**Задание №3 – Регрессия**

Написать программу на Python, которая обучает **четыре регрессионных модели**, построенных на наборе с помощью четырёх алгоритмов:

1. линейный регрессор,
2. полиномиальный регрессор,
3. регрессор, основанный на случайном лесе
4. и один из следующих инструментов: Gaussian Process Regression, Support Vector Regression, Gradient Boosting Regressor или AdaBoost Regressor.

Выбрать признаки, использующиеся при обучении, и, если необходимо, выполнить их предобработку. Разделить выборку на обучающую и тестовую.

В работе необходимо исследовать работу алгоритмов с разными значениями гиперпараметров.

Для моделей на основе деревьев вывести значения важности признаков.

Выбрать наилучшую модель из полученных регрессоров.

Сохранить лучшую модель (pickle).

Спроектировать и реализовать приложение (настольное, Web- или чат-бот) на Python, осуществляющее загрузку модели, проверку корректности ввода данных, требуемых для регрессионной модели. Для реализации программы можно использовать любой инструмент (Flask, Django, Tkinter и т.д.)

Написать короткий отчет по работе, включив в него программы с комментариями, значения качества моделей (коэффициент детерминации, среднюю квадратичную и среднюю абсолютную ошибки).

• если последняя цифра 3 или 8: выборка – Аренда велосипедов (

<https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset>

), предсказываемое значение – количество аренд велосипедов в сутки (Area), файл day.csv;

Надо ли для 4 разных регрессоров приводить данные к одному ввиду?

Обучить **четыре регрессионных модели**, построенных на наборе с помощью четырёх алгоритмов:

1. линейный регрессор,
2. полиномиальный регрессор,
3. регрессор, основанный на случайном лесе
4. и один из следующих инструментов: Gaussian Process Regression, Support Vector Regression, Gradient Boosting Regressor или AdaBoost Regressor.

Выбрать признаки, использующиеся при обучении, и, если необходимо, выполнить их предобработку. Разделить выборку на обучающую и тестовую.

В работе необходимо исследовать работу алгоритмов с разными значениями гиперпараметров.

Для моделей на основе деревьев вывести значения важности признаков.

Выбрать наилучшую модель из полученных регрессоров.

Сохранить лучшую модель (pickle).

Код:  
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures, StandardScaler

from sklearn.pipeline import make\_pipeline

from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor

from sklearn.svm import SVR

from sklearn.metrics import mean\_squared\_error

from joblib import dump

import pickle

#открываем файл с данными и выводим на экран

data = pd.read\_csv("day.csv", index\_col = 0)

data=data.reset\_index()

data

#Описание данных

data.info()

Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 instant 731 non-null int64

1 dteday 731 non-null object

2 season 731 non-null int64

3 yr 731 non-null int64

4 mnth 731 non-null int64

5 holiday 731 non-null int64

6 weekday 731 non-null int64

7 workingday 731 non-null int64

8 weathersit 731 non-null int64

9 temp 731 non-null float64

10 atemp 731 non-null float64

11 hum 731 non-null float64

12 windspeed 731 non-null float64

13 casual 731 non-null int64

14 registered 731 non-null int64

15 cnt 731 non-null int64

dtypes: float64(4), int64(11), object(1)

memory usage: 91.5+ KB

missing\_values = data.isnull().sum()

print("Число пропущенных значений по атрибутам:")

print(missing\_values)

Число пропущенных значений по атрибутам:

instant 0

dteday 0

season 0

yr 0

mnth 0

holiday 0

weekday 0

workingday 0

weathersit 0

temp 0

atemp 0

hum 0

windspeed 0

casual 0

registered 0

cnt 0

dtype: int64

# Выбор признаков для анализа