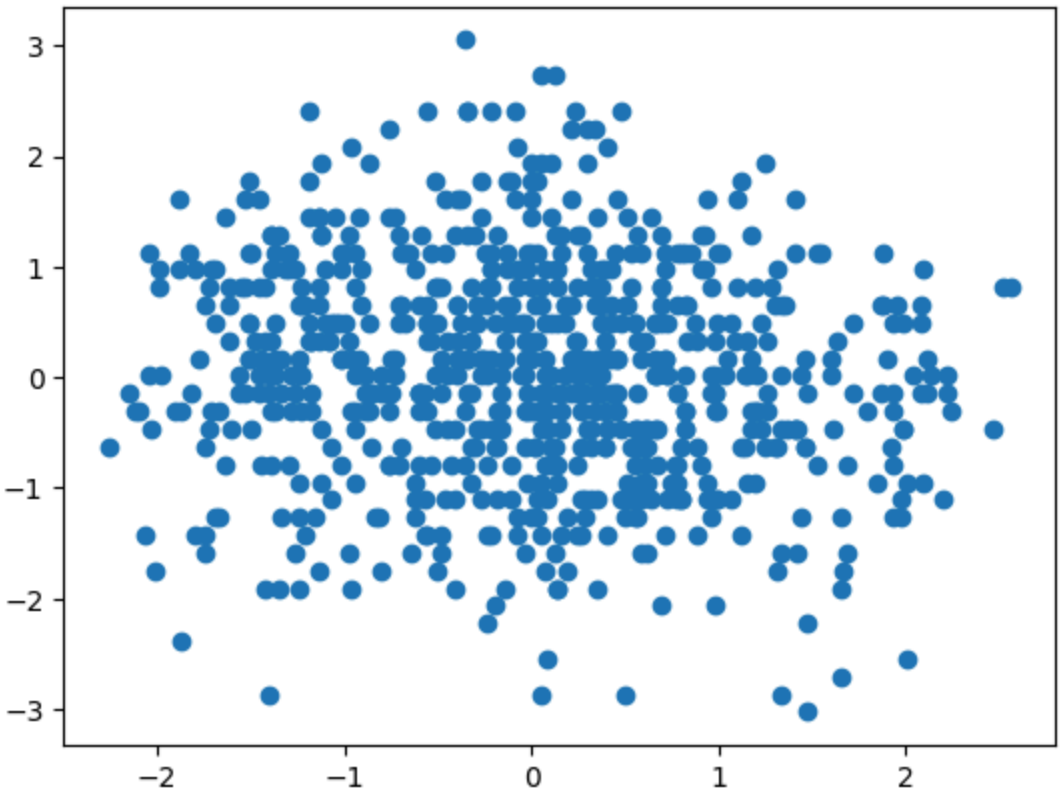
787 rows × 5 columns

In [583…
```
ds['doy'] = (tmp['doy'] - tmp['doy'].mean()) / tmp['doy'].std(ddof=0)
# df['doy'].std()
# ds = ds.dropna().copy()
# display(ds)
print(ds.isna().sum())
# ds
```

```
temp          0
temp_lower    0
doy           0
dtype: int64
```

In [584…
```
# ds[700:734]
plt.scatter(ds['temp'], ds['doy'])
```

Out[584… `<matplotlib.collections.PathCollection at 0x2965800f080>`



8. Сформируйте набор входных данных из двух стандартизованных признаков набора данных (независимый признак и определенный признак), постройте нейронную сеть (нелинейный регресор) с количеством скрытых слоев, количеством нейронов и функцией активации, указанными в

индивидуальном задании, и одним нейроном в выходном слое и обучите ее на наборе данных из двух признаков и отклика. Отследите обучение нейронной сети, изменяя, при необходимости, гиперпараметры (функцию потерь, оптимизатор, шаг обучения и т.п.) или применяя регуляризацию.

Показатель качества регрессии – MSE (mean squared error)

Степень полинома: 3

Параметры глубокой нейронной сети: кол-во скрытых слоев – 3, кол-во нейронов в скрытом слое – 128, функция активации – сигмоида.

In [636…
```
X = np.array(ds[['temp', 'doy']])
feature_normalizer = tf.keras.layers.Normalization(axis=None, input_shape=(2, ))
feature_normalizer.adapt(X)
```

```
C:\Python312\Lib\site-packages\keras\src\layers\preprocessing\normalization.py:99: U
serWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using
Sequential models, prefer using an `Input(shape)` object as the first layer in the m
odel instead.
  super().__init__(**kwargs)
```

In [637…
```
uni_model = tf.keras.Sequential([
    feature_normalizer,
    # tf.keras.Input(shape=(1, )),
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=1)
])
```

In [638…
```
uni_model.summary()
```

```
Model: "sequential_37"
```

Layer (type)	Output Shape	
normalization_24 (Normalization)	(None, 2)	
dense_88 (Dense)	(None, 128)	
dense_89 (Dense)	(None, 128)	
dense_90 (Dense)	(None, 128)	
dense_91 (Dense)	(None, 1)	

Total params: 33,540 (131.02 KB)

Trainable params: 33,537 (131.00 KB)

Non-trainable params: 3 (16.00 B)

In [639…

```
uni_model.compile(
    loss = 'mse',
    optimizer=tf.optimizers.Adam(learning_rate=0.01),
    metrics=['r2_score']
)
```

In [640…

```
uhist = uni_model.fit(
    X, ds['temp_lower'],
    epochs = 100,
    verbose = 1,
    validation_split = 0.2
)
```

```
Epoch 1/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 4s 33ms/step - loss: 4.1410 - r2_score: -10.7216 - val_lo
ss: 1.9583 - val_r2_score: -0.0030
Epoch 2/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 1.7572 - r2_score: -0.5008 - val_los
s: 0.8925 - val_r2_score: 6.4528e-04
Epoch 3/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.8044 - r2_score: -0.1798 - val_los
s: 0.5105 - val_r2_score: -0.2773
Epoch 4/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4443 - r2_score: -0.0150 - val_los
s: 0.3257 - val_r2_score: -0.0938
Epoch 5/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.3026 - r2_score: -0.0509 - val_los
s: 0.2594 - val_r2_score: -0.1787
Epoch 6/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.2778 - r2_score: -0.3274 - val_los
s: 0.2234 - val_r2_score: -0.1691
Epoch 7/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.2143 - r2_score: -0.0309 - val_los
s: 0.2631 - val_r2_score: -0.6431
Epoch 8/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.2099 - r2_score: -0.1930 - val_los
s: 0.1657 - val_r2_score: -0.0171
Epoch 9/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2163 - r2_score: -0.2520 - val_los
s: 0.1832 - val_r2_score: -0.1833
Epoch 10/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1705 - r2_score: -0.0539 - val_los
s: 0.1645 - val_r2_score: -0.1102
Epoch 11/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.2580 - r2_score: -0.4662 - val_los
s: 0.3007 - val_r2_score: -0.9537
Epoch 12/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1959 - r2_score: -0.1399 - val_los
s: 0.1966 - val_r2_score: -0.3470
Epoch 13/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1780 - r2_score: -0.0714 - val_los
s: 0.1466 - val_r2_score: -0.0088
Epoch 14/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.2001 - r2_score: -0.2301 - val_los
s: 0.2464 - val_r2_score: -0.6849
Epoch 15/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1916 - r2_score: -0.2338 - val_los
s: 0.1540 - val_r2_score: -0.0514
Epoch 16/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1643 - r2_score: -0.0740 - val_los
s: 0.1557 - val_r2_score: -0.0675
Epoch 17/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.2038 - r2_score: -0.1653 - val_los
s: 0.1483 - val_r2_score: -0.0012
Epoch 18/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1779 - r2_score: -0.1645 - val_los
s: 0.2734 - val_r2_score: -0.8999
Epoch 19/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1874 - r2_score: -0.2232 - val_los
```

```
s: 0.1507 - val_r2_score: -0.0181
Epoch 20/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1732 - r2_score: -0.1309 - val_los
s: 0.1522 - val_r2_score: -0.0479
Epoch 21/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.2226 - r2_score: -0.2804 - val_los
s: 0.2564 - val_r2_score: -0.7650
Epoch 22/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1706 - r2_score: -0.0955 - val_los
s: 0.1440 - val_r2_score: -1.4865e-04
Epoch 23/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1948 - r2_score: -0.1483 - val_los
s: 0.1797 - val_r2_score: -0.2405
Epoch 24/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1623 - r2_score: -0.0653 - val_los
s: 0.1645 - val_r2_score: -0.1449
Epoch 25/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1591 - r2_score: -0.0169 - val_los
s: 0.1449 - val_r2_score: -0.0076
Epoch 26/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1580 - r2_score: -0.0425 - val_los
s: 0.1942 - val_r2_score: -0.3437
Epoch 27/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1674 - r2_score: -0.0439 - val_los
s: 0.2116 - val_r2_score: -0.4736
Epoch 28/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1677 - r2_score: -0.0379 - val_los
s: 0.2121 - val_r2_score: -0.4770
Epoch 29/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1951 - r2_score: -0.1022 - val_los
s: 0.2058 - val_r2_score: -0.4141
Epoch 30/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.2168 - r2_score: -0.2816 - val_los
s: 0.1513 - val_r2_score: -0.0378
Epoch 31/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1809 - r2_score: -0.1112 - val_los
s: 0.1531 - val_r2_score: -0.0528
Epoch 32/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.2016 - r2_score: -0.2018 - val_los
s: 0.1466 - val_r2_score: -0.0146
Epoch 33/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1644 - r2_score: -0.0462 - val_los
s: 0.1960 - val_r2_score: -0.3582
Epoch 34/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1856 - r2_score: -0.2288 - val_los
s: 0.2605 - val_r2_score: -0.7641
Epoch 35/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1879 - r2_score: -0.1497 - val_los
s: 0.1531 - val_r2_score: -0.0621
Epoch 36/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1522 - r2_score: -0.0423 - val_los
s: 0.2245 - val_r2_score: -0.5616
Epoch 37/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1688 - r2_score: -0.1072 - val_los
s: 0.1456 - val_r2_score: -0.0074
Epoch 38/100
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1695 - r2_score: -0.0786 - val_los
s: 0.1509 - val_r2_score: -0.0496
Epoch 39/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1881 - r2_score: -0.1796 - val_los
s: 0.2287 - val_r2_score: -0.5786
Epoch 40/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1676 - r2_score: -0.0585 - val_los
s: 0.2408 - val_r2_score: -0.6727
Epoch 41/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1579 - r2_score: -0.0595 - val_los
s: 0.1756 - val_r2_score: -0.2209
Epoch 42/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1784 - r2_score: -0.1503 - val_los
s: 0.1459 - val_r2_score: -2.2614e-04
Epoch 43/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1815 - r2_score: -0.1479 - val_los
s: 0.1495 - val_r2_score: -0.0313
Epoch 44/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1928 - r2_score: -0.2069 - val_los
s: 0.1505 - val_r2_score: -0.0123
Epoch 45/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.2374 - r2_score: -0.5517 - val_los
s: 0.1532 - val_r2_score: -0.0167
Epoch 46/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1997 - r2_score: -0.2935 - val_los
s: 0.1473 - val_r2_score: -0.0015
Epoch 47/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1754 - r2_score: -0.0719 - val_los
s: 0.2022 - val_r2_score: -0.4023
Epoch 48/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1646 - r2_score: -0.1244 - val_los
s: 0.2273 - val_r2_score: -0.5764
Epoch 49/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1560 - r2_score: -0.0206 - val_los
s: 0.1780 - val_r2_score: -0.2389
Epoch 50/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1716 - r2_score: -0.0219 - val_los
s: 0.2488 - val_r2_score: -0.7327
Epoch 51/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1849 - r2_score: -0.1933 - val_los
s: 0.1695 - val_r2_score: -0.1778
Epoch 52/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1717 - r2_score: -0.0382 - val_los
s: 0.2051 - val_r2_score: -0.4280
Epoch 53/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1774 - r2_score: -0.1290 - val_los
s: 0.1580 - val_r2_score: -0.0921
Epoch 54/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1972 - r2_score: -0.1933 - val_los
s: 0.1590 - val_r2_score: -0.1021
Epoch 55/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1578 - r2_score: -0.0138 - val_los
s: 0.1495 - val_r2_score: -0.0405
Epoch 56/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1715 - r2_score: -0.0523 - val_los
s: 0.1498 - val_r2_score: -0.0435
```

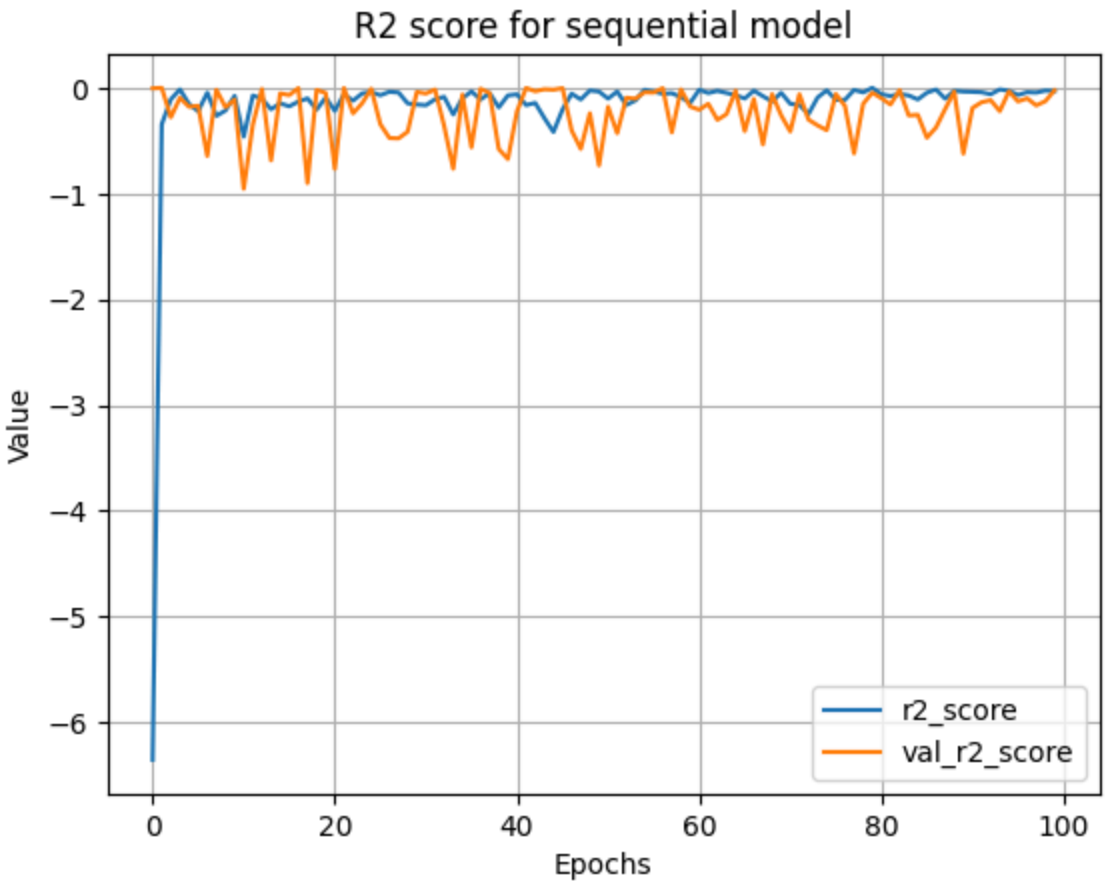
```
Epoch 57/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1865 - r2_score: -0.0766 - val_los
s: 0.1439 - val_r2_score: -0.0014
Epoch 58/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1707 - r2_score: -0.0761 - val_los
s: 0.2040 - val_r2_score: -0.4208
Epoch 59/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1648 - r2_score: -0.0813 - val_los
s: 0.1459 - val_r2_score: -0.0138
Epoch 60/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.2097 - r2_score: -0.2194 - val_los
s: 0.1695 - val_r2_score: -0.1752
Epoch 61/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1669 - r2_score: -0.0113 - val_los
s: 0.1733 - val_r2_score: -0.2063
Epoch 62/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1607 - r2_score: -0.0456 - val_los
s: 0.1651 - val_r2_score: -0.1501
Epoch 63/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1590 - r2_score: -0.0334 - val_los
s: 0.1869 - val_r2_score: -0.3020
Epoch 64/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1653 - r2_score: -0.0389 - val_los
s: 0.1792 - val_r2_score: -0.2477
Epoch 65/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1634 - r2_score: -0.0223 - val_los
s: 0.1477 - val_r2_score: -0.0253
Epoch 66/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1840 - r2_score: -0.1805 - val_los
s: 0.2033 - val_r2_score: -0.4088
Epoch 67/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1683 - r2_score: -0.0265 - val_los
s: 0.1586 - val_r2_score: -0.1032
Epoch 68/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1776 - r2_score: -0.0712 - val_los
s: 0.2206 - val_r2_score: -0.5348
Epoch 69/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1830 - r2_score: -0.2172 - val_los
s: 0.1529 - val_r2_score: -0.0579
Epoch 70/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1625 - r2_score: -0.0505 - val_los
s: 0.1809 - val_r2_score: -0.2593
Epoch 71/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1729 - r2_score: -0.0950 - val_los
s: 0.2039 - val_r2_score: -0.4121
Epoch 72/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1916 - r2_score: -0.1492 - val_los
s: 0.1552 - val_r2_score: -0.0592
Epoch 73/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.2023 - r2_score: -0.2888 - val_los
s: 0.1912 - val_r2_score: -0.2998
Epoch 74/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.1685 - r2_score: -0.0826 - val_los
s: 0.1963 - val_r2_score: -0.3543
Epoch 75/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1606 - r2_score: -0.0260 - val_los
```

```
s: 0.2010 - val_r2_score: -0.3996
Epoch 76/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1619 - r2_score: -0.0819 - val_los
s: 0.1527 - val_r2_score: -0.0577
Epoch 77/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.1831 - r2_score: -0.1235 - val_los
s: 0.1698 - val_r2_score: -0.1699
Epoch 78/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1555 - r2_score: -0.0250 - val_los
s: 0.2327 - val_r2_score: -0.6199
Epoch 79/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1652 - r2_score: -0.0642 - val_los
s: 0.1653 - val_r2_score: -0.1508
Epoch 80/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1578 - r2_score: -0.0080 - val_los
s: 0.1499 - val_r2_score: -0.0443
Epoch 81/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1722 - r2_score: -0.0642 - val_los
s: 0.1576 - val_r2_score: -0.0967
Epoch 82/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1584 - r2_score: -0.0953 - val_los
s: 0.1667 - val_r2_score: -0.1570
Epoch 83/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1523 - r2_score: -0.0408 - val_los
s: 0.1468 - val_r2_score: -0.0203
Epoch 84/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1649 - r2_score: -0.1534 - val_los
s: 0.1812 - val_r2_score: -0.2597
Epoch 85/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1759 - r2_score: -0.1133 - val_los
s: 0.1815 - val_r2_score: -0.2547
Epoch 86/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1672 - r2_score: -0.0359 - val_los
s: 0.2114 - val_r2_score: -0.4709
Epoch 87/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1570 - r2_score: -0.0216 - val_los
s: 0.1971 - val_r2_score: -0.3736
Epoch 88/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1742 - r2_score: -0.0836 - val_los
s: 0.1729 - val_r2_score: -0.1995
Epoch 89/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1780 - r2_score: -0.0315 - val_los
s: 0.1486 - val_r2_score: -0.0335
Epoch 90/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1866 - r2_score: -0.0536 - val_los
s: 0.2329 - val_r2_score: -0.6224
Epoch 91/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1700 - r2_score: -0.0447 - val_los
s: 0.1708 - val_r2_score: -0.1893
Epoch 92/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1623 - r2_score: -0.0309 - val_los
s: 0.1632 - val_r2_score: -0.1352
Epoch 93/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1776 - r2_score: -0.0485 - val_los
s: 0.1604 - val_r2_score: -0.1159
Epoch 94/100
```

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1660 - r2_score: -0.0125 - val_los
s: 0.1750 - val_r2_score: -0.2196
Epoch 95/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1722 - r2_score: -0.0216 - val_los
s: 0.1482 - val_r2_score: -0.0320
Epoch 96/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1716 - r2_score: -0.0645 - val_los
s: 0.1622 - val_r2_score: -0.1278
Epoch 97/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1677 - r2_score: -0.0535 - val_los
s: 0.1578 - val_r2_score: -0.0985
Epoch 98/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1653 - r2_score: -0.0436 - val_los
s: 0.1674 - val_r2_score: -0.1652
Epoch 99/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.1562 - r2_score: -0.0364 - val_los
s: 0.1616 - val_r2_score: -0.1257
Epoch 100/100
19/19 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1680 - r2_score: -0.0234 - val_los
s: 0.1470 - val_r2_score: -0.0229
```

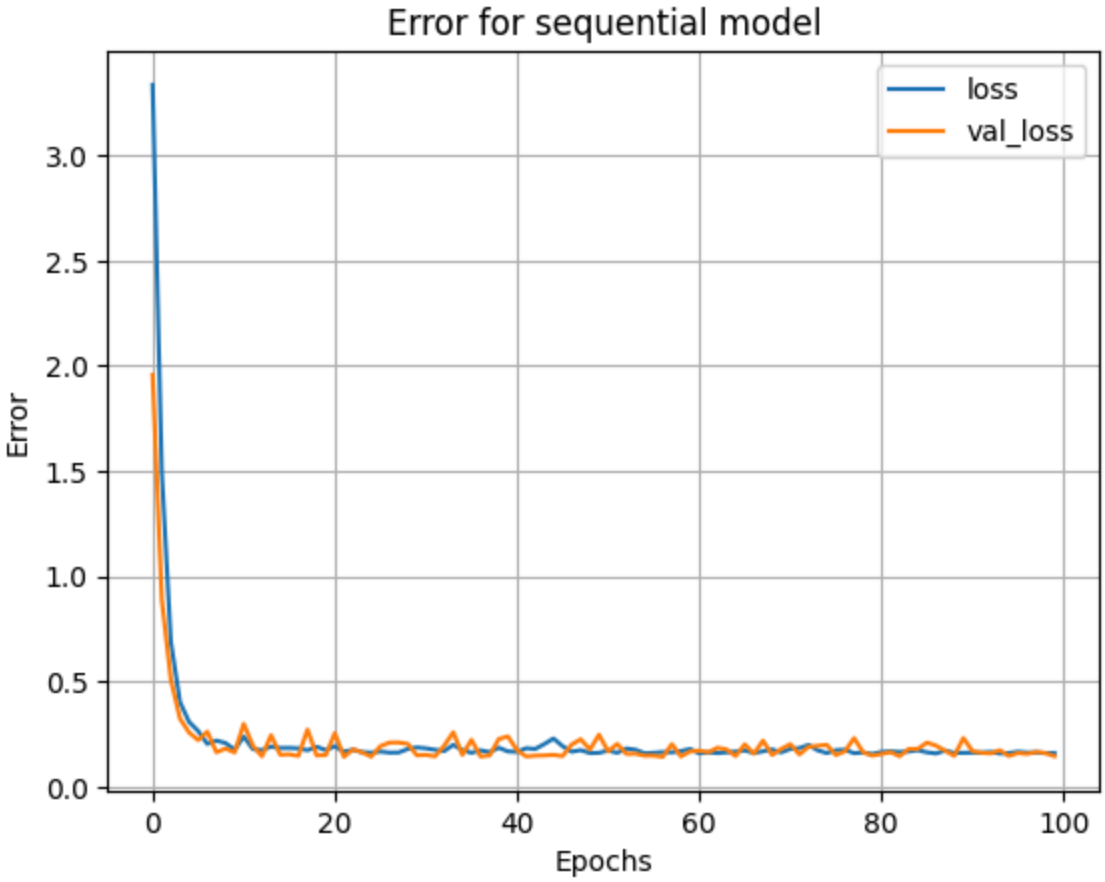
In [641…

```
plot_metrics(uhist)
```



In [642…

```
plot_loss(uhist)
```



9. Визуализируйте набор данных в виде диаграммы рассеяния и прогноз нейронной сети в виде поверхности в трехмерном пространстве, подписывая оси и рисунок.

In [643…
```
from matplotlib import cm
```

In [644…
```
n_plot = 51

x_plot = np.linspace(np.min(xs), np.max(xs), n_plot)
y_plot = np.linspace(np.min(ys), np.max(ys), n_plot)
x_mesh, y_mesh = np.meshgrid(x_plot, y_plot)
x_mesh.shape, y_mesh.shape
```

Out[644…
```
((51, 51), (51, 51))
```

In [645…
```
x_plot2 = np.reshape(x_mesh, [n_plot**2,1])
y_plot2 = np.reshape(y_mesh, [n_plot**2,1])
xy_2 = np.hstack([x_plot2, y_plot2])
xy_2.shape
```

Out[645… (2601, 2)

In [684…
```
z = uni_model.predict(xy_2)
z.shape
xy_2.shape
```

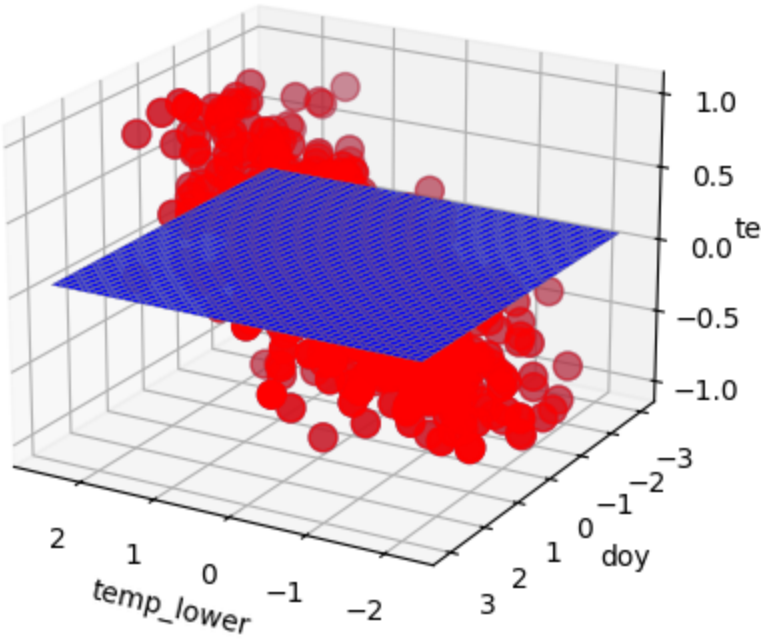
```
82/82 ━━━━━━━━━━━━━━━━━━━━ 0s 846us/step
```

Out[684… (2601, 2)

In [647…
```
z_mesh = z.reshape((n_plot, n_plot))
z_mesh.shape
```

Out[647… (51, 51)

In [649…
```
# plt.scatter(ds['temp'], ds['doy'])
# plt.plot(ds['temp'], y_uhat, c = 'r')
fig = plt.figure()
ax = plt.axes(projection='3d')

xs = X[:, 0]
ys = X[:, 1]
zs = ds['temp_lower']

ax.scatter(xs, ys, zs, s = 100)

ax.set_xlabel('temp_lower')
ax.set_ylabel('doy')
ax.set_zlabel('temp')

ax.plot_surface(x_mesh, y_mesh, z_mesh, linewidth = 0.25, cstride=1, rstride=1,
                cmap=cm.winter, antialiased=True, edgecolors='gray')
ax.scatter( xs, ys, zs, s=100, c='r' )

ax.view_init(azim = 120, elev = 20)
```



К сожалению, вывод удручающий, так как для такой диаграммы рассеяния(см. решение задания 7) метрика обучаемости R^2 всегда будет находится справа от нуля, так как модель не может найти способа предсказывать лучше графика y = m, что заметно при обучении модели(показатель R2 все время приближается к 0).

10. Разбейте набор данных из двух признаков и отклика на обучающую и тестовую выборки и постройте кривые обучения для заданного показателя качества в зависимости от количества точек в обучающей выборке, подписывая оси и рисунок и создавая легенду.

In [655…

```
ds.shape
train = ds.iloc[:534, :]
```

```
test = ds.iloc[534: , :]
```

In [669…
```
xy = np.array(train[['temp', 'doy']])
feature_normalizer = tf.keras.layers.Normalization(axis=None)
feature_normalizer.adapt(xy)
```

In [670…
```
uni_model = tf.keras.Sequential([
    tf.keras.Input(shape=(2, )),
    feature_normalizer,
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=128, activation='sigmoid',
                          kernel_regularizer = tf.keras.regularizers.L2(l2=0.01)),
    tf.keras.layers.Dense(units=1)
])
```

In [671…
```
uni_model.summary()
```

Model: "sequential_41"

Layer (type)	Output Shape	
normalization_27 (Normalization)	(None, 2)	
dense_104 (Dense)	(None, 128)	
dense_105 (Dense)	(None, 128)	
dense_106 (Dense)	(None, 128)	
dense_107 (Dense)	(None, 1)	

Total params: 33,540 (131.02 KB)

Trainable params: 33,537 (131.00 KB)

Non-trainable params: 3 (16.00 B)

In [672…
```
uni_model.compile(
    loss = 'mse',
    optimizer=tf.optimizers.Adam(learning_rate=0.01),
    metrics=['r2_score']
)
```

In [694…
```
uhist = uni_model.fit(
    xy, train['temp_lower'],
    epochs = 100,
    verbose = 1,
    validation_split = 0.2
)
```

```
Epoch 1/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.1937 - r2_score: -0.0364 - val_los
s: 0.1003 - val_r2_score: -0.7649
Epoch 2/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1904 - r2_score: -0.0194 - val_los
s: 0.0576 - val_r2_score: -0.0151
Epoch 3/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2146 - r2_score: -0.0826 - val_los
s: 0.0585 - val_r2_score: -0.0201
Epoch 4/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2286 - r2_score: -0.1739 - val_los
s: 0.1402 - val_r2_score: -1.3926
Epoch 5/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2238 - r2_score: -0.1561 - val_los
s: 0.1923 - val_r2_score: -2.3554
Epoch 6/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2108 - r2_score: -0.0824 - val_los
s: 0.0685 - val_r2_score: -0.1988
Epoch 7/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2097 - r2_score: -0.0574 - val_los
s: 0.0576 - val_r2_score: -0.0055
Epoch 8/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2120 - r2_score: -0.0998 - val_los
s: 0.0846 - val_r2_score: -0.4799
Epoch 9/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1900 - r2_score: -0.0608 - val_los
s: 0.1776 - val_r2_score: -2.1317
Epoch 10/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2414 - r2_score: -0.1944 - val_los
s: 0.1601 - val_r2_score: -1.7757
Epoch 11/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1877 - r2_score: -0.0219 - val_los
s: 0.0714 - val_r2_score: -0.2495
Epoch 12/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2161 - r2_score: -0.0186 - val_los
s: 0.0837 - val_r2_score: -0.4770
Epoch 13/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1956 - r2_score: -0.0222 - val_los
s: 0.0717 - val_r2_score: -0.2658
Epoch 14/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1875 - r2_score: -0.0407 - val_los
s: 0.0702 - val_r2_score: -0.2382
Epoch 15/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1918 - r2_score: -0.0253 - val_los
s: 0.0858 - val_r2_score: -0.5152
Epoch 16/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2024 - r2_score: -0.0182 - val_los
s: 0.0813 - val_r2_score: -0.4357
Epoch 17/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1914 - r2_score: -0.0338 - val_los
s: 0.1690 - val_r2_score: -1.9812
Epoch 18/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.2073 - r2_score: -0.0781 - val_los
s: 0.0598 - val_r2_score: -0.0496
Epoch 19/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2116 - r2_score: -0.0808 - val_los
```

```
s: 0.1036 - val_r2_score: -0.8193
Epoch 20/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1966 - r2_score: -0.0430 - val_los
s: 0.1256 - val_r2_score: -1.2137
Epoch 21/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2103 - r2_score: -0.0659 - val_los
s: 0.0795 - val_r2_score: -0.3933
Epoch 22/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1960 - r2_score: -0.0152 - val_los
s: 0.0633 - val_r2_score: -0.1162
Epoch 23/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2308 - r2_score: -0.1747 - val_los
s: 0.0635 - val_r2_score: -0.0403
Epoch 24/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2197 - r2_score: -0.0998 - val_los
s: 0.0764 - val_r2_score: -0.3136
Epoch 25/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2019 - r2_score: -0.0590 - val_los
s: 0.0771 - val_r2_score: -0.3565
Epoch 26/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2057 - r2_score: -0.0347 - val_los
s: 0.0969 - val_r2_score: -0.7119
Epoch 27/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1840 - r2_score: -0.0550 - val_los
s: 0.0622 - val_r2_score: -0.0915
Epoch 28/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1945 - r2_score: -0.0330 - val_los
s: 0.0789 - val_r2_score: -0.3887
Epoch 29/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2065 - r2_score: -0.0276 - val_los
s: 0.1142 - val_r2_score: -1.0167
Epoch 30/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2091 - r2_score: -0.0318 - val_los
s: 0.1708 - val_r2_score: -2.0134
Epoch 31/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2197 - r2_score: -0.1337 - val_los
s: 0.0639 - val_r2_score: -0.0877
Epoch 32/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2006 - r2_score: -0.0626 - val_los
s: 0.0616 - val_r2_score: -0.0741
Epoch 33/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1923 - r2_score: -0.0531 - val_los
s: 0.0841 - val_r2_score: -0.4837
Epoch 34/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1868 - r2_score: -0.0146 - val_los
s: 0.0739 - val_r2_score: -0.3042
Epoch 35/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1994 - r2_score: -0.0531 - val_los
s: 0.1803 - val_r2_score: -2.1833
Epoch 36/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1976 - r2_score: -0.0614 - val_los
s: 0.1719 - val_r2_score: -2.0098
Epoch 37/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2221 - r2_score: -0.1103 - val_los
s: 0.1236 - val_r2_score: -1.1712
Epoch 38/100
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2018 - r2_score: -0.0271 - val_los
s: 0.0637 - val_r2_score: -0.1199
Epoch 39/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2060 - r2_score: -0.0770 - val_los
s: 0.0937 - val_r2_score: -0.6471
Epoch 40/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1877 - r2_score: -0.0206 - val_los
s: 0.0667 - val_r2_score: -0.1771
Epoch 41/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1983 - r2_score: -0.0264 - val_los
s: 0.0730 - val_r2_score: -0.2893
Epoch 42/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1983 - r2_score: -0.0130 - val_los
s: 0.0715 - val_r2_score: -0.2627
Epoch 43/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2131 - r2_score: -0.0512 - val_los
s: 0.1065 - val_r2_score: -0.8802
Epoch 44/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1895 - r2_score: -0.0172 - val_los
s: 0.1287 - val_r2_score: -1.2737
Epoch 45/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2160 - r2_score: -0.0436 - val_los
s: 0.0711 - val_r2_score: -0.2485
Epoch 46/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2053 - r2_score: -0.1166 - val_los
s: 0.1008 - val_r2_score: -0.7476
Epoch 47/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2103 - r2_score: -0.0905 - val_los
s: 0.1495 - val_r2_score: -1.5980
Epoch 48/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2097 - r2_score: -0.0917 - val_los
s: 0.0998 - val_r2_score: -0.7465
Epoch 49/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2006 - r2_score: -0.0337 - val_los
s: 0.0585 - val_r2_score: -0.0218
Epoch 50/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2107 - r2_score: -0.0995 - val_los
s: 0.0624 - val_r2_score: -0.0720
Epoch 51/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1992 - r2_score: -0.0508 - val_los
s: 0.1014 - val_r2_score: -0.7835
Epoch 52/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1961 - r2_score: -0.0511 - val_los
s: 0.1072 - val_r2_score: -0.8730
Epoch 53/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2016 - r2_score: -0.0351 - val_los
s: 0.1793 - val_r2_score: -2.1620
Epoch 54/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2248 - r2_score: -0.2706 - val_los
s: 0.1401 - val_r2_score: -1.3555
Epoch 55/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2187 - r2_score: -0.0681 - val_los
s: 0.1367 - val_r2_score: -1.3667
Epoch 56/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2035 - r2_score: -0.0064 - val_los
s: 0.1295 - val_r2_score: -1.2817
```

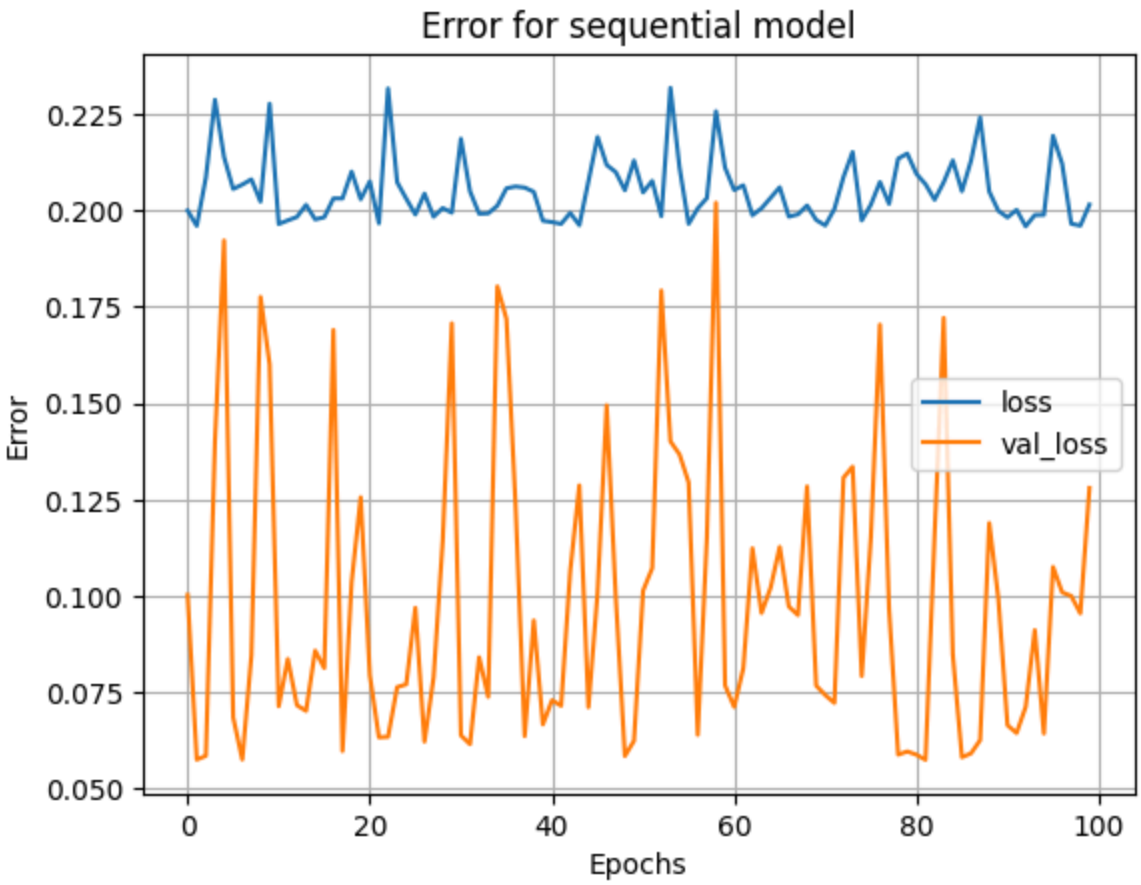
```
Epoch 57/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1952 - r2_score: -0.0376 - val_los
s: 0.0640 - val_r2_score: -0.1293
Epoch 58/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1948 - r2_score: -0.0194 - val_los
s: 0.1151 - val_r2_score: -1.0272
Epoch 59/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2241 - r2_score: -0.1385 - val_los
s: 0.2020 - val_r2_score: -2.4605
Epoch 60/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2188 - r2_score: -0.1132 - val_los
s: 0.0768 - val_r2_score: -0.3134
Epoch 61/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1955 - r2_score: -0.0395 - val_los
s: 0.0712 - val_r2_score: -0.2525
Epoch 62/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2067 - r2_score: -0.0768 - val_los
s: 0.0813 - val_r2_score: -0.4243
Epoch 63/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2063 - r2_score: -0.0323 - val_los
s: 0.1124 - val_r2_score: -0.9805
Epoch 64/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1986 - r2_score: -0.0367 - val_los
s: 0.0956 - val_r2_score: -0.6863
Epoch 65/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1947 - r2_score: -0.0205 - val_los
s: 0.1022 - val_r2_score: -0.8012
Epoch 66/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2049 - r2_score: -0.0360 - val_los
s: 0.1128 - val_r2_score: -0.9696
Epoch 67/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1997 - r2_score: -0.0342 - val_los
s: 0.0973 - val_r2_score: -0.7133
Epoch 68/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1995 - r2_score: -0.0140 - val_los
s: 0.0951 - val_r2_score: -0.6781
Epoch 69/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2076 - r2_score: -0.0308 - val_los
s: 0.1285 - val_r2_score: -1.2647
Epoch 70/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2150 - r2_score: -0.0279 - val_los
s: 0.0768 - val_r2_score: -0.3544
Epoch 71/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1896 - r2_score: -0.0086 - val_los
s: 0.0743 - val_r2_score: -0.3128
Epoch 72/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2078 - r2_score: -0.0319 - val_los
s: 0.0723 - val_r2_score: -0.2743
Epoch 73/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1980 - r2_score: -0.0624 - val_los
s: 0.1307 - val_r2_score: -1.2829
Epoch 74/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2206 - r2_score: -0.1286 - val_los
s: 0.1336 - val_r2_score: -1.3173
Epoch 75/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2066 - r2_score: -0.0218 - val_los
```

```
s: 0.0793 - val_r2_score: -0.3964
Epoch 76/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1726 - r2_score: -0.0224 - val_los
s: 0.1151 - val_r2_score: -1.0269
Epoch 77/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2224 - r2_score: -0.1004 - val_los
s: 0.1704 - val_r2_score: -1.9930
Epoch 78/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2019 - r2_score: -0.0874 - val_los
s: 0.0968 - val_r2_score: -0.7045
Epoch 79/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2105 - r2_score: -0.1117 - val_los
s: 0.0588 - val_r2_score: -0.0053
Epoch 80/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2099 - r2_score: -0.1187 - val_los
s: 0.0597 - val_r2_score: -8.0538e-04
Epoch 81/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2201 - r2_score: -0.0763 - val_los
s: 0.0588 - val_r2_score: -0.0065
Epoch 82/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2213 - r2_score: -0.0858 - val_los
s: 0.0575 - val_r2_score: -0.0020
Epoch 83/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1951 - r2_score: -0.0660 - val_los
s: 0.1158 - val_r2_score: -1.0313
Epoch 84/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2015 - r2_score: -0.0657 - val_los
s: 0.1722 - val_r2_score: -2.0111
Epoch 85/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2073 - r2_score: -0.1094 - val_los
s: 0.0848 - val_r2_score: -0.4514
Epoch 86/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2163 - r2_score: -0.0258 - val_los
s: 0.0582 - val_r2_score: -4.2558e-05
Epoch 87/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2106 - r2_score: -0.1599 - val_los
s: 0.0592 - val_r2_score: -0.0068
Epoch 88/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2209 - r2_score: -0.1469 - val_los
s: 0.0626 - val_r2_score: -0.0060
Epoch 89/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2201 - r2_score: -0.0460 - val_los
s: 0.1190 - val_r2_score: -1.0543
Epoch 90/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2055 - r2_score: -0.0217 - val_los
s: 0.0994 - val_r2_score: -0.7404
Epoch 91/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2184 - r2_score: -0.0183 - val_los
s: 0.0665 - val_r2_score: -0.1696
Epoch 92/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1979 - r2_score: -0.0275 - val_los
s: 0.0645 - val_r2_score: -0.1306
Epoch 93/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1951 - r2_score: -0.0184 - val_los
s: 0.0712 - val_r2_score: -0.2554
Epoch 94/100
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1979 - r2_score: -0.0165 - val_los
s: 0.0912 - val_r2_score: -0.6047
Epoch 95/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1852 - r2_score: -0.0099 - val_los
s: 0.0643 - val_r2_score: -0.1252
Epoch 96/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2104 - r2_score: -0.1395 - val_los
s: 0.1076 - val_r2_score: -0.7897
Epoch 97/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2099 - r2_score: -0.0946 - val_los
s: 0.1010 - val_r2_score: -0.7426
Epoch 98/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1947 - r2_score: -0.0103 - val_los
s: 0.1000 - val_r2_score: -0.7641
Epoch 99/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1948 - r2_score: -0.0174 - val_los
s: 0.0955 - val_r2_score: -0.6865
Epoch 100/100
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2205 - r2_score: -0.0327 - val_los
s: 0.1280 - val_r2_score: -1.2525
```

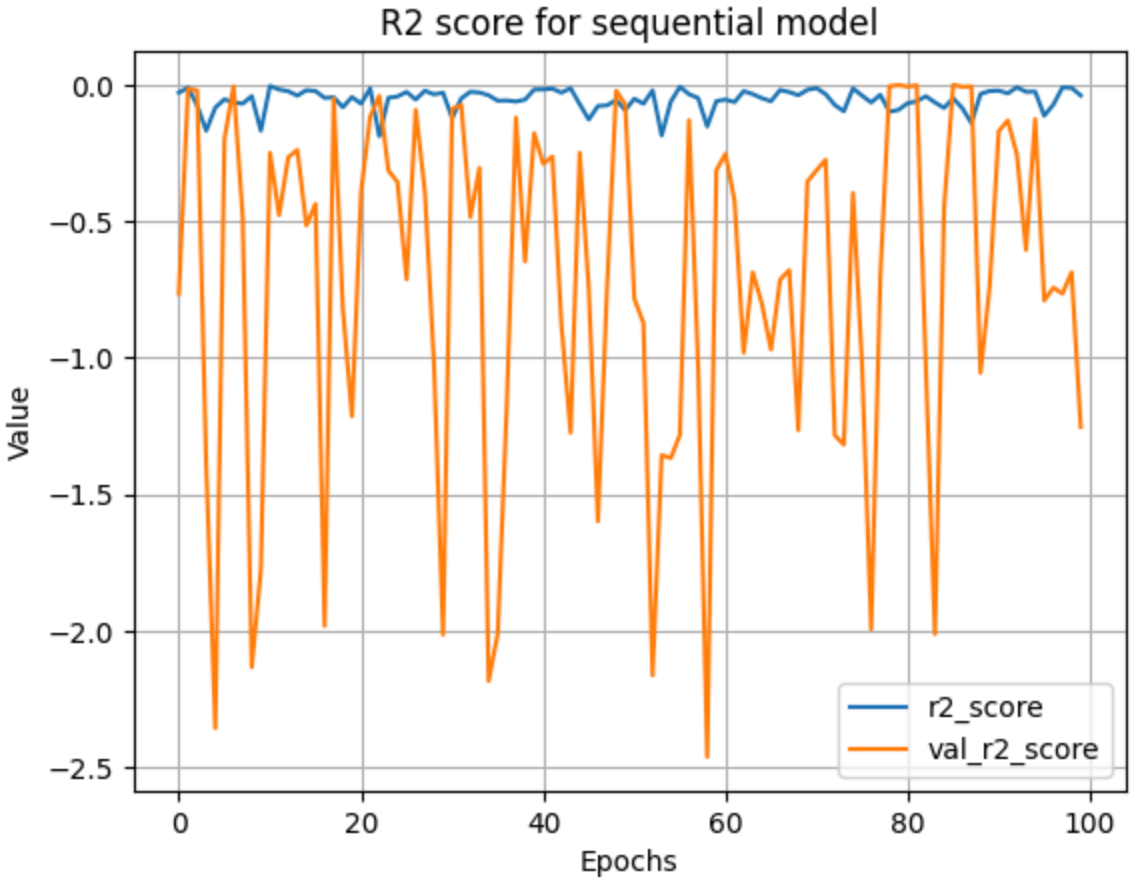
In [695…

```
plot_loss(uhist)
```



In [696…

```
plot_metrics(uhist)
```



In [704…
```
# xy_test = np.array(test[['temp', 'doy']])
# xy_test[:10]
X = np.array(test[['temp', 'doy']])
xs = X[:, 0]
ys = X[:, 1]
zs = np.array(df['temp_lower'])
```

In [705…
```
n_plot = 51

x_plot = np.linspace(np.min(xs), np.max(xs), n_plot)
y_plot = np.linspace(np.min(ys), np.max(ys), n_plot)
x_mesh, y_mesh = np.meshgrid(x_plot, y_plot)
x_mesh.shape, y_mesh.shape
```

Out[705…
```
((51, 51), (51, 51))
```

In [ ]:
```
# x_mesh, y_mesh = np.meshgrid(x_plot, y_plot)
# x_mesh.shape, y_mesh.shape
```

In [706…
```
x_plot2 = np.reshape(x_mesh, [n_plot**2,1])
y_plot2 = np.reshape(y_mesh, [n_plot**2,1])
xy_2 = np.hstack([x_plot2, y_plot2])
xy_2.shape
```

Out[706…
```
(2601, 2)
```

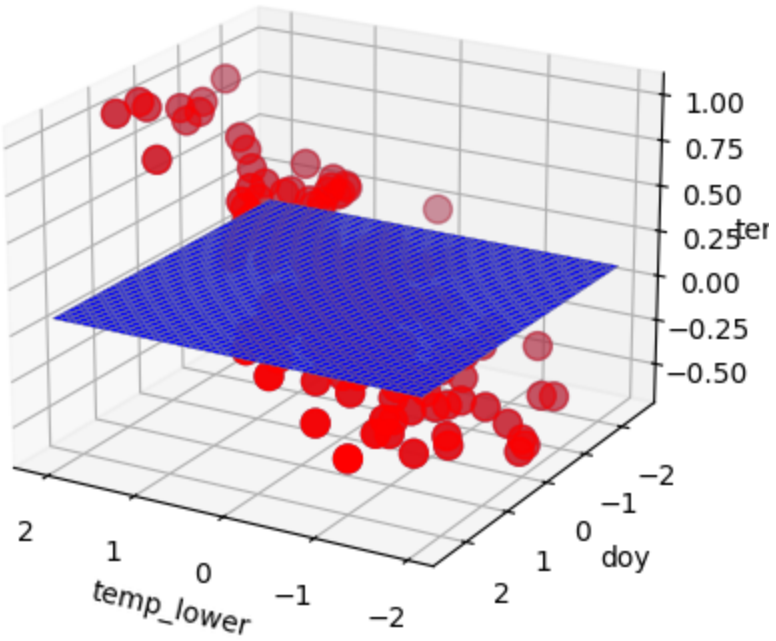
In [707…
```
z = uni_model.predict(xy_2)
```

```
82/82 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

In [708…
```
z_mesh = z.reshape((n_plot, n_plot))
z_mesh.shape
```

Out[708…
```
(51, 51)
```

In [709…
```
fig = plt.figure()
ax = plt.axes(projection='3d')

xs = xy_test[:, 0]
ys = xy_test[:, 1]
zs = test['temp_lower']

ax.scatter(xs, ys, zs, s = 100)

ax.set_xlabel('temp_lower')
ax.set_ylabel('doy')
ax.set_zlabel('temp')

ax.plot_surface(x_mesh, y_mesh, z_mesh, linewidth = 0.25, cstride=1, rstride=1,
                cmap=cm.winter, antialiased=True, edgecolors='gray')
ax.scatter( xs, ys, zs, s=100, c='r' )

ax.view_init(azim = 120, elev = 20)
```



Как мы видим, ситуация не изменилась, ведь признак никак не коррелирует с остальными, что было показано в конце 9ой задачи