РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

ОТЧЕТ ПО КОНТРОЛЬНОЙ РАБОТЕ № 1

Дисциплина: Методы машинного обучения

Студент: Петров Артем Евгеньевич

Группа: НКНбд-01-21

Москва 2024

Вариант №1

Текст задания...

Решение...

В соответствии с индивидуальным заданием, указанным в записной книжке команды Teams, выполните следующее:

Загрузите заданный в индивидуальном задании набор данных из Tensorflow Datasets и оставьте в наборе данных признаки, принимающие непрерывные числовые значения, включая указанные в индивидуальном задании независимую и зависимую переменные. Вычислите матрицу корреляции признаков и выведите названия (номера) пар признаков с наиболее низкой и наиболее высокой корреляцией. Выполните визуализацию независимой и зависимой переменных в соответствии с индивидуальным заданием, подписывая оси и рисунок. Постройте диаграмму рассеяния для независимого и зависимого признаков, подписывая оси и рисунок, определите наличие одиноко расположенные точек и, при наличии, удалите их. Постройте парную линейную регрессию для независимого и зависимого признаков при помощи точного подхода и при помощи нейронной сети с одним нейроном. Определите лучший из двух подходов по показателю качества R^2 (коэффициенту детерминации). Постройте диаграмму рассеяния для независимого и зависимого признаков и изобразите линии двух построенных парных регрессий, подписывая оси и рисунок и создавая легенду для линий регрессии.

Разбейте набор признаков на обучающую и контрольную выборки. Создайте и адаптируйте нормализующий слой Tensorflow для всех признаков набора данных (за исключением зависимого признака). Нормализуйте зависимый признак. Используя созданный нормализующий слой и нормализованный зависимый признак, постройте регресоры на базе следующих моделей множественной регрессии: линейной регрессии гребневой регрессии (L2) лассо регрессии (L1)

Выберите коэффициенты регуляризации l1 и l2 так, чтобы нейронные сети для всех трех моделей обучались (значение ошибки уменьшалось в процессе обучения). Определите на контрольной выборке (с нормализованным зависимым признаком) модель множественной регрессии с наиболее высоким качеством по показателю, указанному в индивидуальном задании, среди построенных моделей. Для лучшего регрессора визуализируйте кривые обучения (в зависимости от эпохи обучения). Определите медианные значения признаков (кроме независимого и зависимого признаков) и для построенных медианных значений визуализируйте на плоскости с независимым признаком в качестве оси абсцисс и зависимым признаком в качестве оси ординат точки тестовой выборки и линии (графики) различных моделей множественной регрессии разными цветами. Подпишите оси и создайте легенду и заголовок для рисунка. Результат контрольной работы оформить в виде отчета в формате файла Jupiter Notebook (шаблон отчета находится в учебных материалах команды в формате .ipynb). Включите в отчет номер варианта, текст индивидуального задания, пункты 1-10 задания, указанные выше, и программный код для решения этих пунктов. Сопроводите программный код необходимыми комментариями. Дополнительно (кроме файла расширением .ipynb) представить распечатку файла с отчетом в формате PDF. Не архивировать файлы.

Контрольная работа 1 – Вариант 13

1. Набор данных: cherry_blossoms

2. Независимая переменная: temp_lower

3. Зависимая переменная: temp_upper

4. Визуализация для независимой переменной – эмпирическая плотность распределения

5. Визуализация для зависимой переменной – столбчатая диаграмма

6. Показатель качества регрессии – R^2 (коэффициент детерминации)

In [1]:

```
import sys

sys.setrecursionlimit(1000)
```

1. Загрузите заданный в индивидуальном задании набор данных из Tensorflow Datasets и оставьте в наборе данных признаки, принимающие непрерывные числовые значения, включая указанные в индивидуальном задании независимую и зависимую переменные. Вычислите матрицу корреляции признаков и выведите названия (номера) пар признаков с наиболее низкой и наиболее высокой корреляцией.

In [114…

```
# import tensorflow_datasets as tfds
# import tensorflow as tf
import pandas as pd
```

In [115…

```
# builder = tfds.builder('cherry_blossoms')
# builder.download_and_prepare()
df = pd.read_csv('https://raw.githubusercontent.com/rmcelreath/rethinking/master/da
```

In [116…

```
display(df)
```

	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
...	...	...	...	...	...
1210	2011	99.0	NaN	NaN	NaN
1211	2012	101.0	NaN	NaN	NaN
1212	2013	93.0	NaN	NaN	NaN
1213	2014	94.0	NaN	NaN	NaN
1214	2015	93.0	NaN	NaN	NaN

1215 rows × 5 columns

In [117...

```
df.info()
```

```
<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
```

In [118...

```
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 [119...

```
df.isna().sum()
```

Out[119…

```
year            0
doy           388
temp           91
temp_upper     91
temp_lower     91
dtype: int64
```

In [124…

```
df = df.dropna()
```

In [125…

```
display(df_fixed)
```

	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 [126…

```
import numpy as np
```

In [132…

```
# df_max_corr = df.corr()
# display(df_max_corr)

print("Data Frame")
print(df)
print()

print("Correlation Matrix")
print(df.corr())
print()

def get_redundant_pairs(df):
    '''Get diagonal and lower triangular pairs of correlation matrix'''
    pairs_to_drop = set()
    cols = df.columns
    for i in range(0, df.shape[1]):
        for j in range(0, i+1):
                pairs_to_drop.add((cols[i], cols[j]))
    return pairs_to_drop

def get_top_abs_correlations(df, n=5, ascending=False):
    au_corr = df.corr().unstack()
    labels_to_drop = get_redundant_pairs(df)
    au_corr = au_corr.drop(labels=labels_to_drop).sort_values(ascending=ascending)
    return au_corr[0:n]

print("Top Absolute Correlations")
print(get_top_abs_correlations(df, 3, ascending=False))
print("Top lowest correlations")
print(get_top_abs_correlations(df, 3, ascending=True))
```

```
Data Frame
      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 x 5 columns]

Correlation Matrix
                year       doy      temp  temp_upper  temp_lower
year        1.000000  0.013970  0.028033   -0.315807    0.386703
doy         0.013970  1.000000 -0.326976   -0.294773   -0.271967
temp        0.028033 -0.326976  1.000000    0.876747    0.858841
temp_upper -0.315807 -0.294773  0.876747    1.000000    0.506662
temp_lower  0.386703 -0.271967  0.858841    0.506662    1.000000

Top Absolute Correlations
temp        temp_upper    0.876747
            temp_lower    0.858841
temp_upper  temp_lower    0.506662
dtype: float64
Top lowest correlations
doy   temp          -0.326976
year  temp_upper    -0.315807
doy   temp_upper    -0.294773
dtype: float64
```

2. Выполните визуализацию независимой и зависимой переменных в соответствии с индивидуальным заданием, подписывая оси и рисунок.

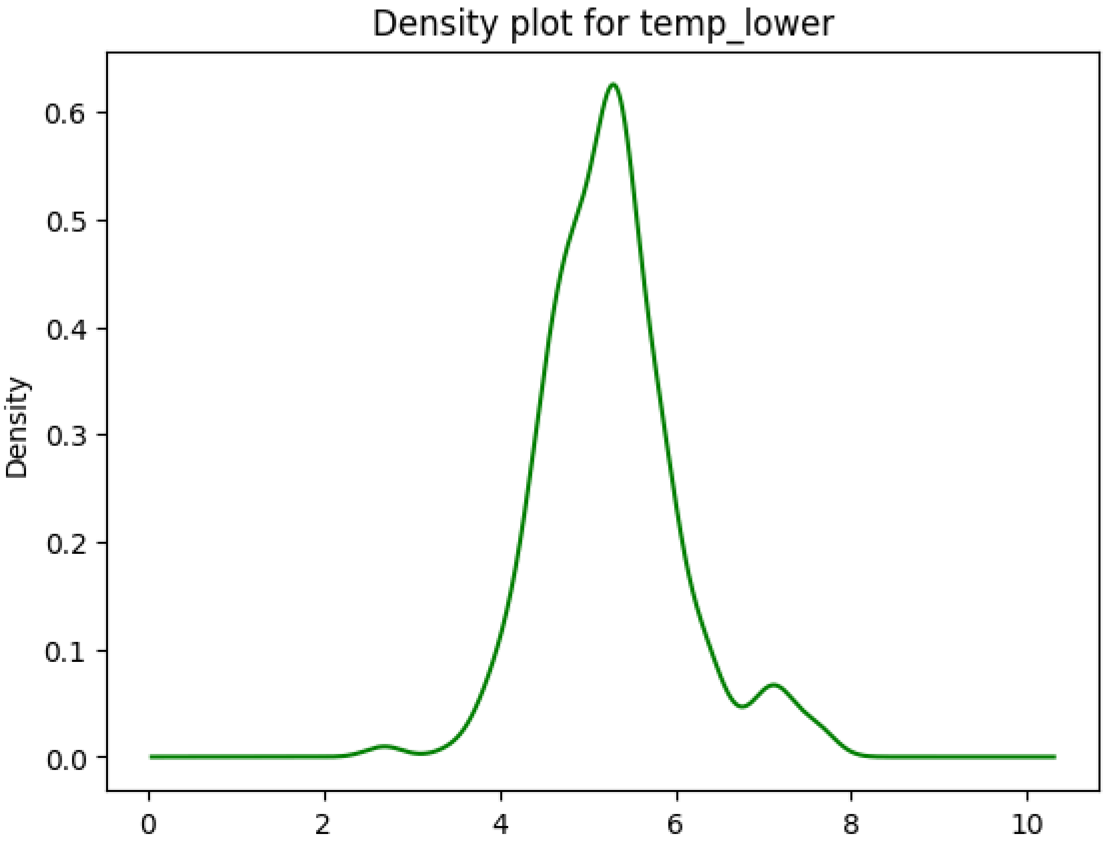
Визуализация для независимой переменной – эмпирическая плотность распределения(Независимая переменная: temp_lower)

Визуализация для зависимой переменной – столбчатая диаграмма (Зависимая переменная: temp_upper)

```
In [138…  import matplotlib.pyplot as plt
```

```
In [139…  df['temp_lower'].plot.density(color = 'green', title = 'Density plot for temp_lower
```

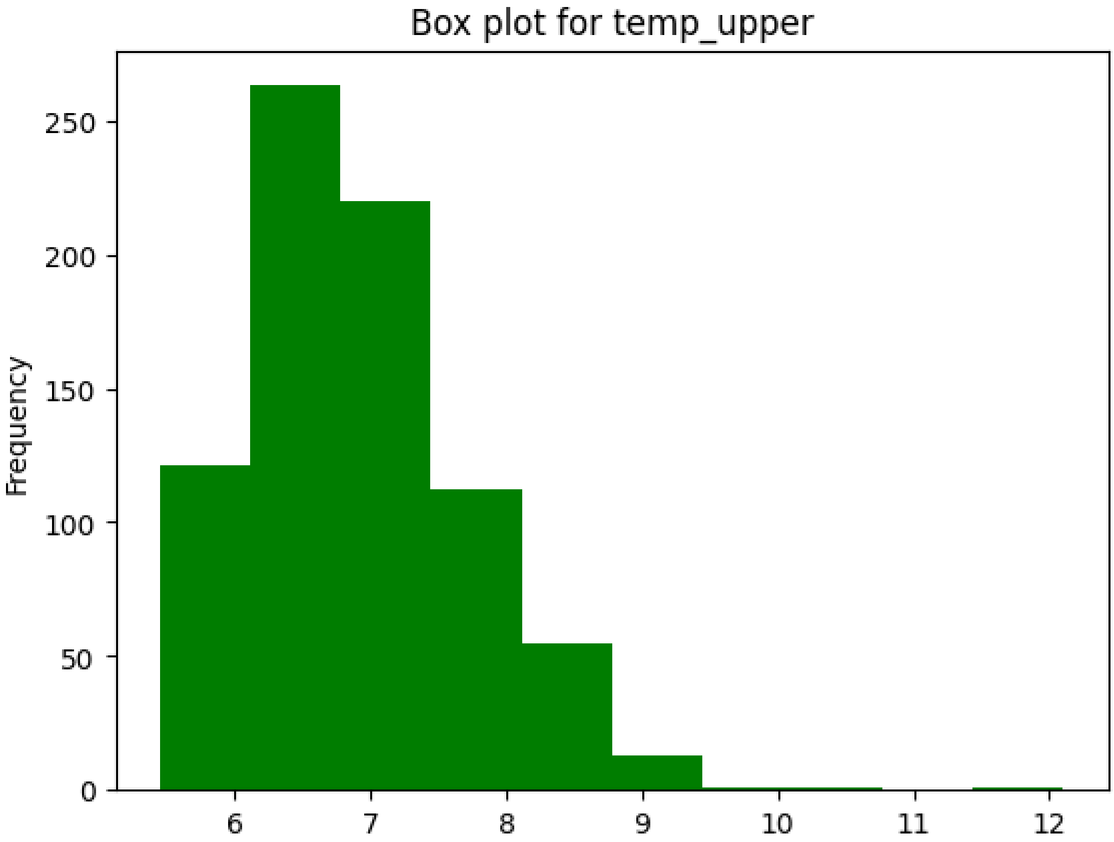
```
Out[139…  <Axes: title={'center': 'Density plot for temp_lower'}, ylabel='Density'>
```



```
In [143…  print(df['temp_upper'].value_counts())
          df['temp_upper'].plot.hist(bins = 10, color = 'green', title = "Box plot for temp_u
temp_upper
6.74    10
6.41     9
6.45     9
7.09     9
6.64     9
        ..
7.70     1
7.47     1
7.23     1
5.84     1
8.86     1
Name: count, Length: 275, dtype: int64
```

Out[143…

```
<Axes: title={'center': 'Box plot for temp_upper'}, ylabel='Frequency'>
```



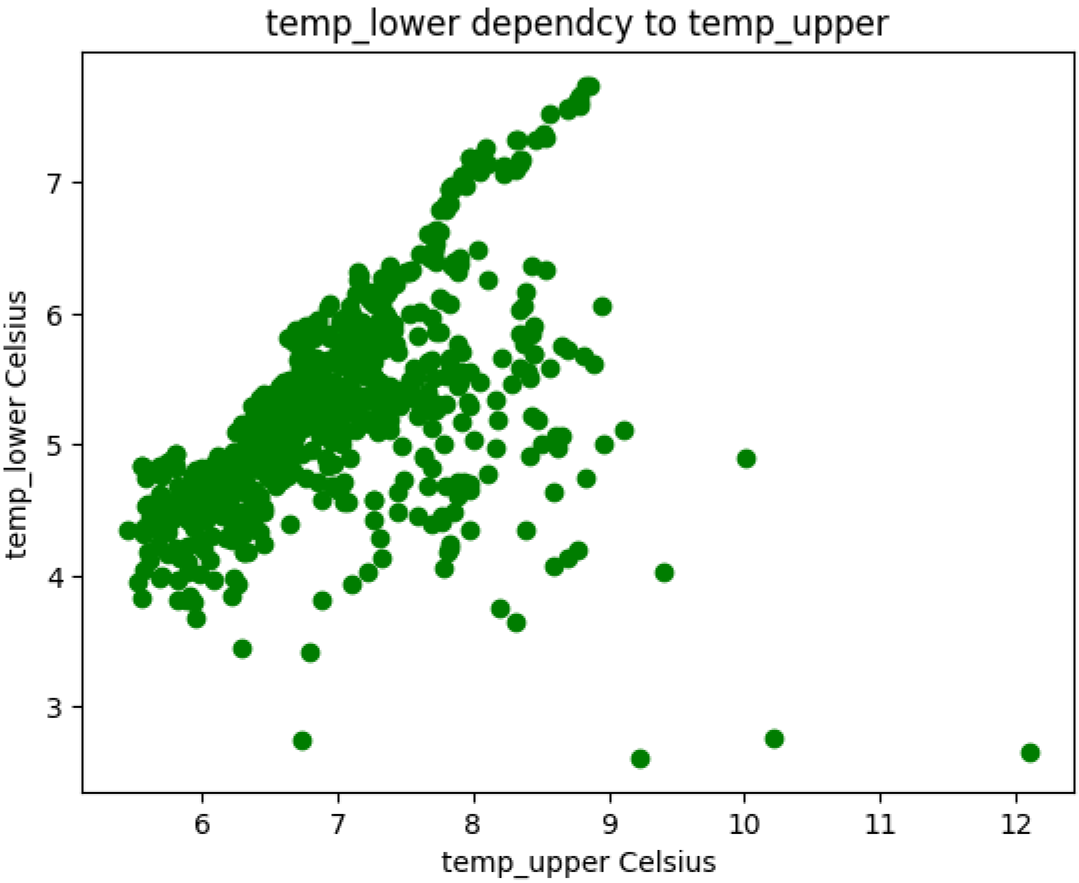
3. Постройте диаграмму рассеяния для независимого и зависимого признаков, подписывая оси и рисунок, определите наличие одиноко расположенные точек и, при наличии, удалите их.

In [164…

```
plt.scatter(df['temp_upper'], df['temp_lower'], color = 'g')
plt.xlabel("temp_upper Celsius")
plt.ylabel("temp_lower Celsius")
plt.title("temp_lower dependcy to temp_upper")
```

Out[164…

```
Text(0.5, 1.0, 'temp_lower dependcy to temp_upper')
```



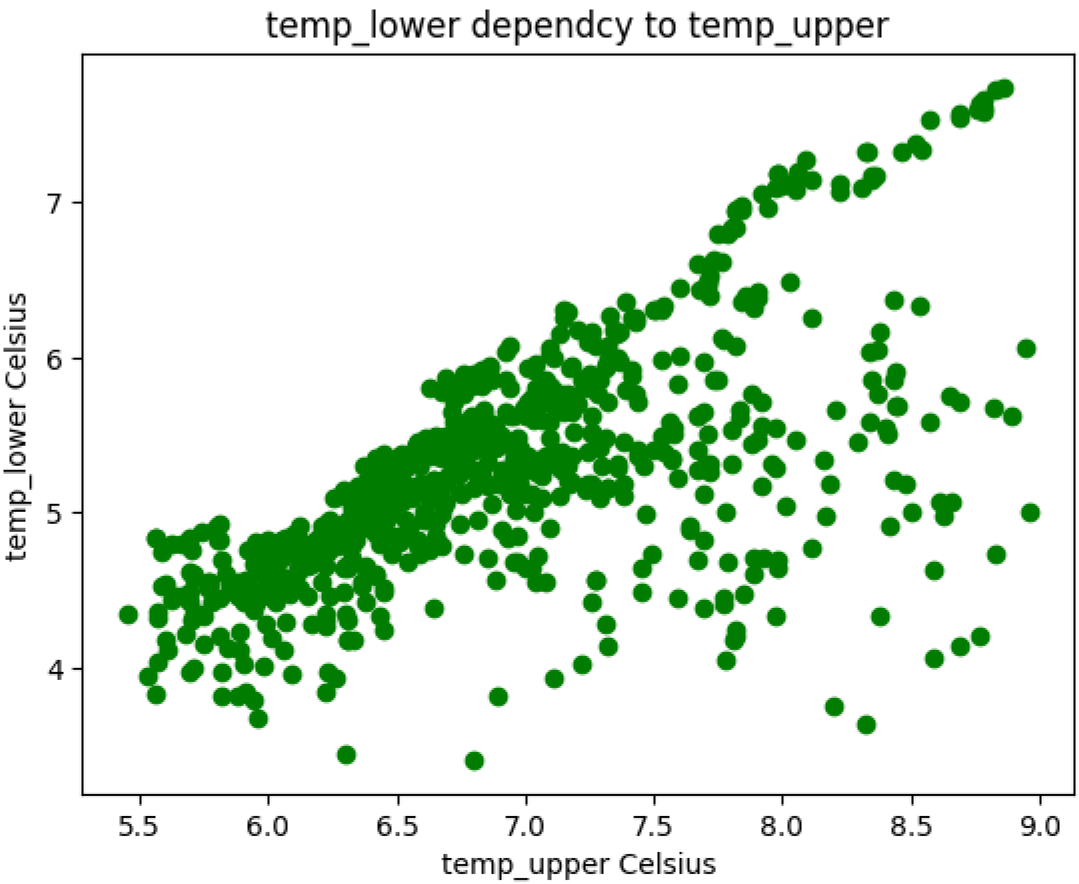
In [165…

```
df_new = df[(df['temp_upper'] < 9) & (df['temp_lower'] < 10) & (df['temp_lower'] >

plt.scatter(df_new['temp_upper'], df_new['temp_lower'], color = 'g')
plt.xlabel("temp_upper Celsius")
plt.ylabel("temp_lower Celsius")
plt.title("temp_lower dependcy to temp_upper")
```

Out[165…

```
Text(0.5, 1.0, 'temp_lower dependcy to temp_upper')
```



In [166…

```
class SimpleLinReg:

    def __init__(self):
        self.a_ = None
        self.b_ = None

    def fit(self, x_train, y_train):
        assert x_train.ndim == 1, \
            "В данных должен быть один признак"
        assert len(x_train) == len(y_train), \
            "Данные должны иметь одинаковый размер"

        x_mean = np.mean(x_train)
        y_mean = np.mean(y_train)

        self.a_ = (x_train - x_mean).dot(y_train - y_mean) / \
                  (x_train - x_mean).dot(x_train - x_mean)
        self.b_ = y_mean - self.a_ * x_mean

        return self

    def predict(self, x_predict):
        assert x_predict.ndim == 1, \
            "В данных должен быть один признак"
        assert self.a_ is not None and self.b_ is not None, \
            "Модель вначале должна быть обучена"

        return np.array([self._predict(x) for x in x_predict])
    def _predict(self, x_single):
        return self.a_ * x_single + self.b_

    def __repr__(self):
        return "SimpleLinearReg()"
```

In [167…

```
reg = SimpleLinReg()
reg
```

Out[167…

```
SimpleLinearReg()
```

In [180…

```
split_df = np.array_split(df_new, 2)
```

```
C:\Python312\Lib\site-packages\numpy\core\fromnumeric.py:59: FutureWarning: 'DataFra
me.swapaxes' is deprecated and will be removed in a future version. Please use 'Data
Frame.transpose' instead.
  return bound(*args, **kwds)
```

In [181…

```
train = split_df[0]
test = split_df[1]
print(train, type(train), test, type(test))
```

```
     year    doy  temp  temp_upper  temp_lower
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
93    894  106.0  6.98        8.40        5.55
..    ...    ...   ...         ...         ...
761  1562   99.0  6.29        6.92        5.65
762  1563   99.0  6.13        6.78        5.49
763  1564  109.0  6.18        6.96        5.39
764  1565  114.0  6.16        6.89        5.43
765  1566   97.0  6.28        6.94        5.63

[390 rows x 5 columns] <class 'pandas.core.frame.DataFrame'>       year    doy  temp
temp_upper  temp_lower
766   1567  108.0  6.09        6.76        5.42
767   1568  106.0  6.09        6.76        5.43
769   1570  112.0  6.16        6.80        5.51
770   1571  108.0  6.35        7.04        5.67
771   1572  107.0  6.38        7.06        5.71
...    ...    ...   ...         ...         ...
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

[390 rows x 5 columns] <class 'pandas.core.frame.DataFrame'>
```

In [182…

```
reg.fit(train['temp_upper'].values, train['temp_lower'].values)
```

Out[182…

```
SimpleLinearReg()
```

In [183…

```
reg.a_, reg.b_
```

Out[183…

```
(0.4197131735986538, 2.152058752186659)
```

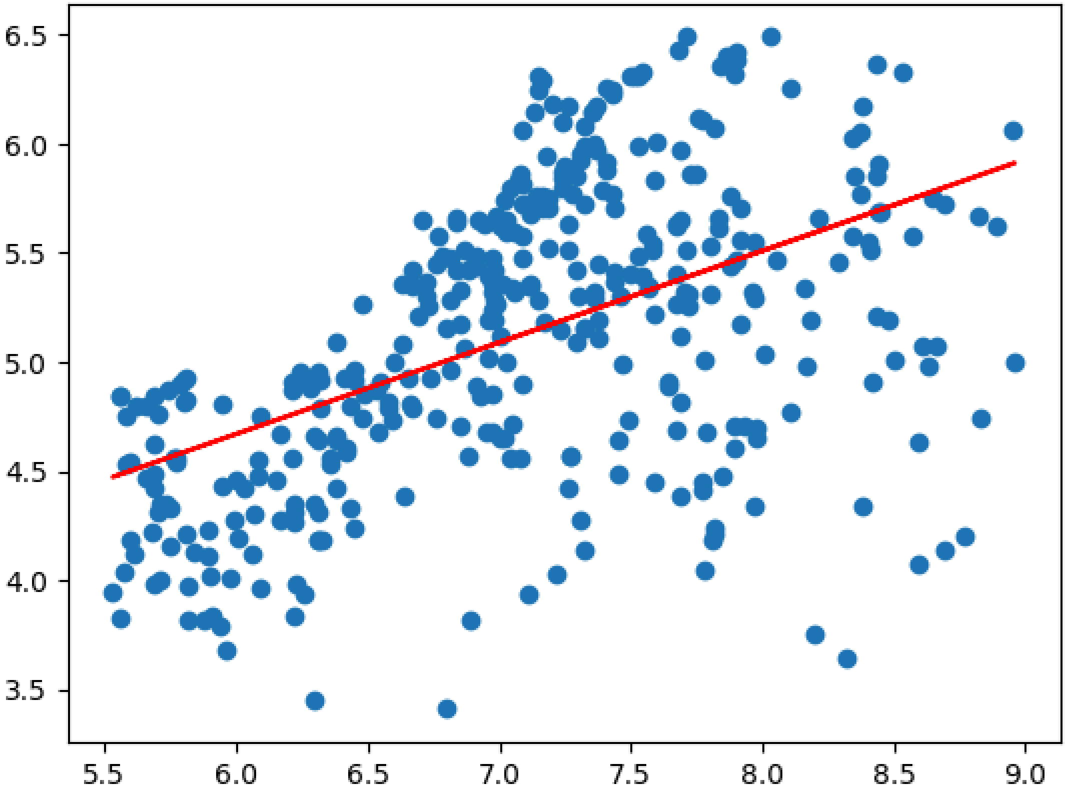
In [184…

```
y_hat = reg.predict(train['temp_upper'].values)

plt.scatter(train['temp_upper'], train['temp_lower'])
plt.plot(train['temp_upper'], y_hat, color = 'r')
```

Out[184…

```
[<matplotlib.lines.Line2D at 0x1696f3e8d40>]
```



In [192…

```
y_test = train['temp_lower'].values
mse_test = np.sum((y_hat - y_test)**2) / len(y_test)
print("mean square error", mse_test)

metric_reg = 1 - mse_test/np.var(y_test)
print("R^2=", metric_reg)
```

```
mean square error 0.34041742579150724
R^2= 0.25615040590148686
```

Через TensorFlow

In [193…

```
import tensorflow as tf
print(tf.__version__)
```

```
2.16.1
```

In [214…

```
model = tf.keras.Sequential([ tf.keras.layers.Dense(1, input_shape=(1,)) ])
```

```
C:\Python312\Lib\site-packages\keras\src\layers\core\dense.py:86: UserWarning: Do no
t pass an `input_shape`/`input_dim` argument to a layer. When using Sequential model
s, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [215…

```python
model.summary()
```

```
Model: "sequential_3"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━
┃ Layer (type)                    ┃ Output Shape            ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━
│ dense_3 (Dense)                 │ (None, 1)               │
└─────────────────────────────────┴─────────────────────────┴──────

 Total params: 2 (8.00 B)

 Trainable params: 2 (8.00 B)

 Non-trainable params: 0 (0.00 B)
```

In [216…

```python
model.compile(
    loss = tf.keras.losses.mean_absolute_error,
    optimizer = tf.keras.optimizers.Adam(learning_rate = 0.25),
    metrics=['r2_score']
)
```

In [217…

```python
model.fit(train['temp_upper'].values, train['temp_lower'].values, epochs=150)
```

```
Epoch 1/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 2.4310 - r2_score: -19.9360
Epoch 2/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.8094 - r2_score: -1.2101
Epoch 3/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.6655 - r2_score: -0.6278
Epoch 4/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5528 - r2_score: -0.1052
Epoch 5/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5606 - r2_score: -0.0660
Epoch 6/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5138 - r2_score: -0.0104
Epoch 7/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5322 - r2_score: -0.2665
Epoch 8/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5346 - r2_score: 0.0220
Epoch 9/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5210 - r2_score: -0.0653
Epoch 10/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4935 - r2_score: 0.0848
Epoch 11/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4871 - r2_score: 0.1146
Epoch 12/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5317 - r2_score: 0.0747
Epoch 13/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5951 - r2_score: -0.2773
Epoch 14/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5338 - r2_score: -0.0601
Epoch 15/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4920 - r2_score: 0.1534
Epoch 16/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5086 - r2_score: 0.1194
Epoch 17/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4958 - r2_score: 0.0346
Epoch 18/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4911 - r2_score: 0.1732
Epoch 19/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4979 - r2_score: 0.0455
Epoch 20/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4761 - r2_score: 0.0710
Epoch 21/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5084 - r2_score: 0.0570
Epoch 22/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4847 - r2_score: 0.1479
Epoch 23/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4859 - r2_score: 0.1565
Epoch 24/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5249 - r2_score: 0.0897
Epoch 25/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5375 - r2_score: 0.0361
Epoch 26/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5554 - r2_score: -0.0324
Epoch 27/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4872 - r2_score: 0.2303
Epoch 28/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5319 - r2_score: -0.0202
```

```
Epoch 29/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4954 - r2_score: 0.1597
Epoch 30/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4893 - r2_score: 0.0842
Epoch 31/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5203 - r2_score: 0.0918
Epoch 32/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4867 - r2_score: 0.2158
Epoch 33/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5843 - r2_score: -0.0995
Epoch 34/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5100 - r2_score: 0.0512
Epoch 35/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5235 - r2_score: 0.0581
Epoch 36/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5150 - r2_score: -0.0022
Epoch 37/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5300 - r2_score: 0.0407
Epoch 38/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4604 - r2_score: 0.2302
Epoch 39/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5241 - r2_score: -0.1001
Epoch 40/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.5670 - r2_score: -0.2622
Epoch 41/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4538 - r2_score: 0.1983
Epoch 42/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5283 - r2_score: 0.0270
Epoch 43/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4694 - r2_score: 0.1815
Epoch 44/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4950 - r2_score: 0.2274
Epoch 45/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 906us/step - loss: 0.4827 - r2_score: 0.2430
Epoch 46/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5124 - r2_score: 0.0064
Epoch 47/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4664 - r2_score: 0.1645
Epoch 48/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5001 - r2_score: 0.1108
Epoch 49/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4677 - r2_score: 0.1910
Epoch 50/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4768 - r2_score: 0.1902
Epoch 51/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5390 - r2_score: -0.0203
Epoch 52/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4503 - r2_score: 0.2218
Epoch 53/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5270 - r2_score: 0.0291
Epoch 54/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4860 - r2_score: 0.1050
Epoch 55/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.6047 - r2_score: -0.3992
Epoch 56/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5031 - r2_score: 0.0856
```

```
Epoch 57/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5323 - r2_score: 0.0243
Epoch 58/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4915 - r2_score: 0.1319
Epoch 59/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4615 - r2_score: 0.2204
Epoch 60/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5953 - r2_score: -0.3576
Epoch 61/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4879 - r2_score: 0.1540
Epoch 62/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5246 - r2_score: 0.1120
Epoch 63/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4772 - r2_score: 0.1213
Epoch 64/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5062 - r2_score: 0.0689
Epoch 65/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5151 - r2_score: 0.0787
Epoch 66/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5174 - r2_score: 0.1344
Epoch 67/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4985 - r2_score: 0.1011
Epoch 68/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4646 - r2_score: 0.2354
Epoch 69/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5586 - r2_score: -0.0350
Epoch 70/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5053 - r2_score: 0.1118
Epoch 71/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5481 - r2_score: 0.0279
Epoch 72/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4773 - r2_score: 0.1501
Epoch 73/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5026 - r2_score: 0.0346
Epoch 74/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5391 - r2_score: -0.1179
Epoch 75/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5725 - r2_score: -0.1376
Epoch 76/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5854 - r2_score: -0.2123
Epoch 77/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4926 - r2_score: 0.2256
Epoch 78/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4791 - r2_score: 0.1286
Epoch 79/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 844us/step - loss: 0.4720 - r2_score: 0.1434
Epoch 80/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4764 - r2_score: 0.1251
Epoch 81/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5281 - r2_score: -0.0431
Epoch 82/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.6319 - r2_score: -0.2648
Epoch 83/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.6065 - r2_score: -0.3075
Epoch 84/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5394 - r2_score: 0.0630
```

```
Epoch 85/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4584 - r2_score: 0.2229
Epoch 86/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4746 - r2_score: 0.1822
Epoch 87/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4609 - r2_score: 0.2295
Epoch 88/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4558 - r2_score: 0.2322
Epoch 89/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5846 - r2_score: -0.1834
Epoch 90/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4995 - r2_score: 0.1496
Epoch 91/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4812 - r2_score: 0.1586
Epoch 92/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4901 - r2_score: 0.1142
Epoch 93/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5441 - r2_score: 0.0735
Epoch 94/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4762 - r2_score: 0.2253
Epoch 95/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 796us/step - loss: 0.4855 - r2_score: 0.1145
Epoch 96/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4931 - r2_score: 0.1508
Epoch 97/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 654us/step - loss: 0.7370 - r2_score: -0.8514
Epoch 98/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5358 - r2_score: -0.0478
Epoch 99/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 919us/step - loss: 0.5006 - r2_score: 0.0479
Epoch 100/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4629 - r2_score: 0.2115
Epoch 101/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5289 - r2_score: -0.0833
Epoch 102/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4906 - r2_score: 0.1244
Epoch 103/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4694 - r2_score: 0.2361
Epoch 104/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5097 - r2_score: 0.0752
Epoch 105/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5049 - r2_score: 0.1225
Epoch 106/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5950 - r2_score: -0.3305
Epoch 107/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4532 - r2_score: 0.2084
Epoch 108/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 900us/step - loss: 0.5072 - r2_score: 0.0655
Epoch 109/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4790 - r2_score: 0.1107
Epoch 110/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4953 - r2_score: 0.1796
Epoch 111/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5425 - r2_score: -0.0582
Epoch 112/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4385 - r2_score: 0.2635
```

```
Epoch 113/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4855 - r2_score: 0.1660
Epoch 114/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5185 - r2_score: 0.0105
Epoch 115/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5094 - r2_score: -0.0536
Epoch 116/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5618 - r2_score: -0.0378
Epoch 117/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.6182 - r2_score: -0.2541
Epoch 118/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5862 - r2_score: -0.1774
Epoch 119/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.6054 - r2_score: -0.2733
Epoch 120/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4934 - r2_score: 0.1609
Epoch 121/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4829 - r2_score: 0.1181
Epoch 122/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5029 - r2_score: -0.0032
Epoch 123/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5454 - r2_score: -0.0557
Epoch 124/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.7718 - r2_score: -1.0522
Epoch 125/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.7270 - r2_score: -0.7790
Epoch 126/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5903 - r2_score: -0.2101
Epoch 127/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5231 - r2_score: -0.0089
Epoch 128/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4759 - r2_score: 0.0967
Epoch 129/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4794 - r2_score: 0.2497
Epoch 130/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4868 - r2_score: 0.1152
Epoch 131/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.4742 - r2_score: 0.1439
Epoch 132/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4905 - r2_score: 0.1235
Epoch 133/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4708 - r2_score: 0.1189
Epoch 134/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4828 - r2_score: 0.1247
Epoch 135/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4598 - r2_score: 0.2147
Epoch 136/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4877 - r2_score: 0.1831
Epoch 137/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.5260 - r2_score: 0.0039
Epoch 138/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4750 - r2_score: 0.1483
Epoch 139/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5189 - r2_score: 0.0356
Epoch 140/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4873 - r2_score: 0.1257
```

```
Epoch 141/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4663 - r2_score: 0.2472
Epoch 142/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4796 - r2_score: 0.1192
Epoch 143/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4540 - r2_score: 0.2398
Epoch 144/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4815 - r2_score: 0.1799
Epoch 145/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 752us/step - loss: 0.5820 - r2_score: -0.0305
Epoch 146/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4907 - r2_score: 0.1521
Epoch 147/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4627 - r2_score: 0.1718
Epoch 148/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4728 - r2_score: 0.0657
Epoch 149/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.5081 - r2_score: 0.0523
Epoch 150/150
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.4684 - r2_score: 0.2022
```

Out[217…

```
<keras.src.callbacks.history.History at 0x1691e992ba0>
```

В итоге получаем, что для точеного подхода коэффицент $R^2

равен 0.25615040590148686, а для нейронной сети с одним нейроном --

0.2022

5. Постройте диаграмму рассеяния для независимого и зависимого признаков и изобразите линии двух построенных парных регрессий, подписывая оси и рисунок и создавая легенду для линий регрессии.

In [218…

```
y_hat1 = model.predict(train['temp_upper'].values)
```

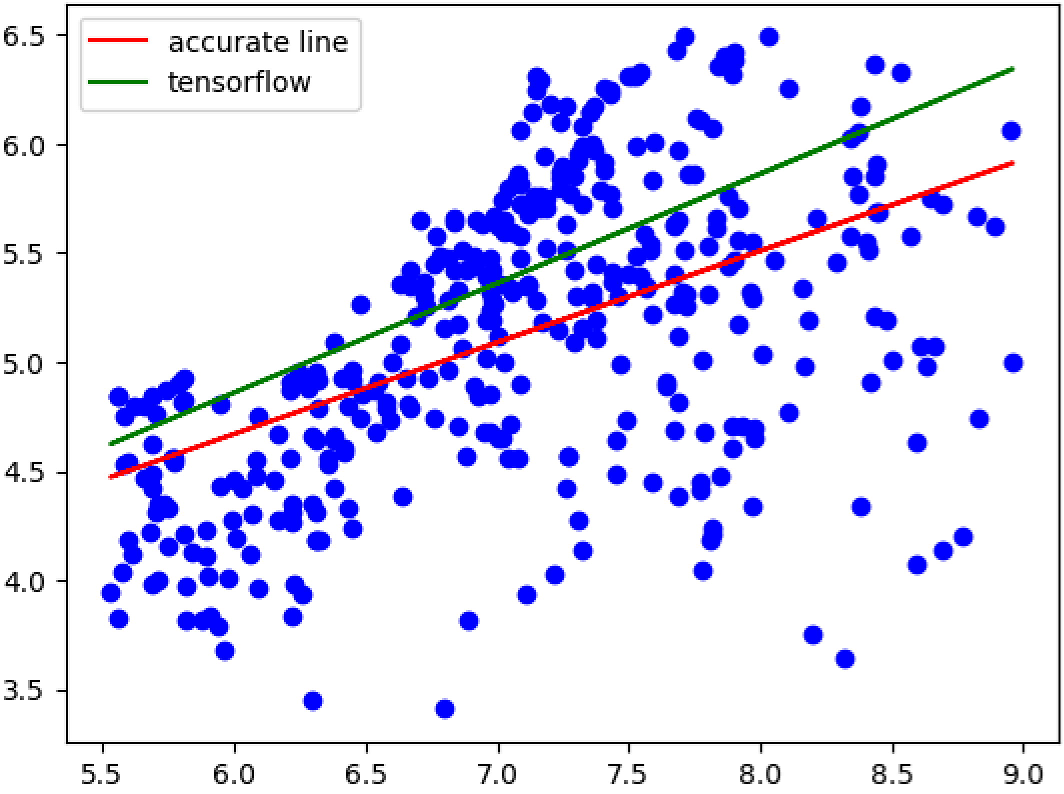
```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
```

In [219…

```
plt.scatter(train['temp_upper'], train['temp_lower'], c='b')
plt.plot(train['temp_upper'], y_hat,  c='r', label = 'accurate line')
plt.plot(train['temp_upper'], y_hat1, c='g', label = 'tensorflow')
plt.legend()
```

Out[219…

```
<matplotlib.legend.Legend at 0x1691fe70620>
```



6. Разбейте набор признаков на обучающую и контрольную выборки. Создайте и адаптируйте нормализующий слой Tensorflow для всех признаков набора данных (за исключением зависимого признака). Нормализуйте зависимый признак.

Независимая переменная: temp_lower

Зависимая переменная: temp_upper

In [222…

```
split_df = np.array_split(df_new, 2)
```

In [236…

```
train = split_df[0]
predict = split_df[1]
display(train)
train.columns
```

	year	doy	temp	temp_upper	temp_lower
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
93	894	106.0	6.98	8.40	5.55
...	...	...	...	...	...
761	1562	99.0	6.29	6.92	5.65
762	1563	99.0	6.13	6.78	5.49
763	1564	109.0	6.18	6.96	5.39
764	1565	114.0	6.16	6.89	5.43
765	1566	97.0	6.28	6.94	5.63

390 rows × 5 columns

Out[236…

```
Index(['year', 'doy', 'temp', 'temp_upper', 'temp_lower'], dtype='object')
```

In [250…

```
normalizer = tf.keras.layers.Normalization()
```

In [251…

```
normalizer.adapt(train[['year', 'doy', 'temp', 'temp_lower']].values)
```

In [252…

```
print(normalizer.mean.numpy())
print(normalizer.variance.numpy())
```

```
[[1298.3513     104.84359      6.1012053    5.123359 ]]
[[3.5666660e+04 4.1224255e+01 4.2067215e-01 4.5764282e-01]]
```

In [253…

```
train[['year', 'doy', 'temp', 'temp_lower']][0:5]
```

Out[253…

	year	doy	temp	temp_lower
63	864	100.0	6.42	4.14
65	866	106.0	6.44	4.77
88	889	104.0	6.83	5.19
90	891	109.0	6.98	5.00
93	894	106.0	6.98	5.55

In [254…

```
normalizer(train[['year', 'doy', 'temp', 'temp_lower']][0:5]).numpy()
```

Out[254…

```
array([[-2.299905  , -0.75438136,  0.4915178 , -1.4536134 ],
       [-2.289315  ,  0.18010904,  0.52235377, -0.52233976],
       [-2.1675293 , -0.13138776,  1.1236557 ,  0.0985093 ],
       [-2.1569393 ,  0.64735425,  1.3549259 , -0.18235102],
       [-2.1410542 ,  0.18010904,  1.3549259 ,  0.6306657 ]],
      dtype=float32)
```

Нормализация зависимого признака

In [277…

```
normalizer1 = tf.keras.layers.Normalization(axis=None, input_shape=(1,))
```

```
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 [278…

```
normalizer1.adapt(train['temp_upper'].values)
```

In [279…

```
print(normalizer1.mean.numpy())
# print(normalizer1.variance.numpy())
```

```
[7.079359]
```

In [280…

```
train['temp_upper']
# normalizer1(train['temp_upper']).numpy()
```

Out[280…

```
63     8.69
65     8.11
88     8.48
90     8.96
93     8.40
       ...
761    6.92
762    6.78
763    6.96
764    6.89
765    6.94
Name: temp_upper, Length: 390, dtype: float64
```

7. Используя созданный нормализующий слой и нормализованный зависимый признак, постройте регресоры на базе следующих моделей множественной регрессии:

1. линейной регрессии
2. гребневой регрессии (L2)

3. лассо регрессии (L1)

Выберите коэффициенты регуляризации l1 и l2 так, чтобы нейронные сети для всех трех моделей обучались (значение ошибки уменьшалось в процессе обучения).

In [327…

```
linear_model = tf.keras.Sequential([
    normalizer1,
    tf.keras.layers.Dense(units=1)
])

linear_model.summary()
```

```
Model: "sequential_10"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━
┃ Layer (type)                         ┃ Output Shape                ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━
│ normalization_5 (Normalization)      │ (None, 1)                   │
├──────────────────────────────────────┼─────────────────────────────┼───────
│ dense_10 (Dense)                     │ (None, 1)                   │
└──────────────────────────────────────┴─────────────────────────────┴───────

 Total params: 5 (24.00 B)

 Trainable params: 2 (8.00 B)

 Non-trainable params: 3 (16.00 B)
```

In [328…

```
linear_model.compile(
    optimizer=tf.optimizers.Adam(learning_rate=0.25),
    loss='mean_absolute_error',
    metrics=['r2_score']
)
```

In [335…

```
%%time
history_linear = linear_model.fit(
    train['temp_upper'], train['temp_lower'],
    epochs=100,
    # подавляем вывод
    # verbose=0,
    # проверка (валидация) на 20% обучающих данных
    validation_split = 0.2)
```

```
Epoch 1/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.5303 - r2_score: -0.0819 - val_los
s: 0.2934 - val_r2_score: 0.4148
Epoch 2/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4968 - r2_score: 0.2049 - val_loss:
0.3036 - val_r2_score: 0.3291
Epoch 3/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5073 - r2_score: 0.1348 - val_loss:
0.3047 - val_r2_score: 0.3544
Epoch 4/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5509 - r2_score: 0.0229 - val_loss:
0.2956 - val_r2_score: 0.3767
Epoch 5/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4999 - r2_score: 0.1675 - val_loss:
0.2942 - val_r2_score: 0.4140
Epoch 6/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4988 - r2_score: 0.1550 - val_loss:
0.2898 - val_r2_score: 0.4069
Epoch 7/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5132 - r2_score: 0.1504 - val_loss:
0.2828 - val_r2_score: 0.4235
Epoch 8/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5173 - r2_score: 0.0574 - val_loss:
0.3175 - val_r2_score: 0.2849
Epoch 9/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5154 - r2_score: 0.0298 - val_loss:
0.2943 - val_r2_score: 0.4112
Epoch 10/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4826 - r2_score: 0.1497 - val_loss:
0.3146 - val_r2_score: 0.3038
Epoch 11/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5685 - r2_score: 0.0404 - val_loss:
0.3019 - val_r2_score: 0.3516
Epoch 12/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5485 - r2_score: -0.0678 - val_los
s: 0.2994 - val_r2_score: 0.3668
Epoch 13/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4963 - r2_score: 0.1202 - val_loss:
0.2868 - val_r2_score: 0.4287
Epoch 14/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5265 - r2_score: 0.1766 - val_loss:
0.2829 - val_r2_score: 0.4205
Epoch 15/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5144 - r2_score: -0.0242 - val_los
s: 0.2991 - val_r2_score: 0.3790
Epoch 16/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5386 - r2_score: 0.1215 - val_loss:
0.2908 - val_r2_score: 0.4051
Epoch 17/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5321 - r2_score: 0.0882 - val_loss:
0.2915 - val_r2_score: 0.4253
Epoch 18/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4997 - r2_score: 0.1189 - val_loss:
0.3204 - val_r2_score: 0.2163
Epoch 19/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5316 - r2_score: 0.1356 - val_loss:
```

```
0.2794 - val_r2_score: 0.4337
Epoch 20/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5698 - r2_score: -0.0054 - val_los
s: 0.2911 - val_r2_score: 0.4268
Epoch 21/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5166 - r2_score: 0.0070 - val_loss:
0.3256 - val_r2_score: 0.2366
Epoch 22/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5392 - r2_score: 0.0889 - val_loss:
0.3177 - val_r2_score: 0.2322
Epoch 23/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5071 - r2_score: 0.1456 - val_loss:
0.3402 - val_r2_score: 0.1612
Epoch 24/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5093 - r2_score: 0.0579 - val_loss:
0.2997 - val_r2_score: 0.3355
Epoch 25/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5022 - r2_score: 0.0830 - val_loss:
0.3061 - val_r2_score: 0.3518
Epoch 26/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4857 - r2_score: 0.1379 - val_loss:
0.3109 - val_r2_score: 0.2781
Epoch 27/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5144 - r2_score: 0.1495 - val_loss:
0.3063 - val_r2_score: 0.3471
Epoch 28/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5086 - r2_score: 0.1129 - val_loss:
0.2773 - val_r2_score: 0.4452
Epoch 29/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5479 - r2_score: -0.0621 - val_los
s: 0.3019 - val_r2_score: 0.3763
Epoch 30/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5096 - r2_score: 0.1096 - val_loss:
0.3164 - val_r2_score: 0.2397
Epoch 31/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5075 - r2_score: 0.1600 - val_loss:
0.3306 - val_r2_score: 0.2132
Epoch 32/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5026 - r2_score: 0.1651 - val_loss:
0.3060 - val_r2_score: 0.3032
Epoch 33/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4940 - r2_score: 0.1840 - val_loss:
0.2931 - val_r2_score: 0.4200
Epoch 34/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5098 - r2_score: 0.1251 - val_loss:
0.2952 - val_r2_score: 0.3944
Epoch 35/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5135 - r2_score: 0.1365 - val_loss:
0.2970 - val_r2_score: 0.3833
Epoch 36/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5126 - r2_score: 0.0811 - val_loss:
0.2876 - val_r2_score: 0.4159
Epoch 37/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5030 - r2_score: 0.1056 - val_loss:
0.2843 - val_r2_score: 0.4427
Epoch 38/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5312 - r2_score: 0.0557 - val_loss:
0.3055 - val_r2_score: 0.3280
Epoch 39/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5074 - r2_score: 0.1329 - val_loss:
0.2840 - val_r2_score: 0.4334
Epoch 40/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4958 - r2_score: 0.0928 - val_loss:
0.3012 - val_r2_score: 0.3767
Epoch 41/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5090 - r2_score: 0.1225 - val_loss:
0.2830 - val_r2_score: 0.4508
Epoch 42/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5131 - r2_score: 0.1405 - val_loss:
0.3057 - val_r2_score: 0.3156
Epoch 43/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5236 - r2_score: 0.1151 - val_loss:
0.3082 - val_r2_score: 0.3368
Epoch 44/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4765 - r2_score: 0.1168 - val_loss:
0.2963 - val_r2_score: 0.3718
Epoch 45/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5432 - r2_score: 0.1229 - val_loss:
0.3055 - val_r2_score: 0.3516
Epoch 46/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5152 - r2_score: 0.0243 - val_loss:
0.2998 - val_r2_score: 0.3450
Epoch 47/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5219 - r2_score: 0.1571 - val_loss:
0.2847 - val_r2_score: 0.4450
Epoch 48/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.4997 - r2_score: 0.1306 - val_loss:
0.2970 - val_r2_score: 0.4025
Epoch 49/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5101 - r2_score: 0.1070 - val_loss:
0.3183 - val_r2_score: 0.2270
Epoch 50/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5246 - r2_score: 0.0286 - val_loss:
0.2934 - val_r2_score: 0.4080
Epoch 51/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5181 - r2_score: 0.1306 - val_loss:
0.2942 - val_r2_score: 0.3976
Epoch 52/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5272 - r2_score: 0.1689 - val_loss:
0.2794 - val_r2_score: 0.4323
Epoch 53/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5256 - r2_score: 0.1518 - val_loss:
0.2854 - val_r2_score: 0.4084
Epoch 54/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5451 - r2_score: -0.0534 - val_los
s: 0.3772 - val_r2_score: -0.0370
Epoch 55/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5629 - r2_score: 0.0184 - val_loss:
0.3342 - val_r2_score: 0.1397
Epoch 56/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.6027 - r2_score: -0.2355 - val_los
s: 0.3084 - val_r2_score: 0.3380
```

```
Epoch 57/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5337 - r2_score: 0.1262 - val_loss:
0.3394 - val_r2_score: 0.1234
Epoch 58/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5189 - r2_score: 0.1373 - val_loss:
0.3004 - val_r2_score: 0.3850
Epoch 59/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5050 - r2_score: 0.2183 - val_loss:
0.2962 - val_r2_score: 0.3644
Epoch 60/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5173 - r2_score: 0.1168 - val_loss:
0.2815 - val_r2_score: 0.4540
Epoch 61/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5759 - r2_score: -0.0543 - val_los
s: 0.2941 - val_r2_score: 0.3612
Epoch 62/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5513 - r2_score: -0.0615 - val_los
s: 0.3037 - val_r2_score: 0.3665
Epoch 63/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5361 - r2_score: 0.1677 - val_loss:
0.3821 - val_r2_score: -0.0648
Epoch 64/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5495 - r2_score: 0.0156 - val_loss:
0.2943 - val_r2_score: 0.3595
Epoch 65/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5012 - r2_score: 0.0935 - val_loss:
0.3211 - val_r2_score: 0.2633
Epoch 66/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5627 - r2_score: 0.0184 - val_loss:
0.3016 - val_r2_score: 0.3430
Epoch 67/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5157 - r2_score: 0.1389 - val_loss:
0.2980 - val_r2_score: 0.3793
Epoch 68/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5218 - r2_score: 0.1357 - val_loss:
0.2809 - val_r2_score: 0.4560
Epoch 69/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5471 - r2_score: 0.0130 - val_loss:
0.3021 - val_r2_score: 0.3730
Epoch 70/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5074 - r2_score: 0.0590 - val_loss:
0.2950 - val_r2_score: 0.3778
Epoch 71/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4951 - r2_score: 0.1521 - val_loss:
0.2976 - val_r2_score: 0.3665
Epoch 72/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5246 - r2_score: 0.1419 - val_loss:
0.2913 - val_r2_score: 0.4090
Epoch 73/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5088 - r2_score: 0.1221 - val_loss:
0.2877 - val_r2_score: 0.4385
Epoch 74/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5189 - r2_score: 0.1184 - val_loss:
0.2902 - val_r2_score: 0.3840
Epoch 75/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5462 - r2_score: 0.0838 - val_loss:
```

```
0.3073 - val_r2_score: 0.3316
Epoch 76/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5110 - r2_score: 0.1087 - val_loss:
0.2843 - val_r2_score: 0.4251
Epoch 77/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5360 - r2_score: 0.2067 - val_loss:
0.2800 - val_r2_score: 0.4571
Epoch 78/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4645 - r2_score: 0.1792 - val_loss:
0.3177 - val_r2_score: 0.2439
Epoch 79/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5016 - r2_score: 0.1419 - val_loss:
0.2825 - val_r2_score: 0.4324
Epoch 80/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4911 - r2_score: 0.2464 - val_loss:
0.2889 - val_r2_score: 0.4160
Epoch 81/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5113 - r2_score: 0.1511 - val_loss:
0.2822 - val_r2_score: 0.4402
Epoch 82/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5367 - r2_score: 0.0400 - val_loss:
0.2925 - val_r2_score: 0.4115
Epoch 83/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5425 - r2_score: -0.0457 - val_los
s: 0.2971 - val_r2_score: 0.3670
Epoch 84/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5247 - r2_score: 0.1046 - val_loss:
0.3108 - val_r2_score: 0.3275
Epoch 85/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.5332 - r2_score: 0.1622 - val_los
s: 0.2885 - val_r2_score: 0.3896
Epoch 86/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5197 - r2_score: 0.1579 - val_loss:
0.2992 - val_r2_score: 0.3621
Epoch 87/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5101 - r2_score: 0.1105 - val_loss:
0.2849 - val_r2_score: 0.4469
Epoch 88/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5011 - r2_score: 0.0712 - val_loss:
0.2886 - val_r2_score: 0.4357
Epoch 89/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4927 - r2_score: 0.1121 - val_loss:
0.2873 - val_r2_score: 0.4182
Epoch 90/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4679 - r2_score: 0.2141 - val_loss:
0.3007 - val_r2_score: 0.3666
Epoch 91/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5138 - r2_score: 0.1847 - val_loss:
0.2787 - val_r2_score: 0.4486
Epoch 92/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5302 - r2_score: 0.0652 - val_loss:
0.3053 - val_r2_score: 0.3028
Epoch 93/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5322 - r2_score: 0.0769 - val_loss:
0.3029 - val_r2_score: 0.3645
Epoch 94/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5251 - r2_score: 0.1460 - val_loss:
0.2808 - val_r2_score: 0.4430
Epoch 95/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5053 - r2_score: 0.1101 - val_loss:
0.3164 - val_r2_score: 0.2438
Epoch 96/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5088 - r2_score: 0.1173 - val_loss:
0.2872 - val_r2_score: 0.4163
Epoch 97/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5034 - r2_score: 0.1556 - val_loss:
0.2989 - val_r2_score: 0.3727
Epoch 98/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5241 - r2_score: 0.1679 - val_loss:
0.2828 - val_r2_score: 0.4220
Epoch 99/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5073 - r2_score: 0.1454 - val_loss:
0.2969 - val_r2_score: 0.3976
Epoch 100/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4782 - r2_score: 0.2185 - val_loss:
0.2945 - val_r2_score: 0.3872
CPU times: total: 3.25 s
Wall time: 10.4 s
```

In [336…

```
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()
```

Out[336…

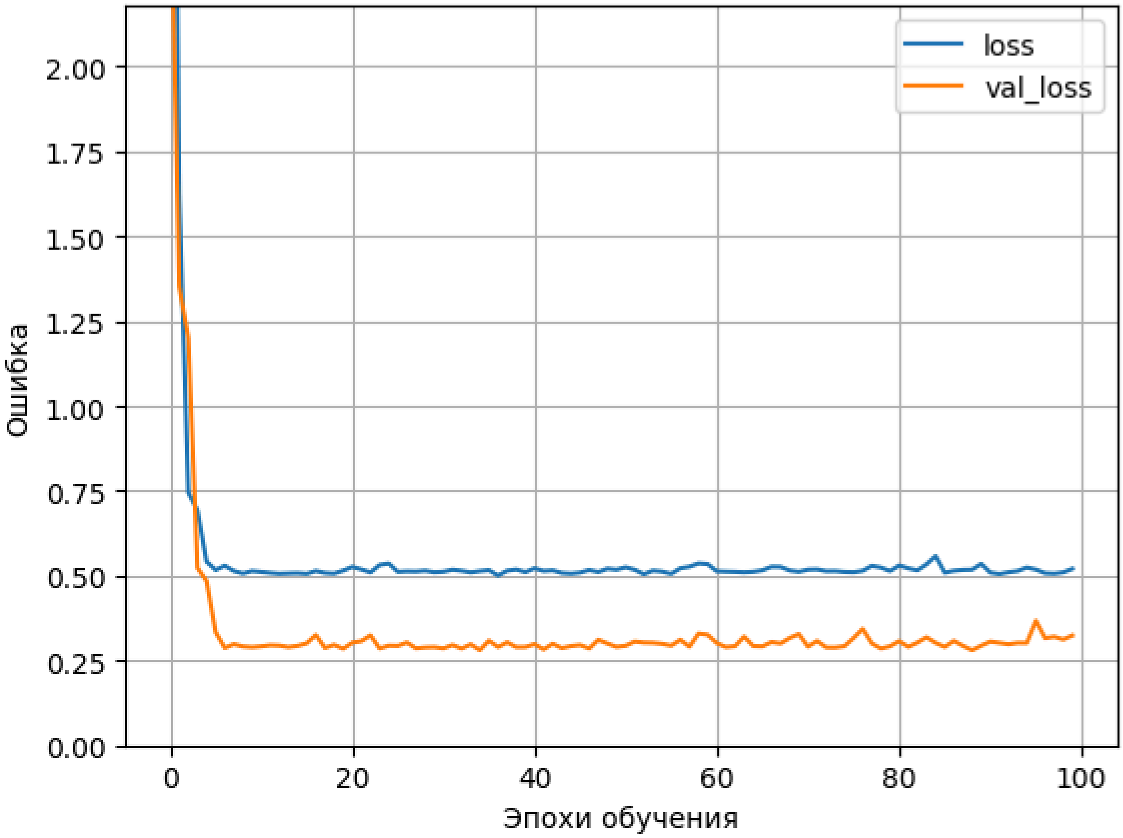
	loss	val_loss	epoch
95	0.518174	0.367523	95
96	0.507600	0.315754	96
97	0.506301	0.320101	97
98	0.509951	0.311963	98
99	0.520420	0.323843	99

In [339…

```
def plot_loss(history):
  plt.plot(history.history['loss'], label='loss')
  plt.plot(history.history['val_loss'], label='val_loss')
  # plt.plot(history.history['r2_score'], label='r2_score')
  # plt.plot(history.history['val_r2_score'], label='r2_score')
  plt.ylim([0, max(history.history['loss'])*0.5])
  plt.xlabel('Эпохи обучения')
  plt.ylabel('Ошибка')
  plt.legend()
  plt.grid(True)
```

In [340…

```
plot_loss(history)
```



In [341…

```
test_results = {} # пустой словарь

test_results['feature_model'] = linear_model.evaluate(
    train['temp_upper'],
    train['temp_lower'], verbose=0)
```

In [342…

```
x = tf.linspace(4., 9., 51)
y = linear_model.predict(x)
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
```

In [346…

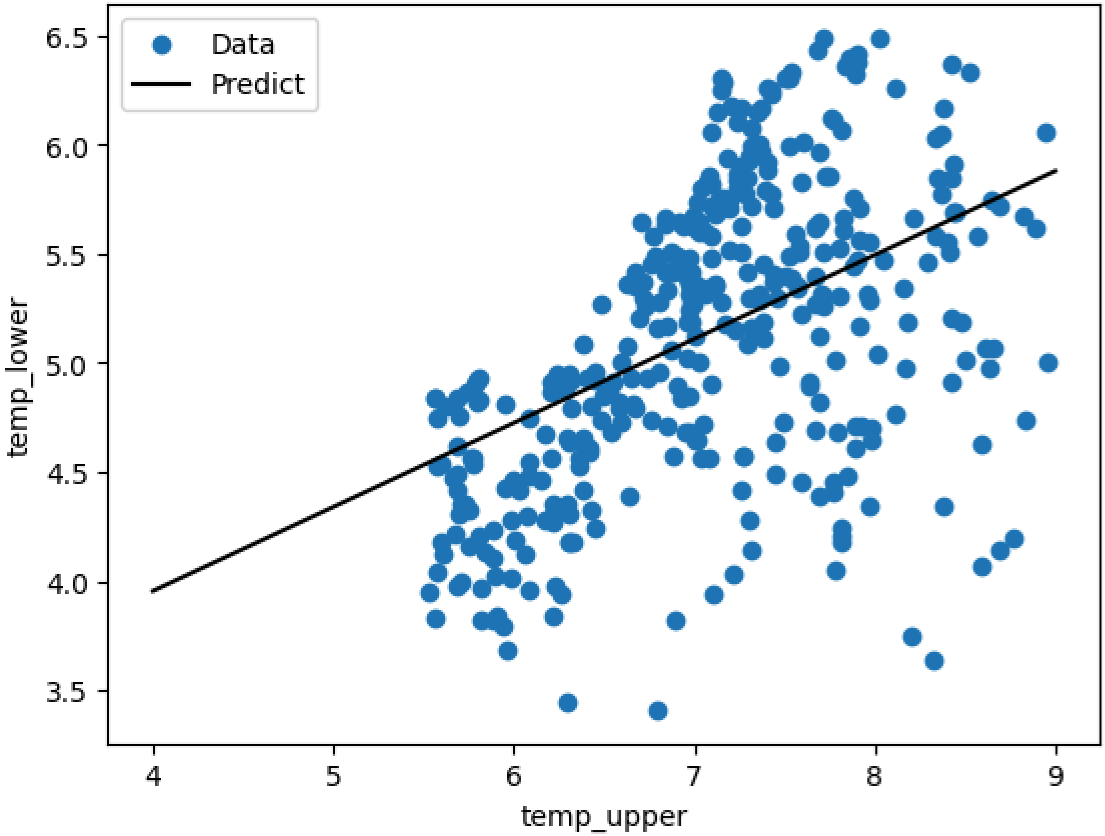
```
def plot_rm(x, y):
  plt.scatter(train['temp_upper'], train['temp_lower'], label='Data')
  plt.plot(x, y, color='k', label='Predict')
  plt.xlabel('temp_upper')
  plt.ylabel('temp_lower')
  plt.legend()
```

In [369…

```
def plot_all_regressions(x_lin, y_lin, x_l1, y_l1, x_l2, y_l2):
  plt.scatter(train['temp_upper'], train['temp_lower'], label='Data')
  plt.plot(x_lin, y_lin, color='k', label='Linear')
  plt.plot(x_l1, y_l1, color='r', label='L1')
  plt.plot(x_l2, y_l2, color='g', label='L2')
  plt.xlabel('temp_upper')
  plt.ylabel('temp_lower')
  plt.title('All plotted regressions(linear, L1, L2)')
  plt.legend()
```

In [347…

```
plot_rm(x, y)
```



L1

In [348…

```
df_normalizer = tf.keras.layers.Normalization(axis=None, input_shape = (1,))
df_normalizer.adapt(train['temp_upper'].values)
print(df_normalizer.mean.numpy())
# print(df_normalizer.variance.numpy())
```

```
[7.079359]
```

```
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 [349…

```
l1_model = tf.keras.Sequential([
    df_normalizer,
    tf.keras.layers.Dense(units=1,
                          kernel_regularizer=tf.keras.regularizers.L1(l1=0.01))
])
```

In [350…

```
l1_model.compile(
    optimizer=tf.optimizers.Adam(learning_rate=0.25),
    loss='mean_absolute_error',
    metrics=['r2_score']
)
```

In [351…

```
%%time
history = l1_model.fit(
    train['temp_upper'], train['temp_lower'],
    epochs=100,
    # подавляем вывод
    verbose=1,
    # проверка (валидация) на 20% обучающих данных
    validation_split = 0.2)
```

```
Epoch 1/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 4.2199 - r2_score: -44.8796 - val_lo
ss: 5.1591 - val_r2_score: -139.8089
Epoch 2/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 2.2635 - r2_score: -13.0760 - val_los
s: 1.3604 - val_r2_score: -9.3439
Epoch 3/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.6920 - r2_score: -0.3722 - val_los
s: 1.4879 - val_r2_score: -12.0604
Epoch 4/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.7967 - r2_score: -1.1678 - val_los
s: 0.4119 - val_r2_score: -0.2089
Epoch 5/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5107 - r2_score: 0.0170 - val_loss:
0.4556 - val_r2_score: -0.4092
Epoch 6/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5341 - r2_score: 0.1298 - val_loss:
0.3242 - val_r2_score: 0.2106
Epoch 7/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5304 - r2_score: 0.1497 - val_loss:
0.2875 - val_r2_score: 0.4248
Epoch 8/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4948 - r2_score: 0.2016 - val_loss:
0.2944 - val_r2_score: 0.4140
Epoch 9/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5015 - r2_score: 0.1133 - val_loss:
0.2911 - val_r2_score: 0.4130
Epoch 10/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4943 - r2_score: 0.1597 - val_loss:
0.2938 - val_r2_score: 0.4052
Epoch 11/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4937 - r2_score: 0.2129 - val_loss:
0.2968 - val_r2_score: 0.4052
Epoch 12/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4894 - r2_score: 0.1656 - val_loss:
0.2825 - val_r2_score: 0.4501
Epoch 13/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5386 - r2_score: -0.0212 - val_los
s: 0.3170 - val_r2_score: 0.3082
Epoch 14/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5362 - r2_score: 0.1748 - val_loss:
0.2867 - val_r2_score: 0.4533
Epoch 15/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5422 - r2_score: 0.0773 - val_loss:
0.2903 - val_r2_score: 0.4282
Epoch 16/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5003 - r2_score: 0.0795 - val_loss:
0.3159 - val_r2_score: 0.3202
Epoch 17/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5642 - r2_score: 0.1142 - val_loss:
0.2824 - val_r2_score: 0.4517
Epoch 18/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5456 - r2_score: 0.0537 - val_loss:
0.3069 - val_r2_score: 0.3365
Epoch 19/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5261 - r2_score: 0.1937 - val_loss:
```

```
0.3027 - val_r2_score: 0.3933
Epoch 20/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5170 - r2_score: 0.1049 - val_loss:
0.3138 - val_r2_score: 0.2735
Epoch 21/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5253 - r2_score: 0.1245 - val_loss:
0.2917 - val_r2_score: 0.4372
Epoch 22/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5282 - r2_score: 0.0553 - val_loss:
0.2962 - val_r2_score: 0.4244
Epoch 23/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5412 - r2_score: -0.0036 - val_los
s: 0.3027 - val_r2_score: 0.3662
Epoch 24/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5224 - r2_score: 0.0891 - val_loss:
0.2952 - val_r2_score: 0.4229
Epoch 25/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5052 - r2_score: 0.1144 - val_loss:
0.3065 - val_r2_score: 0.3454
Epoch 26/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5035 - r2_score: 0.1643 - val_loss:
0.2858 - val_r2_score: 0.4546
Epoch 27/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5247 - r2_score: 0.1462 - val_loss:
0.2908 - val_r2_score: 0.4244
Epoch 28/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4993 - r2_score: 0.0653 - val_loss:
0.2939 - val_r2_score: 0.4107
Epoch 29/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5082 - r2_score: 0.1516 - val_loss:
0.3042 - val_r2_score: 0.3889
Epoch 30/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5138 - r2_score: 0.1368 - val_loss:
0.2961 - val_r2_score: 0.3710
Epoch 31/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4888 - r2_score: 0.1647 - val_loss:
0.3132 - val_r2_score: 0.3320
Epoch 32/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5129 - r2_score: 0.1499 - val_loss:
0.2973 - val_r2_score: 0.3703
Epoch 33/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4949 - r2_score: 0.1642 - val_loss:
0.2996 - val_r2_score: 0.3756
Epoch 34/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5118 - r2_score: 0.1555 - val_loss:
0.3329 - val_r2_score: 0.2337
Epoch 35/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5274 - r2_score: 0.0769 - val_loss:
0.3557 - val_r2_score: 0.0408
Epoch 36/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5661 - r2_score: 0.0749 - val_loss:
0.3160 - val_r2_score: 0.3184
Epoch 37/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5426 - r2_score: 0.1128 - val_loss:
0.2817 - val_r2_score: 0.4496
Epoch 38/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5229 - r2_score: 0.1660 - val_loss:
0.3153 - val_r2_score: 0.2811
Epoch 39/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5112 - r2_score: 0.1347 - val_loss:
0.2972 - val_r2_score: 0.4214
Epoch 40/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5052 - r2_score: 0.2121 - val_loss:
0.2918 - val_r2_score: 0.3993
Epoch 41/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5243 - r2_score: 0.0339 - val_loss:
0.3079 - val_r2_score: 0.3365
Epoch 42/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5162 - r2_score: 0.1294 - val_loss:
0.2831 - val_r2_score: 0.4573
Epoch 43/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5462 - r2_score: -0.0220 - val_los
s: 0.2869 - val_r2_score: 0.4417
Epoch 44/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5204 - r2_score: 0.0776 - val_loss:
0.3000 - val_r2_score: 0.3936
Epoch 45/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5101 - r2_score: 0.1940 - val_loss:
0.3010 - val_r2_score: 0.3640
Epoch 46/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5161 - r2_score: 0.1662 - val_loss:
0.2995 - val_r2_score: 0.3875
Epoch 47/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5152 - r2_score: 0.1626 - val_loss:
0.2977 - val_r2_score: 0.3618
Epoch 48/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4764 - r2_score: 0.1648 - val_loss:
0.2976 - val_r2_score: 0.4177
Epoch 49/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5218 - r2_score: 0.1055 - val_loss:
0.2882 - val_r2_score: 0.4508
Epoch 50/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5255 - r2_score: 0.1674 - val_loss:
0.3129 - val_r2_score: 0.2968
Epoch 51/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5298 - r2_score: 0.0728 - val_loss:
0.2898 - val_r2_score: 0.4257
Epoch 52/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5157 - r2_score: 0.0985 - val_loss:
0.3055 - val_r2_score: 0.3772
Epoch 53/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4943 - r2_score: 0.2006 - val_loss:
0.3096 - val_r2_score: 0.3205
Epoch 54/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5586 - r2_score: 0.1225 - val_los
s: 0.3383 - val_r2_score: 0.2010
Epoch 55/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5542 - r2_score: -0.1576 - val_los
s: 0.3161 - val_r2_score: 0.2887
Epoch 56/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5329 - r2_score: 0.1272 - val_loss:
0.2901 - val_r2_score: 0.4384
```

```
Epoch 57/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5363 - r2_score: 0.1194 - val_loss:
0.3338 - val_r2_score: 0.1586
Epoch 58/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5253 - r2_score: 0.1257 - val_loss:
0.2882 - val_r2_score: 0.4143
Epoch 59/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5471 - r2_score: 0.1338 - val_loss:
0.2901 - val_r2_score: 0.4270
Epoch 60/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5113 - r2_score: 0.0738 - val_loss:
0.3069 - val_r2_score: 0.3702
Epoch 61/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5220 - r2_score: 0.1362 - val_loss:
0.3313 - val_r2_score: 0.1715
Epoch 62/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5229 - r2_score: 0.1586 - val_loss:
0.2884 - val_r2_score: 0.4491
Epoch 63/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5017 - r2_score: 0.1616 - val_loss:
0.2939 - val_r2_score: 0.4037
Epoch 64/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5247 - r2_score: 0.1043 - val_loss:
0.2897 - val_r2_score: 0.4433
Epoch 65/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5380 - r2_score: 0.0897 - val_loss:
0.2968 - val_r2_score: 0.4109
Epoch 66/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5203 - r2_score: 0.1562 - val_loss:
0.2999 - val_r2_score: 0.3760
Epoch 67/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5099 - r2_score: 0.1146 - val_loss:
0.2876 - val_r2_score: 0.4524
Epoch 68/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5512 - r2_score: -0.0235 - val_los
s: 0.2974 - val_r2_score: 0.4026
Epoch 69/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5218 - r2_score: 0.0391 - val_loss:
0.3158 - val_r2_score: 0.2838
Epoch 70/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5381 - r2_score: 0.1379 - val_loss:
0.2816 - val_r2_score: 0.4497
Epoch 71/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5098 - r2_score: 0.0973 - val_loss:
0.3200 - val_r2_score: 0.2946
Epoch 72/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5394 - r2_score: 0.0962 - val_loss:
0.3028 - val_r2_score: 0.3509
Epoch 73/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5073 - r2_score: 0.1916 - val_loss:
0.2874 - val_r2_score: 0.4405
Epoch 74/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5160 - r2_score: -0.0273 - val_los
s: 0.2855 - val_r2_score: 0.4555
Epoch 75/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5409 - r2_score: 0.0199 - val_loss:
```

```
0.2981 - val_r2_score: 0.4000
Epoch 76/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5113 - r2_score: 0.1416 - val_loss:
0.2945 - val_r2_score: 0.4059
Epoch 77/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4824 - r2_score: 0.1731 - val_los
s: 0.3034 - val_r2_score: 0.3320
Epoch 78/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4951 - r2_score: 0.1919 - val_loss:
0.2962 - val_r2_score: 0.4227
Epoch 79/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5206 - r2_score: 0.1665 - val_loss:
0.3030 - val_r2_score: 0.3444
Epoch 80/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5343 - r2_score: 0.1277 - val_loss:
0.2904 - val_r2_score: 0.4263
Epoch 81/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5185 - r2_score: 0.1042 - val_loss:
0.2943 - val_r2_score: 0.4047
Epoch 82/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5223 - r2_score: 0.1911 - val_loss:
0.2958 - val_r2_score: 0.3897
Epoch 83/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4891 - r2_score: 0.1967 - val_loss:
0.2919 - val_r2_score: 0.4213
Epoch 84/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5295 - r2_score: 0.1433 - val_loss:
0.2836 - val_r2_score: 0.4484
Epoch 85/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.5008 - r2_score: 0.1824 - val_los
s: 0.3042 - val_r2_score: 0.3582
Epoch 86/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5038 - r2_score: 0.1266 - val_loss:
0.2886 - val_r2_score: 0.4032
Epoch 87/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5300 - r2_score: 0.0900 - val_loss:
0.2936 - val_r2_score: 0.4318
Epoch 88/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5372 - r2_score: 0.0968 - val_loss:
0.3046 - val_r2_score: 0.3391
Epoch 89/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5369 - r2_score: 0.1059 - val_loss:
0.2902 - val_r2_score: 0.4383
Epoch 90/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5120 - r2_score: 0.0983 - val_loss:
0.2931 - val_r2_score: 0.4134
Epoch 91/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5061 - r2_score: 0.1234 - val_loss:
0.3047 - val_r2_score: 0.3293
Epoch 92/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5174 - r2_score: 0.1809 - val_loss:
0.2930 - val_r2_score: 0.4282
Epoch 93/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5375 - r2_score: 0.1032 - val_loss:
0.2864 - val_r2_score: 0.4283
Epoch 94/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5121 - r2_score: 0.1541 - val_loss:
0.2922 - val_r2_score: 0.4113
Epoch 95/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4903 - r2_score: 0.1613 - val_loss:
0.3020 - val_r2_score: 0.3707
Epoch 96/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5152 - r2_score: 0.1497 - val_loss:
0.2903 - val_r2_score: 0.4435
Epoch 97/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5268 - r2_score: 0.1451 - val_loss:
0.2929 - val_r2_score: 0.4017
Epoch 98/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5035 - r2_score: 0.1257 - val_loss:
0.2893 - val_r2_score: 0.4228
Epoch 99/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5438 - r2_score: 0.1149 - val_los
s: 0.2969 - val_r2_score: 0.3960
Epoch 100/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5054 - r2_score: 0.1416 - val_los
s: 0.2850 - val_r2_score: 0.4259
CPU times: total: 3.09 s
Wall time: 11.2 s
```

In [352…

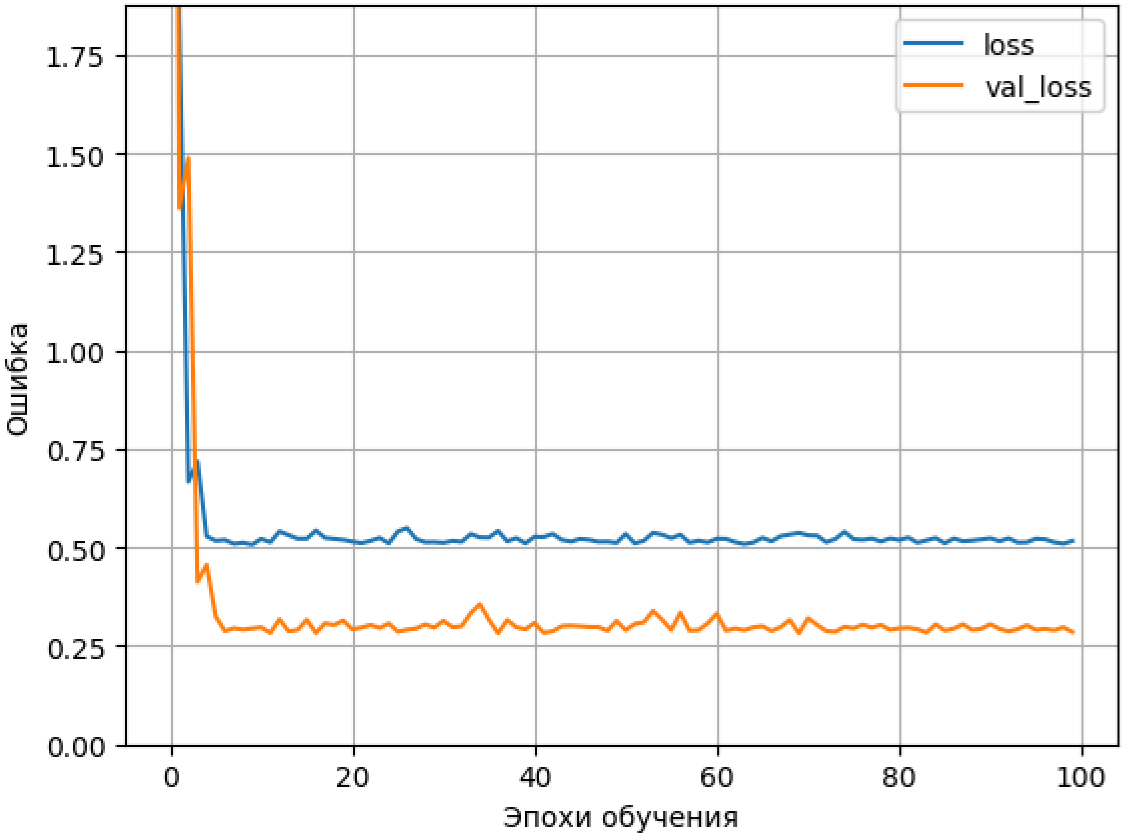
```
l1_model.layers[1].kernel
```

Out[352…

```
<KerasVariable shape=(1, 1), dtype=float32, path=sequential_11/dense_11/kernel>
```

In [353…

```
plot_loss(history)
```



In [354…

```
test_results = {} # пустой словарь

test_results['feature_model'] = linear_model.evaluate(
    train['temp_upper'],
    train['temp_lower'], verbose=0)
```

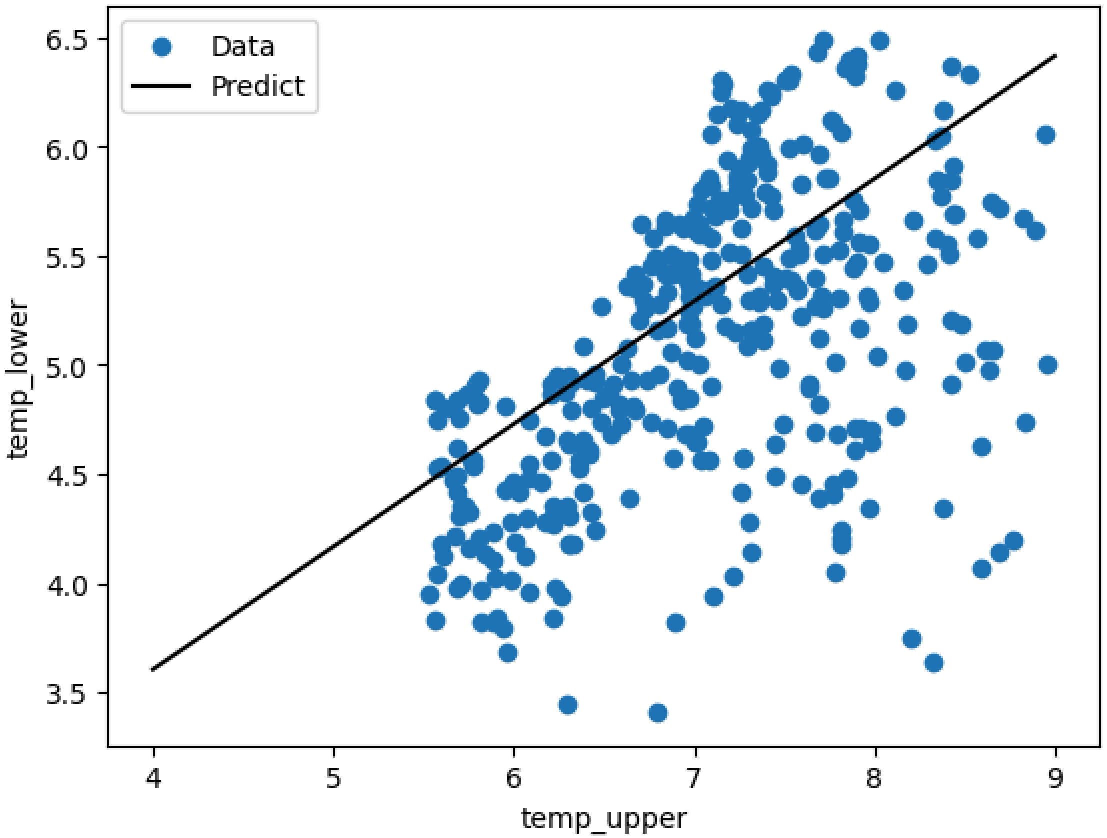
In [355…

```
x = tf.linspace(4., 9., 51)
y = l1_model.predict(x)
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
```

In [357…

```
plot_rm(x, y)
```



L2

In [358…

```
l2_model = tf.keras.Sequential([
    df_normalizer,
    tf.keras.layers.Dense(units=1,
                          kernel_regularizer=tf.keras.regularizers.L2(l2=0.01))
])
```

In [362…

```
l2_model.compile(
    optimizer=tf.optimizers.Adam(learning_rate=0.25),
    loss='mean_absolute_error',
    metrics=['r2_score']
)
```

In [363…

```
%%time
history = l2_model.fit(
    train['temp_upper'], train['temp_lower'],
    epochs=100,
    # подавляем вывод
    verbose=1,
    # проверка (валидация) на 20% обучающих данных
    validation_split = 0.2)
```

```
Epoch 1/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 4.4940 - r2_score: -39.7843 - val_lo
ss: 4.5864 - val_r2_score: -107.6450
Epoch 2/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 2.1786 - r2_score: -12.7533 - val_los
s: 1.6297 - val_r2_score: -14.0223
Epoch 3/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.7650 - r2_score: -0.9503 - val_los
s: 1.4146 - val_r2_score: -11.1917
Epoch 4/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.7557 - r2_score: -1.1300 - val_los
s: 0.6157 - val_r2_score: -1.4903
Epoch 5/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5545 - r2_score: -0.0192 - val_los
s: 0.4784 - val_r2_score: -0.5512
Epoch 6/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5471 - r2_score: -0.0356 - val_los
s: 0.3227 - val_r2_score: 0.2082
Epoch 7/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5085 - r2_score: 0.0746 - val_loss:
0.2942 - val_r2_score: 0.4085
Epoch 8/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5116 - r2_score: 0.2032 - val_loss:
0.2941 - val_r2_score: 0.4206
Epoch 9/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5065 - r2_score: 0.0878 - val_loss:
0.3159 - val_r2_score: 0.2461
Epoch 10/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5031 - r2_score: 0.1240 - val_loss:
0.3373 - val_r2_score: 0.1878
Epoch 11/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5145 - r2_score: 0.2084 - val_loss:
0.3142 - val_r2_score: 0.2576
Epoch 12/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5323 - r2_score: 0.1567 - val_loss:
0.3050 - val_r2_score: 0.3683
Epoch 13/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5414 - r2_score: 0.0458 - val_loss:
0.2907 - val_r2_score: 0.4353
Epoch 14/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5131 - r2_score: 0.0430 - val_loss:
0.3204 - val_r2_score: 0.2211
Epoch 15/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5341 - r2_score: 0.1287 - val_loss:
0.2878 - val_r2_score: 0.4437
Epoch 16/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4819 - r2_score: 0.1859 - val_loss:
0.2931 - val_r2_score: 0.4186
Epoch 17/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4947 - r2_score: 0.2486 - val_loss:
0.2945 - val_r2_score: 0.3723
Epoch 18/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4756 - r2_score: 0.2031 - val_loss:
0.2966 - val_r2_score: 0.3854
Epoch 19/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5408 - r2_score: 0.1303 - val_loss:
```

```
0.3050 - val_r2_score: 0.3690
Epoch 20/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5498 - r2_score: 0.0828 - val_loss:
0.2879 - val_r2_score: 0.4083
Epoch 21/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4795 - r2_score: 0.1765 - val_loss:
0.2951 - val_r2_score: 0.4023
Epoch 22/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5078 - r2_score: 0.1922 - val_loss:
0.2881 - val_r2_score: 0.4032
Epoch 23/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5105 - r2_score: 0.1807 - val_loss:
0.2920 - val_r2_score: 0.4108
Epoch 24/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5313 - r2_score: 0.0789 - val_loss:
0.2862 - val_r2_score: 0.4459
Epoch 25/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4911 - r2_score: 0.1676 - val_loss:
0.3028 - val_r2_score: 0.3714
Epoch 26/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5069 - r2_score: 0.1785 - val_loss:
0.2997 - val_r2_score: 0.3431
Epoch 27/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5011 - r2_score: 0.1213 - val_loss:
0.2999 - val_r2_score: 0.3937
Epoch 28/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5160 - r2_score: 0.0894 - val_loss:
0.3027 - val_r2_score: 0.3534
Epoch 29/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5048 - r2_score: 0.1856 - val_loss:
0.2941 - val_r2_score: 0.4204
Epoch 30/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5107 - r2_score: 0.2187 - val_loss:
0.3053 - val_r2_score: 0.3441
Epoch 31/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5024 - r2_score: 0.1966 - val_loss:
0.2864 - val_r2_score: 0.4302
Epoch 32/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.4925 - r2_score: 0.0949 - val_loss:
0.2814 - val_r2_score: 0.4571
Epoch 33/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5329 - r2_score: 0.0143 - val_loss:
0.3008 - val_r2_score: 0.3649
Epoch 34/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5749 - r2_score: 0.0373 - val_loss:
0.2804 - val_r2_score: 0.4405
Epoch 35/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5460 - r2_score: -0.0987 - val_los
s: 0.2959 - val_r2_score: 0.3923
Epoch 36/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5232 - r2_score: 0.1780 - val_loss:
0.2879 - val_r2_score: 0.3931
Epoch 37/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5532 - r2_score: 0.0707 - val_loss:
0.3068 - val_r2_score: 0.3342
Epoch 38/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5295 - r2_score: 0.1319 - val_loss:
0.2830 - val_r2_score: 0.4481
Epoch 39/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5074 - r2_score: 0.0989 - val_loss:
0.2957 - val_r2_score: 0.4178
Epoch 40/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5192 - r2_score: 0.1457 - val_loss:
0.3040 - val_r2_score: 0.3434
Epoch 41/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5327 - r2_score: 0.1606 - val_loss:
0.3017 - val_r2_score: 0.3851
Epoch 42/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5363 - r2_score: 0.0276 - val_loss:
0.3069 - val_r2_score: 0.2978
Epoch 43/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5319 - r2_score: 0.0486 - val_los
s: 0.3047 - val_r2_score: 0.3555
Epoch 44/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5500 - r2_score: 0.1380 - val_loss:
0.2813 - val_r2_score: 0.4414
Epoch 45/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5374 - r2_score: 0.0086 - val_loss:
0.2822 - val_r2_score: 0.4436
Epoch 46/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5060 - r2_score: 0.1043 - val_loss:
0.3016 - val_r2_score: 0.3587
Epoch 47/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5064 - r2_score: 0.1735 - val_loss:
0.2839 - val_r2_score: 0.4485
Epoch 48/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5422 - r2_score: 0.0838 - val_los
s: 0.2964 - val_r2_score: 0.4116
Epoch 49/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5444 - r2_score: 0.0567 - val_loss:
0.3138 - val_r2_score: 0.2689
Epoch 50/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5322 - r2_score: 0.1425 - val_los
s: 0.2884 - val_r2_score: 0.4307
Epoch 51/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4821 - r2_score: 0.2036 - val_loss:
0.3028 - val_r2_score: 0.3263
Epoch 52/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5202 - r2_score: 0.1220 - val_loss:
0.3014 - val_r2_score: 0.3891
Epoch 53/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5320 - r2_score: 0.0624 - val_loss:
0.2923 - val_r2_score: 0.3771
Epoch 54/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5262 - r2_score: 0.1228 - val_los
s: 0.2984 - val_r2_score: 0.3748
Epoch 55/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.5065 - r2_score: 0.1832 - val_los
s: 0.2963 - val_r2_score: 0.4090
Epoch 56/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5372 - r2_score: 0.1907 - val_loss:
0.2931 - val_r2_score: 0.3870
```

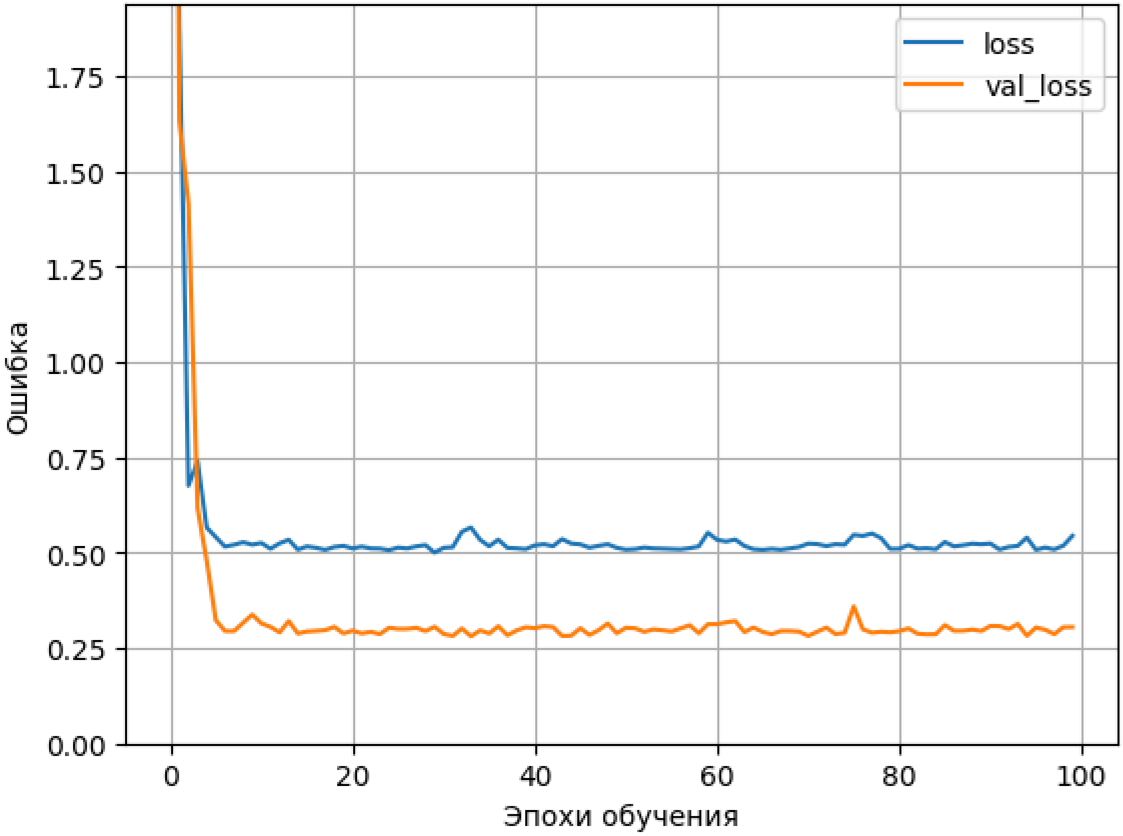
```
Epoch 57/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.4940 - r2_score: 0.0794 - val_loss:
0.3011 - val_r2_score: 0.3915
Epoch 58/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.4665 - r2_score: 0.2171 - val_loss:
0.3094 - val_r2_score: 0.3138
Epoch 59/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5337 - r2_score: 0.1081 - val_loss:
0.2884 - val_r2_score: 0.4362
Epoch 60/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5515 - r2_score: -0.1005 - val_los
s: 0.3122 - val_r2_score: 0.3014
Epoch 61/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5248 - r2_score: 0.1123 - val_loss:
0.3121 - val_r2_score: 0.2685
Epoch 62/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5403 - r2_score: 0.0512 - val_los
s: 0.3168 - val_r2_score: 0.3056
Epoch 63/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5222 - r2_score: 0.0083 - val_loss:
0.3205 - val_r2_score: 0.2209
Epoch 64/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5214 - r2_score: 0.1614 - val_loss:
0.2916 - val_r2_score: 0.3965
Epoch 65/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4884 - r2_score: 0.1096 - val_los
s: 0.3034 - val_r2_score: 0.3793
Epoch 66/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5201 - r2_score: 0.1209 - val_loss:
0.2919 - val_r2_score: 0.3955
Epoch 67/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5352 - r2_score: 0.1453 - val_los
s: 0.2860 - val_r2_score: 0.4315
Epoch 68/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5110 - r2_score: 0.1942 - val_loss:
0.2944 - val_r2_score: 0.3860
Epoch 69/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5051 - r2_score: 0.1227 - val_loss:
0.2943 - val_r2_score: 0.4184
Epoch 70/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5190 - r2_score: 0.1215 - val_loss:
0.2927 - val_r2_score: 0.4281
Epoch 71/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5233 - r2_score: 0.1145 - val_loss:
0.2813 - val_r2_score: 0.4496
Epoch 72/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5239 - r2_score: 0.0728 - val_loss:
0.2925 - val_r2_score: 0.4033
Epoch 73/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5087 - r2_score: 0.0980 - val_loss:
0.3032 - val_r2_score: 0.3244
Epoch 74/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5144 - r2_score: 0.1601 - val_loss:
0.2861 - val_r2_score: 0.4147
Epoch 75/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5194 - r2_score: 0.0616 - val_loss:
```

```
0.2896 - val_r2_score: 0.4386
Epoch 76/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5228 - r2_score: 0.1059 - val_loss:
0.3588 - val_r2_score: 0.0381
Epoch 77/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 0.5248 - r2_score: 0.0425 - val_loss:
0.2988 - val_r2_score: 0.3717
Epoch 78/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5514 - r2_score: 0.1110 - val_loss:
0.2899 - val_r2_score: 0.4206
Epoch 79/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5386 - r2_score: 0.0838 - val_loss:
0.2922 - val_r2_score: 0.4296
Epoch 80/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5102 - r2_score: 0.1645 - val_loss:
0.2909 - val_r2_score: 0.4196
Epoch 81/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5092 - r2_score: 0.1768 - val_los
s: 0.2938 - val_r2_score: 0.4053
Epoch 82/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4965 - r2_score: 0.2034 - val_loss:
0.3012 - val_r2_score: 0.3304
Epoch 83/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5010 - r2_score: 0.1266 - val_loss:
0.2871 - val_r2_score: 0.4412
Epoch 84/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5299 - r2_score: 0.0667 - val_loss:
0.2855 - val_r2_score: 0.4440
Epoch 85/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5110 - r2_score: 0.1707 - val_loss:
0.2858 - val_r2_score: 0.4504
Epoch 86/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5011 - r2_score: 0.1871 - val_loss:
0.3097 - val_r2_score: 0.3166
Epoch 87/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5180 - r2_score: 0.1456 - val_loss:
0.2946 - val_r2_score: 0.3818
Epoch 88/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5151 - r2_score: 0.1171 - val_loss:
0.2949 - val_r2_score: 0.4004
Epoch 89/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5130 - r2_score: 0.1040 - val_loss:
0.2979 - val_r2_score: 0.3730
Epoch 90/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.5381 - r2_score: 0.1130 - val_loss:
0.2943 - val_r2_score: 0.4162
Epoch 91/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5083 - r2_score: 0.1885 - val_loss:
0.3075 - val_r2_score: 0.3020
Epoch 92/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5068 - r2_score: 0.1130 - val_loss:
0.3071 - val_r2_score: 0.3552
Epoch 93/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4937 - r2_score: 0.1926 - val_loss:
0.2993 - val_r2_score: 0.3612
Epoch 94/100
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4876 - r2_score: 0.1706 - val_loss:
0.3127 - val_r2_score: 0.3208
Epoch 95/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5442 - r2_score: 0.0475 - val_loss:
0.2816 - val_r2_score: 0.4574
Epoch 96/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.4801 - r2_score: 0.1860 - val_loss:
0.3035 - val_r2_score: 0.3484
Epoch 97/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - loss: 0.5100 - r2_score: 0.0833 - val_loss:
0.2975 - val_r2_score: 0.4058
Epoch 98/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.5127 - r2_score: 0.1457 - val_los
s: 0.2862 - val_r2_score: 0.4293
Epoch 99/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.5072 - r2_score: 0.1187 - val_loss:
0.3042 - val_r2_score: 0.3554
Epoch 100/100
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5534 - r2_score: 0.0763 - val_los
s: 0.3045 - val_r2_score: 0.3143
CPU times: total: 4.02 s
Wall time: 13.5 s
```

In [364…

```
plot_loss(history)
```



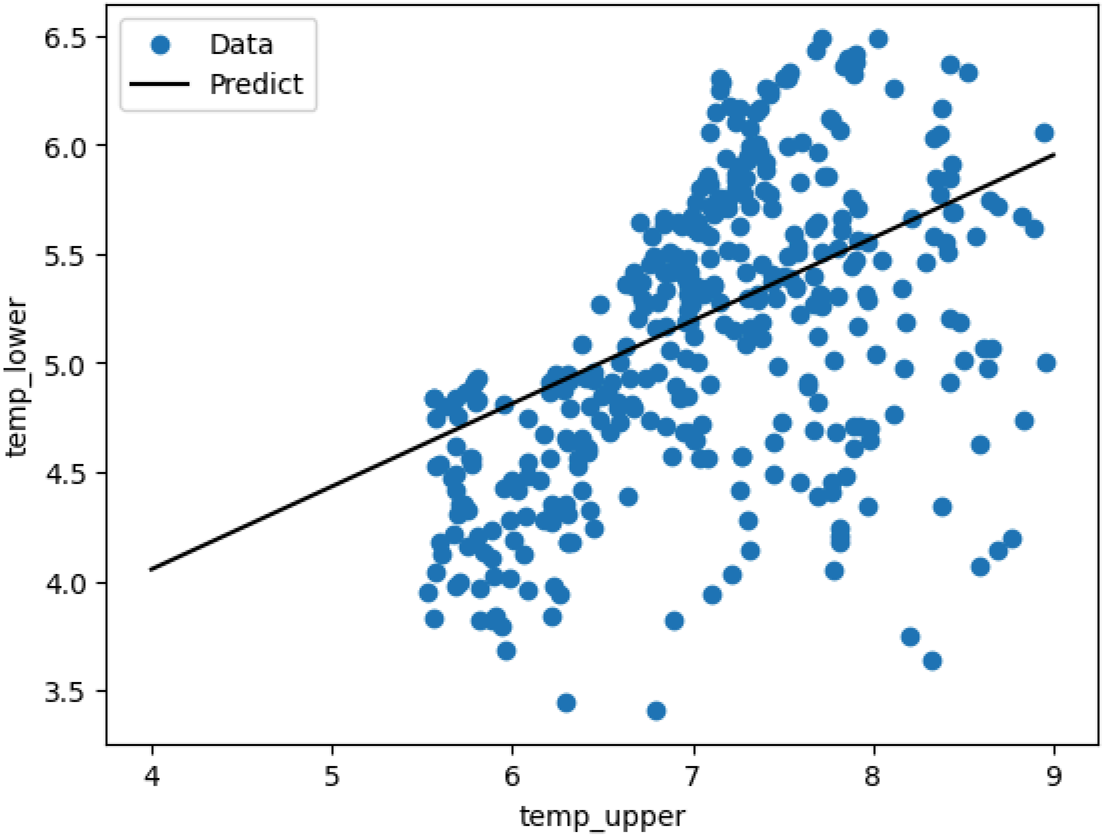
In [365…

```
x = tf.linspace(4., 9., 51)
y = l2_model.predict(x)
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
```

In [366…

```
plot_rm(x, y)
```



8. Определите на контрольной выборке (с нормализованным зависимым признаком) модель множественной регрессии с наиболее высоким качеством по показателю, указанному в индивидуальном задании, среди построенных моделей(R^2).

Linear = 0.2185

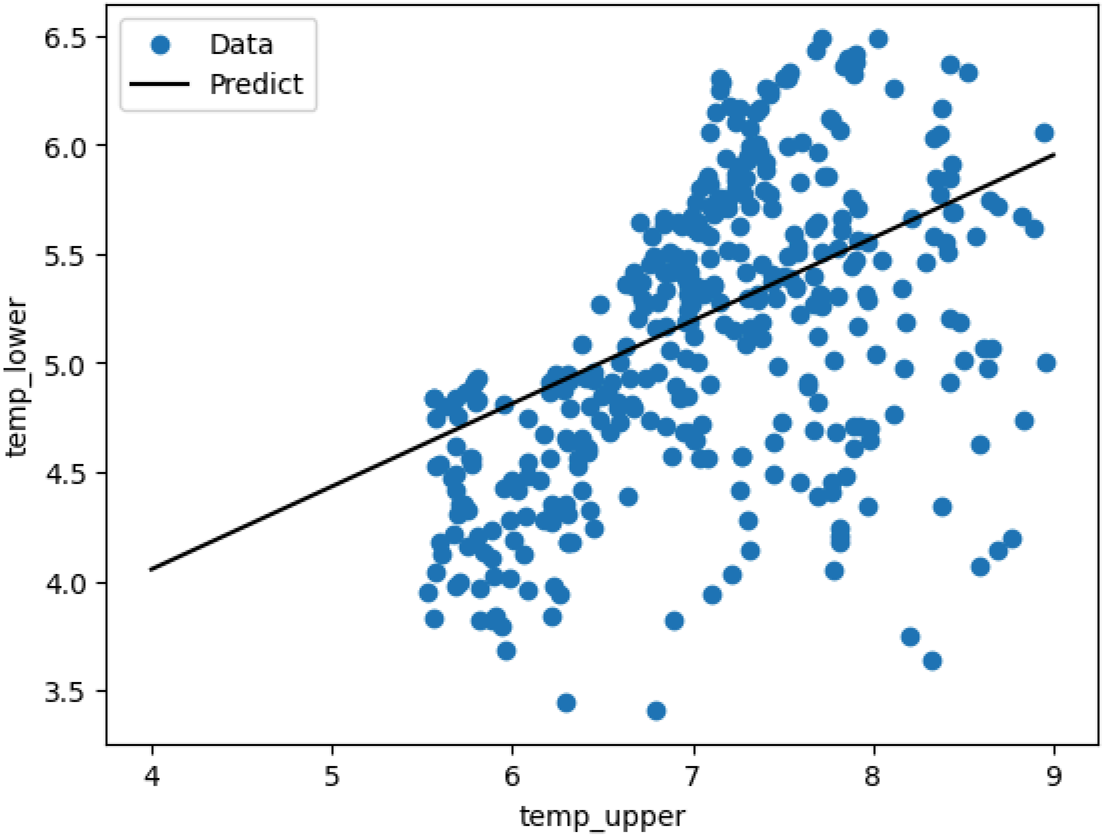
L1 = 0.1416

L2 = 0.0763

Данные можно найти выше по каждой из регрессоров в пунтке н.7 в конце обучения

9. Для лучшего регрессора визуализируйте кривые обучения (в зависимости от эпохи обучения).

```
In [368…  plot_rm(x, y)
```



10. Определите медианные значения признаков (кроме независимого и зависимого признаков) и для построенных медианных значений визуализируйте на плоскости с независимым признаком в качестве оси абсцисс и зависимым признаком в

качестве оси ординат точки тестовой выборки и линии (графики) различных моделей множественной регрессии разными цветами. Подпишите оси и создайте легенду и заголовок для рисунка.

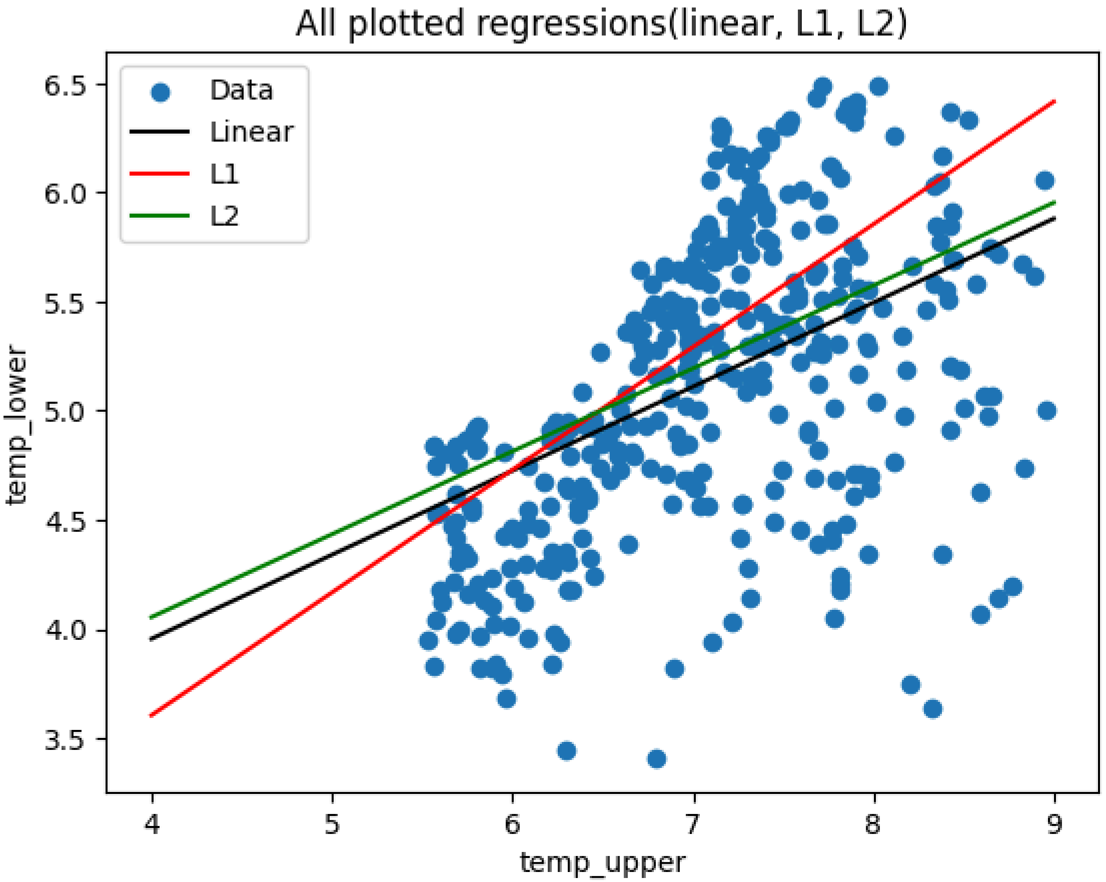
In [371…

```
x = tf.linspace(4., 9., 51)
y_lin = linear_model.predict(x)
y_l1 = l1_model.predict(x)
y_l2 = l2_model.predict(x)
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
```

In [372…

```
plot_all_regressions(x, y_lin, x, y_l1, x, y_l2)
```



In [ ]: