

Лабораторная работа №6
по дисциплине
«Технологии машинного обучения»
на тему
«Ансамбли моделей машинного обучения»

Выполнил:
студент группы ИУ5-61Б
Белоусов Е. А.

1. Цель

изучение ансамблей моделей машинного обучения.

2. Задание

Выберите набор данных (датасет) для решения задачи классификации или регрессии. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую. Обучите две ансамблевые модели. Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.

```
[1]: import numpy as np
import pandas as pd
import sklearn
```

```
[2]: data = pd.read_csv('../data/winequality-red.csv')
```

```
[3]: data.head()
```

```
[3]: fixed acidity volatile acidity citric acid residual sugar chlorides \
0      7.4          0.70      0.00      1.9    0.076
1      7.8          0.88      0.00      2.6    0.098
2      7.8          0.76      0.04      2.3    0.092
3     11.2          0.28      0.56      1.9    0.075
4      7.4          0.70      0.00      1.9    0.076
```

```
free sulfur dioxide total sulfur dioxide density pH sulphates \
0         11.0          34.0  0.9978  3.51    0.56
1         25.0          67.0  0.9968  3.20    0.68
2         15.0          54.0  0.9970  3.26    0.65
3         17.0          60.0  0.9980  3.16    0.58
4         11.0          34.0  0.9978  3.51    0.56
```

```
alcohol quality
0    9.4      5
1    9.8      5
2    9.8      5
3    9.8      6
4    9.4      5
```

```
[4]: data.shape
```

```
[4]: (1599, 12)
```

```
[5]: y = data['quality']
del data['quality']
```

```
[6]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.3, random_state=42)
```

```
[7]: from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
forest = RandomForestRegressor()
forest.fit(X_train, y_train)
mean_squared_error(forest.predict(X_test), y_test)
```

[7]: 0.33750583333333334

```
[8]: import xgboost as xgb
boosting = xgb.XGBRegressor()
boosting.fit(X_train, y_train)
mean_squared_error(boosting.predict(X_test), y_test)
```

[8]: 0.39578926361093103

```
[9]: from gmdhpy import gmdh
model = gmdh.Regressor(ref_functions=('linear_cov', 'quadratic', 'cubic', 'linear'))
y_train_pd = pd.DataFrame(y_train)
model.fit(X_train.values, y_train_pd.values )
mean_squared_error(boosting.predict(X_test), y_test)
```

train layer0 in 0.21 sec
train layer1 in 0.83 sec
train layer2 in 0.80 sec
train layer3 in 0.80 sec
train layer4 in 0.78 sec
train layer5 in 0.79 sec
train layer6 in 0.83 sec
train layer7 in 0.80 sec
train layer8 in 0.79 sec
train layer9 in 0.79 sec
train layer10 in 0.83 sec
train layer11 in 0.94 sec

[9]: 0.39578926361093103

[]: