```
In [ ]:
```

```
! pip install eli5
! pip install shap
```

```
In [ ]:
```

```python
import os
import warnings

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

from sklearn.metrics import (explained_variance_score, mean_absolute_error,
mean_absolute_percentage_error,
                            mean_squared_error)
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.preprocessing import (MinMaxScaler,    # Standardization
                                    StandardScaler)

from tensorflow import keras
from keras.callbacks import EarlyStopping    # Early Stopping Callback
from keras.layers import Dense, Dropout, Activation
from keras.models import Sequential

import eli5
from eli5.sklearn import PermutationImportance
from eli5.sklearn import explain_weights_sklearn

from eli5.formatters import format_as_dataframe

import shap

from shap.plots import _waterfall

pd.options.display.max_columns = 60
pd.options.display.max_rows = 60
```

```
In [ ]:
```

```python
df_final = pd.read_csv("df.csv")
best_models=pd.read_csv("nn_results.csv")
```

```
In [ ]:
```

```python
best_models
```

```
Out[ ]:
```

	district	nodes	function	optimizer	MSE	RMSE	MAE	MAPE	va
0	Адмиралтейский	(900, 600, 300, 200)	relu	rmsprop	5.735887e+12	2.394484e+06	1.529950e+06	0.095571	0.
1	Василеостровский	(700, 600, 300, 200)	relu	adam	1.418911e+12	1.191176e+06	6.972433e+05	0.044369	0.

	district	nodes	function	optimizer	MSE	RMSE	MAE	MAPE	va
2	Всеволожский	(500, 300, 200, 100)	elu	adam	7.474442e+10	2.731549e+05	1.937065e+05	0.032861	0.
3	Выборгский	(700, 600, 300, 200)	leaky_relu	adam	9.249094e+11	9.616759e+05	6.535724e+05	0.046423	0.
4	Калининский	(800, 600, 300, 200)	elu	amsgrad	7.371719e+11	8.585849e+05	5.890579e+05	0.045940	0.
5	Кировский	(800, 600, 300, 200)	relu	rmsprop	2.157907e+10	1.464360e+05	1.059646e+05	0.020736	0.
6	Колпинский	(700, 600, 300, 200)	relu	rmsprop	2.086386e+11	4.552913e+05	3.062644e+05	0.038626	0.
7	Красногвардейский	(900, 600, 300, 200)	elu	adam	1.506467e+11	3.881187e+05	2.142705e+05	0.024757	0.
8	Красносельский	(400, 300, 200, 100)	relu	rmsprop	1.257268e+11	3.541551e+05	2.165371e+05	0.026597	0.
9	Курортный	(700, 600, 300, 200)	elu	adam	1.011010e+12	1.005450e+06	7.252853e+05	0.052479	0.
10	Ломоносовский	(700, 500, 300, 200)	elu	adam	1.200104e+11	3.461406e+05	2.629920e+05	0.039831	0.
11	Московский	(800, 600, 300, 200)	elu	adam	6.832705e+11	8.265959e+05	5.224075e+05	0.038525	0.
12	Невский	(900, 600, 300, 200)	relu	adam	3.170771e+11	5.630856e+05	3.827311e+05	0.038594	0.
13	Петроградский	(900, 600, 300, 200)	elu	rmsprop	2.239664e+13	4.732492e+06	3.235153e+06	0.096715	0.
14	Петродворцовый	(700, 500, 300, 200)	elu	rmsprop	3.651652e+11	6.042699e+05	3.525507e+05	0.039982	0.
15	Приморский	(900, 600, 300, 200)	relu	rmsprop	1.332080e+12	1.154062e+06	7.140434e+05	0.047433	0.
16	Пушкинский	(900, 600, 300,	leaky_relu	amsgrad	5.109195e+10	2.260228e+05	1.642825e+05	0.031499	0.

	district	200) nodes	function	optimizer	MSE	RMSE	MAE	MAPE	va
17	Фрунзенский	(700, 500, 300, 200)	relu	rmsprop	4.389163e+11	6.621107e+05	4.423768e+05	0.039278	0.
18	Центральный	(900, 600, 300, 200)	relu	rmsprop	5.961994e+13	7.719899e+06	6.052025e+06	0.184736	0.

In [ ]:

```python
list_of_features=[]
list_of_feature_names=[]
list_of_shaps=[]
#list_of_predictions=[]

for i in range (len(best_models)) :
    if ((best_models['district'][[i]]=='Адмиралтейский-Центральный').values[0]):
        df = df_final[(df_final["district"] == 'Адмиралтейский')|(df_final["distr
ict"] == 'Центральный')]
    elif ((best_models['district'][[i]]=='Красносельский-Кировский').values[0]):
        df = df_final[(df_final["district"] == 'Красносельский')|(df_final["distr
ict"] == 'Кировский')]
    elif ((best_models['district'][[i]]=='Приморский-Курортный').values[0]):
        df = df_final[(df_final["district"] == 'Приморский')|(df_final["district"
] == 'Курортный')]
    else:
        df = df_final[df_final["district"] == best_models['district'][[i]].values
[0]]

    df = pd.get_dummies(df, drop_first=True)

    X = df.drop("full_price", axis=1)
    y = df["full_price"]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rand
om_state=42)

    scaler = MinMaxScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    node=list(best_models['nodes'][[i]].astype(str).str.replace("(", "").str.repl
ace(")", "").str.split(", "))[0]

    if best_models['function'][i]=='relu':
        func=keras.layers.ReLU()
    elif best_models['function'][i]=='leaky_relu':
        func=keras.layers.LeakyReLU()
    elif best_models['function'][i]=='elu':
        func=keras.layers.ELU()

    if best_models['optimizer'][i]=='rmsprop':
        opt=keras.optimizers.legacy.RMSprop()
    elif best_models['optimizer'][i]=='adam':
        opt=keras.optimizers.legacy.Adam()
    elif best_models['optimizer'][i]=='amsgrad':
        opt=keras.optimizers.legacy.Adam(amsgrad=True)

    model = keras.models.Sequential(
                    [
```

```
                    Dense(
                        int(node[0]),
                        activation=func,
                        kernel_initializer="he_uniform",
                        input_dim=X_train.shape[1],
                    ),
                    Dense(
                        int(node[1]), activation=func, kernel_initializer="he
uniform"
                    ),
                    Dropout(0.3, seed=123),
                    Dense(
                        int(node[2]), activation=func, kernel_initializer="he
uniform"
                    ),
                    Dropout(0.3, seed=123),
                    Dense(
                        int(node[3]), activation=func, kernel_initializer="he
uniform"
                    ),
                    Dense(1, activation="linear"),
                ]
            )
    model.compile(
            loss="mse",
            optimizer=opt,
        )

    early_stop = EarlyStopping(
            monitor="val_loss", mode="min", verbose=0, patience=5
        )

    history=model.fit(
            X_train,
            y_train.values,
            epochs=500,
            batch_size=128,
            validation_data=(X_test, y_test.values),
            callbacks=[early_stop],
            verbose=0,
        )

    e = shap.KernelExplainer(model, shap.sample(X_train, 10))
    shap_values = e.shap_values(X_test)
    list_of_shaps.append(shap_values)
    list_of_features.append(X_test)
    list_of_feature_names.append(X.columns)

    del model
```

In [ ]:

```
plt.rcParams.update({'font.size': 10})
plt.rcParams.update({'font.family': "Times New Roman"})

shap.initjs()
pltRowsNmb, pltColsNmb = 7, 3
fig = plt.figure(figsize=[50, 100])
plt.subplots_adjust(wspace=0.4, hspace=0.6)


ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 1)
shap.summary_plot(shap_values=list_of_shaps[0][0]*(10**-3),
```

```python
                            features=list_of_features[0],
                            feature_names=list_of_feature_names[0], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[0]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")


ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 2)
shap.summary_plot(shap_values=list_of_shaps[1][0]*(10**-3),
                            features=list_of_features[1],
                            feature_names=list_of_feature_names[1], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[1]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 3)
shap.summary_plot(shap_values=list_of_shaps[2][0]*(10**-3),
                            features=list_of_features[2],
                            feature_names=list_of_feature_names[2], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[2]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 4)
shap.summary_plot(shap_values=list_of_shaps[3][0]*(10**-3),
                            features=list_of_features[3],
                            feature_names=list_of_feature_names[3], # названия фичей
```

```python
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[3]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 5)
shap.summary_plot(shap_values=list_of_shaps[4][0]*(10**-3),
                            features=list_of_features[4],
                            feature_names=list_of_feature_names[4], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[4]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 6)
shap.summary_plot(shap_values=list_of_shaps[5][0]*(10**-3),
                            features=list_of_features[5],
                            feature_names=list_of_feature_names[5], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
                            plot_size=(20,25)
                            )
ax.set_title(best_models['district'][[5]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 7)
shap.summary_plot(shap_values=list_of_shaps[6][0]*(10**-3),
                            features=list_of_features[6],
                            feature_names=list_of_feature_names[6], # названия фичей
                            max_display=5, # сколько фич отображать
                            show=False,
                            color_bar=False,
```

```python
                                plot_size=(20,25)
                                )
ax.set_title(best_models['district'][[6]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 8)
shap.summary_plot(shap_values=list_of_shaps[7][0]*(10**-3),
                  features=list_of_features[7],
                  feature_names=list_of_feature_names[7], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[7]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 9)
shap.summary_plot(shap_values=list_of_shaps[8][0]*(10**-3),
                  features=list_of_features[8],
                  feature_names=list_of_feature_names[8], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[8]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 10)
shap.summary_plot(shap_values=list_of_shaps[9][0]*(10**-3),
                  features=list_of_features[9],
                  feature_names=list_of_feature_names[9], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[9]].values[0], fontsize = 12)
```

```python
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 11)
shap.summary_plot(shap_values=list_of_shaps[10][0]*(10**-3),
                  features=list_of_features[10],
                  feature_names=list_of_feature_names[10], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[10]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 12)
shap.summary_plot(shap_values=list_of_shaps[11][0]*(10**-3),
                  features=list_of_features[11],
                  feature_names=list_of_feature_names[11], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[11]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 13)
shap.summary_plot(shap_values=list_of_shaps[12][0]*(10**-3),
                  features=list_of_features[12],
                  feature_names=list_of_feature_names[12], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[12]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
```

```python
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 14)
shap.summary_plot(shap_values=list_of_shaps[13][0]*(10**-3),
                  features=list_of_features[13],
                  feature_names=list_of_feature_names[13], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[13]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 15)
shap.summary_plot(shap_values=list_of_shaps[14][0]*(10**-3),
                  features=list_of_features[14],
                  feature_names=list_of_feature_names[14], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[14]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 16)
shap.summary_plot(shap_values=list_of_shaps[15][0]*(10**-3),
                  features=list_of_features[15],
                  feature_names=list_of_feature_names[15], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[15]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
```

```python
        t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 17)
shap.summary_plot(shap_values=list_of_shaps[16][0]*(10**-3),
                  features=list_of_features[16],
                  feature_names=list_of_feature_names[16], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[16]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 18)
shap.summary_plot(shap_values=list_of_shaps[17][0]*(10**-3),
                  features=list_of_features[17],
                  feature_names=list_of_feature_names[17], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[17]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")

ax = fig.add_subplot(pltRowsNmb, pltColsNmb, 20)
shap.summary_plot(shap_values=list_of_shaps[18][0]*(10**-3),
                  features=list_of_features[18],
                  feature_names=list_of_feature_names[18], # названия фичей
                  max_display=5, # сколько фич отображать
                  show=False,
                  color_bar=False,
                  plot_size=(20,25)
                  )
ax.set_title(best_models['district'][[18]].values[0], fontsize = 12)
ax.set_xlabel("Значение SHAP, тыс", fontsize=12)
ax.set_ylabel("Переменные", fontsize=12)
cbar = plt.colorbar()
for t in cbar.ax.get_yticklabels():
    t.set_fontsize(10)
    t.set_fontname("Times New Roman")
    t.set_color("black")
plt.xticks(color="black", fontsize=10, fontname="Times New Roman")
plt.yticks(color="black", fontsize=10, fontname="Times New Roman")
```

```
plt.savefig('nn_biggest_vars.png', bbox_inches='tight', dpi=600)
```

No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored
No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ign
ored

Пер

rooms_2
remont_чистовая

Значение SHAP, тыс

Пер

station_name_driving_Новочеркасская
developer_GloraX

Значение SHAP, тыс

Пер

full_square
parking_подземная

Значение SHAP, тыс

Курортный

Переменные

house_type_монолитный
remont_предчистовая
rooms_2
parking_подземная
quarts_remained

Значение SHAP, тыс

Ломоносовский

Переменные

rooms_2
house_type_панельный
ЖК_Новое Горелово
developer_ЛенРусСтрой
parking_отдельная многоуровневая

Значение SHAP, тыс

Московский

Переменные

full_square
living_square
kitchen
rooms_2
station_name_driving_Московские ворота

Значение SHAP, тыс

Невский

Переменные

ЖК_class_Комфорт
house_type_монолитный
full_square
parking_отдельная многоуровневая
living_square

Значение SHAP, тыс

Петроградский

Переменные

rooms_3
full_square
living_square
station_name_geo_Чкаловская
quarts_remained

Значение SHAP, тыс

Петродворцовый

Переменные

rooms_2
full_square
kitchen
living_square
rooms_3

Значение SHAP, тыс

Приморский

Переменные

full_square
developer_LEGENDA Intelligent Development
floor1
ЖК_Черная речка, 41
living_square

Значение SHAP, тыс

Пушкинский

Переменные

station_name_geo_Купчино
parking_подземная
kitchen
rooms_2
ЖК_Simple (Симпл)

Значение SHAP, тыс

Фрунзенский

Переменные

rooms_2
station_name_geo_Обводный канал
height_total
full_square
living_square

Значение SHAP, тыс

Центральный

Переменные

rooms_3
full_square
living_square
kitchen
rooms_2

Значение SHAP, тыс