



Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления» – ИУ5

Факультет «Радиотехнический» – РТ5

Отчёт по лабораторной работе №6 по курсу Технологии машинного обучения

5

(количество листов)

Исполнитель

студент группы РТ5-616

Нижаметдинов М. Ш.

“ ____ ” _____ 2023 г.

Проверил

Преподаватель кафедры ИУ5

Гапанюк Ю. Е.

“ ____ ” _____ 2023 г.

Москва, 2023 г.

Задание

1. Выберите набор данных (датасет) для решения задачи прогнозирования временного ряда.
2. Визуализируйте временной ряд и его основные характеристики.
3. Разделите временной ряд на обучающую и тестовую выборку.
4. Произведите прогнозирование временного ряда с использованием как минимум двух методов.
5. Визуализируйте тестовую выборку и каждый из прогнозов. Оцените качество прогноза в каждом случае с помощью метрик.

Набор данных

<https://raw.githubusercontent.com/mansurik1/MLT/master/LW%206/Project/data/DailyDelhiClimateTrain.csv>

<https://raw.githubusercontent.com/mansurik1/MLT/master/LW%206/Project/data/DailyDelhiClimateTest.csv>

Исходный текст проекта

```
import numpy as np
import pandas as pd
from matplotlib import pyplot
import matplotlib.pyplot as plt

def clear(data):
    data['humidity'] = None
    data['meantemp'] = None
    data['meanpressure'] = None

data_start =
pd.read_csv('https://raw.githubusercontent.com/mansurik1/MLT/master/LW%206/Project/data/DailyDelhiClimateTrain.csv', header=0, index_col=0, parse_dates=True)

clear(data_start)

data_test =
pd.read_csv('https://raw.githubusercontent.com/mansurik1/MLT/master/LW%206/Project/data/DailyDelhiClimateTest.csv', header=0, index_col=0, parse_dates=True)
clear(data_test)

data_train = data_start['wind_speed']

data_train.head()
```

```
fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Временной ряд в виде графика')
data_train.plot(ax=ax, legend=False)
pyplot.show()
```

```
for i in range(1, 5):
    fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(5,4))
    fig.suptitle(f'Лар порядка {i}')
    pd.plotting.lag_plot(data_train, lag=i, ax=ax)
    pyplot.show()
```

```
fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Плотность вероятности распределения данных')
data_train.plot(ax=ax, kind='kde', legend=False)
pyplot.show()
```

```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(data_train, lags=30)
plt.tight_layout()
```

```
# https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal\_decompose.html
from statsmodels.tsa.seasonal import seasonal_decompose
# Аддитивная модель
def plot_decompose(data=data_train, model='add'):
    result_add = seasonal_decompose(data, model='add')
    fig = result_add.plot()
    fig.set_size_inches((12, 8))
    # Перерисовка
    fig.tight_layout()
    plt.show()
```

```
plot_decompose()
```

```
data_train2 = data_start.copy()
```

```
# Простое скользящее среднее (SMA)
data_train2['SMA_10'] = data_train2['wind_speed'].rolling(10, min_periods=1).mean()
data_train2['SMA_20'] = data_train2['wind_speed'].rolling(20, min_periods=1).mean()
```

```
print(data_train2)
```

```
fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Временной ряд со скользящими средними')
data_train2[:100].plot(ax=ax, legend=True)
pyplot.show()
```

```

!pip install gplearn
from gplearn.genetic import SymbolicRegressor
xnum = list(range(data_train.shape[0]))
print(data_train)

function_set = ['add', 'sub', 'mul', 'div', 'sin']
est_gp = SymbolicRegressor(population_size=2000, metric='mse',
                           generations=40, stopping_criteria=0.01,
                           init_depth=(5, 10), verbose=1, function_set=function_set,
                           const_range=(-100, 100), random_state=0)

est_gp.fit(np.array(xnum).reshape(-1, 1), data_train.values.reshape(-1, 1))

import graphviz
dot_data = est_gp._program.export_graphviz()
graph = graphviz.Source(dot_data)
graph

# Записываем предсказания в DataFrame
data_test['predictions_GPLEARN'] = list(y_gp)

fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Предсказания временного ряда (тестовая выборка)')
data_test.plot(ax=ax, legend=True)
pyplot.show()

# Целочисленная метка шкалы времени
xnum = list(range(data_start.shape[0] + data_test.shape[0]))
# Разделение выборки на обучающую и тестовую
train_size = len(data_start)
train, test = data_start['wind_speed'].values, data_test['wind_speed'].values
history_arima = [x for x in train]
history_es = [x for x in train]

print(train)
print(test)

from sklearn.metrics import mean_squared_error
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Параметры модели (p,d,q)
arima_order = (6,1,0)
# Формирование предсказаний
predictions_arima = list()

```

```
for t in range(len(test)):
    model_arima = ARIMA(history_arima, order=arima_order)
    model_arima_fit = model_arima.fit()
    yhat_arima = model_arima_fit.forecast()[0]
    predictions_arima.append(yhat_arima)
    history_arima.append(test[t])
# Вычисление метрики RMSE

mean_squared_error(test, predictions_arima, squared=False)

data_test['predictions_ARIMA'] = list(predictions_arima)

fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Предсказания временного ряда (тестовая выборка)')
data_test.plot(ax=ax, legend=True)
pyplot.show()
```