

**Московский государственный технический
университет им. Н. Э. Баумана**

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №6

Выполнил:
Флоринский В. А.
группа ИУ5-64Б

Проверил:
Гапанюк Ю.Е.

Дата: 07.04.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

Ансамбли моделей машинного обучения. Часть 2.

Цель лабораторной работы: изучение ансамблей моделей машинного обучения.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую.
4. Обучите следующие ансамблевые модели:
 - одну из моделей группы стекинга.
 - модель [многослойного персептрона](#). По желанию, вместо библиотеки `scikit-learn` возможно использование библиотек [TensorFlow](#), [PyTorch](#) или других аналогичных библиотек.
 - двумя методами на выбор из семейства МГУА (один из линейных методов [COMBI](#) / [MULTI](#) + один из нелинейных методов [MIA](#) / [RIA](#)) с использованием библиотеки [gmdh](#).
 - В настоящее время библиотека МГУА не позволяет решать задачу классификации !!!

- 5. Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.**
- 6. В телеграмм-канале потока ИУ5 в теме ТМО_МГУА напишите обратную связь по использованию библиотеки gmdh:**
- обнаруженные баги с приложением скриншотов ошибок, за каждый найденный баг +1 балл на экзамене;
 - опечатки в документации или учебном пособии МГУА;
 - возникшие вопросы или трудности при установке и использовании библиотеки;
 - любая другая информация (критика, предложения по улучшению и тд).
- 7. Справочные материалы по МГУА:**
- [Видеозапись доклада.](#)
 - [Учебное пособие по МГУА \(предварительная версия\).](#)
 - [Примеры использования библиотеки.](#)

Ход выполнения:

```

from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

data = load_diabetes()
X, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

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

```

[16]

Python

```

from sklearn.ensemble import StackingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor

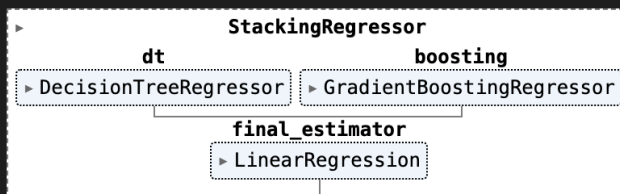
estimators = [
    ('dt', DecisionTreeRegressor(random_state=42)),
    ('boosting', GradientBoostingRegressor(random_state=42))
]

stacking = StackingRegressor(estimators=estimators, final_estimator=LinearRegression())
stacking.fit(X_train, y_train)

```

[17]

Python



```

from sklearn.neural_network import MLPRegressor

mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)

```

[18]

Python

C:\Users\zusha\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\loc
warnings.warn(

```

MLPRegressor
MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)

```

```
from gmdh import Combi, Mia

combi = Combi()
combi.fit(X_train, y_train)
y_pred_combi = combi.predict(X_test)

mia = Mia()
mia.fit(X_train, y_train)
y_pred_mia = mia.predict(X_test)
```

19]

Python

```
from sklearn.metrics import mean_squared_error, r2_score

y_pred_stack = stacking.predict(X_test)
print(f"Stacking: MSE = {mean_squared_error(y_test, y_pred_stack):.4f}, R² = {r2_score(y_test, y_pre

y_pred_mlp = mlp.predict(X_test)
print(f"MLP: MSE = {mean_squared_error(y_test, y_pred_mlp):.4f}, R² = {r2_score(y_test, y_pred_mlp):

print(f"COMBI: MSE = {mean_squared_error(y_test, y_pred_combi):.4f}, R² = {r2_score(y_test, y_pred_c

print(f"MIA: MSE = {mean_squared_error(y_test, y_pred_mia):.4f}, R² = {r2_score(y_test, y_pred_mia):
```

20]

Python

```
.. Stacking: MSE = 3066.0214, R² = 0.4320
MLP: MSE = 3026.0175, R² = 0.4395
COMBI: MSE = 2943.8537, R² = 0.4547
MIA: MSE = 2748.4422, R² = 0.4909
```